

53-1002494-02
20 July 2012



FastIron

Configuration Guide

Supporting FastIron Software Release 07.4.00a

BROCADE

© 2012 Brocade Communications Systems, Inc. All Rights Reserved.

Brocade, Brocade Assurance, the B-wing symbol, DCX, Fabric OS, MLX, SAN Health, VCS, and VDX are registered trademarks, and AnyIO, Brocade One, CloudPlex, Effortless Networking, ICX, NET Health, OpenScript, and The Effortless Network are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Other brands, products, or service names mentioned may be trademarks of their respective owners.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government

The authors and Brocade Communications Systems, Inc. shall have no liability or responsibility to any person or entity with respect to any loss, cost, liability, or damages arising from the information contained in this book or the computer programs that accompany it.

The product described by this document may contain "open source" software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit <http://www.brocade.com/support/oscd>.

Brocade Communications Systems, Incorporated

Corporate and Latin American Headquarters
Brocade Communications Systems, Inc.
130 Holger Way
San Jose, CA 95134
Tel: 1-408-333-8000
Fax: 1-408-333-8101
E-mail: info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems China HK, Ltd.
No. 1 Guanghua Road
Chao Yang District
Units 2718 and 2818
Beijing 100020, China
Tel: +8610 6588 8888
Fax: +8610 6588 9999
E-mail: china-info@brocade.com

European Headquarters
Brocade Communications Switzerland Sàrl
Centre Swissair
Tour B - 4ème étage
29, Route de l'Aéroport
Case Postale 105
CH-1215 Genève 15
Switzerland
Tel: +41 22 799 5640
Fax: +41 22 799 5641
E-mail: emea-info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems Co., Ltd. (Shenzhen WFOE)
Citic Plaza
No. 233 Tian He Road North
Unit 1308 - 13th Floor
Guangzhou, China
Tel: +8620 3891 2000
Fax: +8620 3891 2111
E-mail: china-info@brocade.com

Document History

Title	Publication number	Summary of changes	Date
<i>FastIron Configuration Guide</i>	53-1002494-01	Release 07.4.00	March 2012
<i>FastIron Configuration Guide</i>	53-1002494-02	Release 07.4.00 document updated with enhancements in Release 07.4.00a	July 2012

Contents

About This Guide

Introduction	li
Device nomenclature	li
Audience	lii
What's new in this document	lii
Summary of enhancements in FastIron release 07.4.00a	lii
Summary of enhancements in FastIron release 07.4.00	liii
Unsupported features	lv
Document conventions	lvi
Text formatting	lvi
Command syntax conventions	lvii
Notes, cautions, and danger notices	lvii
Related publications	lvii
Getting technical help	lviii

Chapter 1

Management Applications

Management port overview	1
How the management port works.	2
CLI Commands for use with the management port.	2
Logging on through the CLI.	4
Online help	4
Command completion	4
Scroll control.	5
Line editing commands.	5
Using stack-unit, slot number, and port number with CLI commands.	6
CLI nomenclature on Chassis-based models.	6
CLI nomenclature on FESX Compact devices	6
CLI nomenclature on Stackable devices	7
Searching and filtering output from CLI commands	7
Using special characters in regular expressions	9
Creating an alias for a CLI command	11
Logging on through the Web Management Interface	12
Navigating the Web Management Interface	13

Chapter 2

Basic Software Features

Basic system parameter configuration	18
Entering system administration information	19
SNMP parameter configuration	19
Displaying virtual routing interface statistics	22
Disabling Syslog messages and traps for CLI access	23
Cancelling an outbound Telnet session	24
Specifying an SNTP server	24
Configuring the device as an SNTP server	28
Displaying SNTP server information	29
Enabling broadcast mode for an SNTP client	30
Setting the system clock	30
Limiting broadcast, multicast, and unknown unicast traffic	32
CLI banner configuration	37
Local MAC address for Layer 2 management traffic	40
Basic port parameter configuration	40
Assigning a port name	40
Port speed and duplex mode modification	41
Enabling auto-negotiation maximum port speed advertisement and down-shift	42
Modifying port duplex mode	45
MDI and MDIX configuration	45
Disabling or re-enabling a port	46
Flow control configuration	47
Symmetric flow control on FCX devices	49
PHY FIFO Rx and Tx depth configuration	53
Interpacket Gap (IPG) on a FastIron X Series switch	53
IPG on FastIron Stackable devices	55
Enabling and disabling support for 100BaseTX	57
Enabling and disabling support for 100BaseFX	57
Changing the Gbps fiber negotiation mode	59
Port priority (QoS) modification	60
Dynamic configuration of Voice over IP (VoIP) phones	60
Port flap dampening configuration	61
Port loop detection	65

Chapter 3

Operations, Administration, and Maintenance

OAM Overview	71
Software versions installed and running on a device	72
Determining the flash image version running on the device ..	72
Displaying the boot image version running on the device	74
Displaying the image versions installed in flash memory	74
Flash image verification	75
Image file types	76
Software upgrades	77
Boot code synchronization feature	77
Viewing the contents of flash files	77

Using SNMP to upgrade software	80
Software reboot.....	81
Software boot configuration notes.....	81
Displaying the boot preference	81
Loading and saving configuration files	82
Replacing the startup configuration with the running configuration	83
Replacing the running configuration with the startup configuration.....	83
Logging changes to the startup-config file.....	83
Copying a configuration file to or from a TFTP server	83
Dynamic configuration loading	84
Maximum file sizes for startup-config file and running-config	86
Loading and saving configuration files with IPv6	87
Using the IPv6 copy command	87
Copying a file from an IPv6 TFTP server.....	88
IPv6 ncopy command	89
IPv6 TFTP server file upload	90
Using SNMP to save and load configuration information	91
Erasing image and configuration files	92
System reload scheduling	92
Reloading at a specific time	92
Reloading after a specific amount of time.....	93
Displaying the amount of time remaining before a scheduled reload	93
Canceling a scheduled reload.....	93
Diagnostic error codes and remedies for TFTP transfers.....	93
Network connectivity testing	95
Pinging an IPv4 address	95
Tracing an IPv4 route.....	96
Hitless management on the FSX 800 and FSX 1600.....	97
Benefits of hitless management.....	98
Supported protocols and services for hitless management events	98
Hitless management configuration notes and feature limitations	100
Hitless reload or switchover requirements and limitations ..	100
What happens during a Hitless switchover or failover	101
Enabling hitless failover on the FSX 800 and FSX 1600	103
Executing a hitless switchover on the FSX 800 and FSX 1600	104
Hitless OS upgrade on the FSX 800 and FSX 1600	104
Syslog message for Hitless management events	107
Displaying diagnostic information.....	108

Chapter 4

Security Access

Securing access methods	110
-------------------------------	-----

Remote access to management function restrictions	112
ACL usage to restrict remote access	112
Defining the console idle time	115
Remote access restrictions.....	115
Restricting access to the device based on IP or MAC address	117
Defining the Telnet idle time	118
Changing the login timeout period for Telnet sessions.....	118
Specifying the maximum number of login attempts for Telnet access	118
Changing the login timeout period for Telnet sessions.....	118
Restricting remote access to the device to specific VLAN IDs	119
Designated VLAN for Telnet management sessions to a Layer 2 Switch.....	120
Device management security	121
Disabling specific access methods.....	122
Passwords used to secure access	124
Setting a Telnet password	124
Setting passwords for management privilege levels	125
Recovering from a lost password	127
Displaying the SNMP community string	128
Specifying a minimum password length.....	128
Local user accounts	129
Enhancements to username and password	129
Local user account configuration	133
Creating a password option.....	135
Changing a local user password.....	136
SSL security for the Web Management Interface	136
Enabling the SSL server on the Brocade device	137
Changing the SSL server certificate key size.....	137
Support for SSL digital certificates larger than 2048 bits ...	138
Importing digital certificates and RSA private key files.....	138
Generating an SSL certificate.....	138

TACACS and TACACS+ security	139
How TACACS+ differs from TACACS.	139
TACACS/TACACS+ authentication, authorization, and accounting	139
TACACS authentication	142
TACACS/TACACS+ configuration considerations	145
Enabling TACACS	146
Identifying the TACACS/TACACS+ servers.	146
Specifying different servers for individual AAA functions ...	147
Setting optional TACACS and TACACS+ parameters	147
Configuring authentication-method lists for TACACS and TACACS+	149
Configuring TACACS+ authorization	151
TACACS+ accounting configuration.	154
Configuring an interface as the source for all TACACS and TACACS+ packets	155
Displaying TACACS/TACACS+ statistics and configuration information	156
RADIUS security.	157
RADIUS authentication, authorization, and accounting	157
RADIUS configuration considerations.	160
Configuring RADIUS.	161
Brocade-specific attributes on the RADIUS server	161
Enabling SNMP to configure RADIUS	163
Identifying the RADIUS server to the Brocade device	163
Specifying different servers for individual AAA functions ...	163
RADIUS server per port	164
RADIUS server to individual ports mapping.	165
RADIUS parameters.	166
Setting authentication-method lists for RADIUS	167
RADIUS authorization	169
RADIUS accounting	171
Configuring an interface as the source for all RADIUS packets	172
Displaying RADIUS configuration information	172
Authentication-method lists	174
Configuration considerations for authentication- method lists	175
Examples of authentication-method lists.	175
TCP Flags - edge port security	177
Using TCP Flags in combination with other ACL features	178

Chapter 5

SSH2 and SCP

SSH version 2 overview	179
Tested SSH2 clients.	180
SSH2 supported features	180
SSH2 unsupported features	180

SSH2 authentication types	181
Enabling and disabling SSH by generating and deleting host keys	181
Configuring DSA or RSA challenge-response authentication	183
Optional SSH parameters	185
Setting the number of SSH authentication retries	186
Deactivating user authentication	186
Enabling empty password logins	186
Setting the SSH port number	187
Setting the SSH login timeout value	187
Designating an interface as the source for all SSH packets	187
Configuring the maximum idle time for SSH sessions	187
Filtering SSH access using ACLs	188
Terminating an active SSH connection	188
Displaying SSH information	188
Displaying SSH connection information	188
Displaying SSH configuration information	189
Displaying additional SSH connection information	190
Secure copy with SSH2	191
Enabling and disabling SCP	191
Secure copy configuration notes	191
Example file transfers using SCP	191
SSH2 client	195
Enabling SSH2 client	195
Configuring SSH2 client public key authentication	195
Using SSH2 client	196
Displaying SSH2 client information	197

Chapter 6

Software-based Licensing

Software license terminology	199
Software-based licensing overview	200
How software-based licensing works	200
Seamless transition for legacy devices	201
License types	201
Non-licensed features	202
Licensed features and part numbers	202
Licensing rules	206
General notes about licensing	206
Licensing rules for FCX and ICX 6610 devices	207
Licensing rules for FESX6, FSX 800, and FSX 1600 devices	208
Licensing for Ports on Demand	208
Configuring PoD on an interface	208
Configuring the upper PoD ports in a stack for ICX 6610 devices only	209
Displaying license configuration for PoD ports after a license upgrade	210

Upgrading or downgrading configuration considerations for PoD .	212
Configuration considerations for stacking or trunking PoD ports .	213
Configuration considerations when configuring PoD on an interface .	213
Configuration considerations when configuring PoD for ICX 6450 devices only .	214
Software licensing configuration tasks .	216
Obtaining a license .	216
Installing a license file .	221
Using TFTP to copy a license file on FESX, SX 800 and SX 1600, and FWS devices .	221
Using TFTP to copy a license file on FCX and ICX devices .	221
Using Secure Copy to install a license .	221
Verifying the license file installation .	222
Deleting a license file .	222
Deleting a license on FESX, SX 800 and SX 1600, and FWS devices .	222
Deleting a license on FCX and ICX devices .	223
Using a trial license .	223
What happens when a trial license expires .	223
Console, syslog, and trap messages for trial license expiration .	224
Renewing or extending a trial license .	224
Viewing software license information from the Brocade software portal .	224
Transferring a license .	226
Special replacement instructions for legacy devices .	226
Syslog messages and trap information .	227
Viewing information about software licenses .	227
Viewing the License ID .	227
Viewing the license database .	229
Viewing software packages installed in the device .	232

Chapter 7

Brocade Stackable Devices

Brocade IronStack overview .	236
Brocade IronStack features .	236
Brocade stackable models .	236
Brocade IronStack terminology .	237
Supported IronStack topologies .	239
Brocade IronStack topologies .	239

Connecting ICX 6450 and ICX 6430 devices in a stack	244
Connecting ICX 6450 devices in a stack.	245
Connecting ICX 6430 devices in a stack	245
Trunking configuration considerations for ICX 6430 and ICX 6450 devices.	245
Software requirements	248
IronStack construction methods.	249
Scenario 1 - Three-member IronStack in a ring topology using secure-setup.	250
Scenario 2 - Three-member IronStack in a ring topology using the automatic setup process.	254
Scenario 3 - Three member IronStack in a ring topology using the manual configuration process	257
FCX IronStack configuration.	258
Configuring FCX stacking ports.	259
Configuring a default stacking port to function as a data port	264
Configuring an ICX 6610 IronStack	265
ICX 6610 trunked stacking ports configuration.	265
Configuration notes for ICX stack topologies	265
Periodic background stack diagnosis for ICX 6610 devices	266
Configuring an ICX 6430 and ICX 6450 IronStack	266
Configuring ICX 6430 or ICX 6450 trunked stacking ports	267
Configuring ICX 6430 or ICX 6450 multi-trunked stacking ports.	267
Periodic background stack diagnosis for ICX 6430 and ICX 6450 devices	268
Error messages encountered during the configuration of an ICX 6430 or ICX 6450 IronStack	269
Verifying an IronStack configuration	270
Brocade IronStack management	273
Logging in through the CLI.	273
Logging in through the console port.	273
IronStack management MAC address	275
Removing MAC address entries	277
CLI command syntax for stack units.	279
IronStack CLI commands	279
Important notes about stacking images	281
Copying the flash image to a stack unit from the Active Controller.	283
Reloading a stack unit.	283
Controlling stack topology	283
Managing IronStack partitioning.	284
MIB support for the IronStack.	285
Persistent MAC address for the IronStack	286
Unconfiguring an IronStack.	287
Displaying IronStack information	289
Adding, removing, or replacing units in an IronStack	311
Renumbering stack units	313
Syslog, SNMP, and traps for stack units.	314

IronStack troubleshooting	315
Troubleshooting an unsuccessful stack build	315
Troubleshooting a stacking upgrade	317
Troubleshooting image copy issues	318
Stack mismatches	318
Image mismatches	319
Advanced feature privileges (FCX devices only)	319
Configuration mismatch for stack units	320
Auto Image Copy for stack units	321
Memory allocation failure	322
Recovering from a stack unit mismatch	323
Troubleshooting secure-setup	324
Troubleshooting unit replacement issues	324
More about IronStack technology	325
Configuration, startup configuration files, and stacking flash	325
IronStack topologies	326
Port down and aging	326
Brocade IronStack device roles and elections	326
Hitless stacking	329
Supported hitless stacking events	329
Non-supported hitless stacking events	330
Supported hitless stacking protocols and services	330
Hitless stacking configuration notes and feature limitations	333
What happens during a hitless stacking switchover or failover	333
Standby Controller role in hitless stacking	335
Support during stack formation, stack merge, and stack split	337
Hitless stacking default behavior	340
Hitless stacking failover	342
Hitless stacking switchover	343
Displaying information about hitless stacking	350
Displaying information about stack failover	350
Displaying information about link synchronization status	350
Syslog messages for hitless stacking failover and switchover	351

Chapter 8

IPv6 Configuration on FastIron X Series, FCX, and ICX Series Switches

Full Layer 3 IPv6 feature support	355
IPv6 addressing overview	355
IPv6 address types	356
IPv6 stateless auto-configuration	358
IPv6 CLI command support	358

IPv6 host address on a Layer 2 switch	360
Configuring a global or site-local IPv6 address with a manually configured interface ID	361
Configuring a link-local IPv6 address as a system-wide address for a switch.	361
Configuring the management port for an IPv6 automatic address configuration.	362
Configuring basic IPv6 connectivity on a Layer 3 switch.	362
Enabling IPv6 routing.	362
IPv6 configuration on each router interface	362
Configuring IPv4 and IPv6 protocol stacks.	365
IPv6 management (IPv6 host support)	366
Configuring IPv6 management ACLs	366
Restricting SNMP access to an IPv6 node.	366
Specifying an IPv6 SNMP trap receiver	367
Configuring SNMP V3 over IPv6	367
Configuring SNMP over IPv6.	367
Secure Shell, SCP, and IPv6	367
IPv6 Telnet	367
IPv6 traceroute.	368
IPv6 Web management using HTTP and HTTPS	369
Restricting Web management access	369
Configuring name-to-IPv6 address resolution using IPv6 DNS resolver	370
Defining an IPv6 DNS entry.	370
Pinging an IPv6 address	370
Configuring an IPv6 Syslog server	372
Viewing IPv6 SNMP server addresses	372
Disabling router advertisement and solicitation messages ..	373
Disabling IPv6 on a Layer 2 switch.	373
Static IPv6 route configuration.	373
IPv6 over IPv4 tunnels	376
IPv6 over IPv4 tunnel configuration notes	376
Configuring a manual IPv6 tunnel	377
Clearing IPv6 tunnel statistics	378
Displaying IPv6 tunnel information.	378
ECMP load sharing for IPv6	381
Disabling or re-enabling ECMP load sharing for IPv6	382
Changing the maximum load sharing paths for IPv6	382
Enabling support for network-based ECMP load sharing for IPv6	382
Displaying ECMP load-sharing information for IPv6	382
IPv6 ICMP feature configuration	383
Configuring ICMP rate limiting	383
Enabling IPv6 ICMP redirect messages	384

IPv6 neighbor discovery configuration	384
IPv6 neighbor discovery configuration notes	385
Neighbor solicitation and advertisement messages	385
Router advertisement and solicitation messages	386
Neighbor redirect messages	386
Setting neighbor solicitation parameters for duplicate address detection	387
Setting IPv6 router advertisement parameters	387
Prefixes advertised in IPv6 router advertisement messages	389
Setting flags in IPv6 router advertisement messages	390
Enabling and disabling IPv6 router advertisements	390
Configuring reachable time for remote IPv6 nodes	391
IPv6 MTU	391
Configuration notes and feature limitations for IPv6 MTU	391
Changing the IPv6 MTU	392
Static neighbor entries configuration	392
Limiting the number of hops an IPv6 packet can traverse	393
IPv6 source routing security enhancements	393
TCAM space on FCX device configuration	394
Allocating TCAM space for IPv4 routing information	394
Allocating TCAM space for GRE tunnel information	394
Clearing global IPv6 information	395
Clearing the IPv6 cache	395
Clearing IPv6 neighbor information	395
Clearing IPv6 routes from the IPv6 route table	396
Clearing IPv6 traffic statistics	396
Displaying global IPv6 information	396
Displaying IPv6 cache information	397
Displaying IPv6 interface information	398
Displaying IPv6 neighbor information	400
Displaying the IPv6 route table	401
Displaying local IPv6 routers	402
Displaying IPv6 TCP information	403
Displaying IPv6 traffic statistics	407
.....	410

Chapter 9

FWS Series Switch IPv6 management

IPv6 management overview	412
IPv6 addressing	412
Enabling and disabling IPv6	413

IPv6 management features	413
IPv6 management ACLs	413
IPv6 Web management using HTTP and HTTPS	413
IPv6 logging	414
Name-to-IPv6 address resolution using IPv6 DNS server. . .	415
Defining an IPv6 DNS entry.	415
Pinging IPv6	415
SNTP over IPv6.	417
SNMP3 over IPv6.	417
Specifying an IPv6 SNMP trap receiver	417
Secure Shell, SCP, and IPv6	417
IPv6 Telnet	417
IPv6 traceroute.	419
IPv6 management commands.....	419

Chapter 10

SNMP Access

SNMP overview	421
SNMP community strings	422
Encryption of SNMP community strings.....	422
Adding an SNMP community string	422
Displaying the SNMP community strings	424
User-based security model	425
Configuring your NMS	426
Configuring SNMP version 3 on Brocade devices	426
Defining the engine id	426
Defining an SNMP group.....	427
Defining an SNMP user account.....	428
Defining SNMP views	429
SNMP version 3 traps.....	430
Defining an SNMP group and specifying which view is notified of traps	430
Defining the UDP port for SNMP v3 traps	431
Trap MIB changes	432
Specifying an IPv6 host as an SNMP trap receiver	432
SNMP v3 over IPv6	433
Specifying an IPv6 host as an SNMP trap receiver	433
Viewing IPv6 SNMP server addresses	433
Displaying SNMP Information.....	434
Displaying the Engine ID	434
Displaying SNMP groups	435
Displaying user information.....	435
Interpreting varbinds in report packets	435
SNMP v3 configuration examples	436
Simple SNMP v3 configuration.....	436
More detailed SNMP v3 configuration	436

Chapter 11	Foundry Discovery Protocol (FDP) and Cisco Discovery Protocol (CDP) Packets
	FDP Overview.....437
	FDP configuration438
	Displaying FDP information.....439
	Clearing FDP and CDP information.....442
	CDP packets443
	Enabling interception of CDP packets globally443
	Enabling interception of CDP packets on an interface443
	Displaying CDP information.....443
	Clearing CDP information445
Chapter 12	LLDP and LLDP-MED
	LLDP terms used in this chapter448
	LLDP overview449
	Benefits of LLDP450
	LLDP-MED overview450
	Benefits of LLDP-MED451
	LLDP-MED class.....452
	General LLDP operating principles452
	LLDP operating modes452
	LLDP packets453
	TLV support.....454
	MIB support.....457
	Syslog messages.....457
	LLDP configuration458
	LLDP configuration notes and considerations.....458
	Enabling and disabling LLDP.....459
	Enabling support for tagged LLDP packets459
	Changing a port LLDP operating mode459
	Maximum number of LLDP neighbors462
	Enabling LLDP SNMP notifications and Syslog messages ..462
	Changing the minimum time between LLDP transmissions .463
	Changing the interval between regular LLDP transmissions .464
	Changing the holdtime multiplier for transmit TTL464
	Changing the minimum time between port reinitializations .465
	LLDP TLVs advertised by the Brocade device465
	LLDP-MED configuration.....474
	Enabling LLDP-MED.....475
	Enabling SNMP notifications and Syslog messages for LLDP-MED topology changes.....475
	Changing the fast start repeat count476
	Defining a location id.....476
	Defining an LLDP-MED network policy483

LLDP-MED attributes advertised by the Brocade device	485
Extended power-via-MDI information	486
Displaying LLDP statistics and configuration settings.	488
LLDP configuration summary	488
Displaying LLDP statistics	489
Displaying LLDP neighbors	491
Displaying LLDP neighbors detail	492
Displaying LLDP configuration details	493
Resetting LLDP statistics	495
Clearing cached LLDP neighbor information.	495

Chapter 13

Hardware Component Monitoring

Virtual cable testing	497
Virtual cable testing configuration notes	497
Virtual cable testing command syntax	498
Viewing the results of the cable analysis	498
Digital optical monitoring	500
Digital optical monitoring configuration limitations.	500
Enabling digital optical monitoring	500
Setting the alarm interval	500
Displaying information about installed media	501
Viewing optical monitoring information	502
Syslog messages for optical transceivers	505

Chapter 14

Syslog

About Syslog messages	508
Displaying Syslog messages.	508
Enabling real-time display of Syslog messages	509
Enabling real-time display for a Telnet or SSH session	509
Displaying real-time Syslog messages	510
Syslog service configuration.	510
Displaying the Syslog configuration	510
Disabling or re-enabling Syslog.	514
Specifying a Syslog server	514
Specifying an additional Syslog server	514
Disabling logging of a message level	515
Changing the number of entries the local buffer can hold	515
Changing the log facility	516
Displaying interface names in Syslog messages.	517
Displaying TCP or UDP port numbers in Syslog messages	517
Retaining Syslog messages after a soft reboot	518
Clearing the Syslog messages from the local buffer	518
Syslog messages for hardware errors	518

Chapter 15

Network Monitoring

Basic system management	521
Viewing system information	521
Viewing configuration information	522
Viewing port statistics	523
Viewing STP statistics	525
Clearing statistics	526
Traffic counters for outbound traffic	526
Viewing egress queue counters on FCX devices	530
RMON support	531
Maximum number of entries allowed in the RMON control table	531
Statistics (RMON group 1)	532
History (RMON group 2)	534
Alarm (RMON group 3)	534
Event (RMON group 9)	535
sFlow	535
sFlow version 5	536
sFlow support for IPv6 packets	536
sFlow configuration considerations	537
Configuring and enabling sFlow	539
Enabling sFlow forwarding	545
sFlow version 5 feature configuration	546
Displaying sFlow information	549
Utilization list for an uplink port	552
Utilization list for an uplink port command syntax	553
Displaying utilization percentages for an uplink	553

Chapter 16

Basic Layer 2 Features

About port regions	556
FastIron X Series device port regions	556
FCX and FWS device port regions	557
ICX 6610 device port regions	557
ICX 6430 device port regions	557
ICX 6450 device port regions	558
Enabling or disabling the Spanning Tree Protocol (STP)	558
Modifying STP bridge and port parameters	558
Management MAC address for stackable devices	559
MAC learning rate control	559
Changing the MAC age time and disabling MAC address learning	560
Disabling the automatic learning of MAC addresses	560
Displaying the MAC address table	561
Static MAC entry configuration	561
Multi-port static MAC address	561
VLAN-based static MAC entries configuration	563

Clearing MAC address entries	563
Flow-based MAC address learning	564
Flow-based learning overview	564
Flow-based learning configuration considerations	565
Configuring flow-based MAC address learning	566
Displaying information about flow-based MACs	567
Clearing flow-based MAC address entries	567
Enabling port-based VLANs	567
Assigning IEEE 802.1Q tagging to a port	568
Defining MAC address filters	569
MAC address filters configuration notes and limitations	569
MAC address filters command syntax	570
Enabling logging of management traffic permitted by MAC address filters	571
MAC address filter override for 802.1X-enabled ports	572
Locking a port to restrict addresses	574
Lock address configuration notes	574
Lock address command syntax	574
Monitoring MAC address movement	574
Configuring the MAC address movement threshold rate	575
Viewing the MAC address movement threshold rate configuration	576
Configuring an interval for collecting MAC address move notifications	577
Viewing MAC address movement statistics for the interval history	577
Displaying and modifying system parameter default settings	578
System default settings configuration considerations	578
Displaying system parameter default values	578
Modifying system parameter default values	584
Dynamic buffer allocation for QoS priorities for FastIron X Series devices	584
Default queue depth limits for FastIron X Series devices	585
Configuring the total transmit queue depth limit for FastIron X Series devices	585
Configuring the transmit queue depth limit for a given traffic class on FastIron X Series devices	586
Removing buffer allocation limits on FastIron X Series devices	587
Buffer profile configuration	587
Default queue depth limits for FastIron X Series devices	589
Dynamic buffer allocation for FCX, FWS, and ICX devices	589
Configuring buffer profiles	590
Configuring buffer sharing on FCX and ICX devices	599
Removing buffer allocation limits on FCX, FWS, and ICX	602
Buffer profiles for VoIP on FastIron stackable devices	603
Remote Fault Notification on 1Gbps fiber connections	603
Enabling and disabling remote fault notification	604

Link Fault Signaling for 10Gbps Ethernet devices	604
Jumbo frame support	606

Chapter 17

Metro Features

Topology groups	607
Master VLAN and member VLANs	608
Control ports and free ports	608
Topology group configuration considerations	608
Configuring a topology group	609
Displaying topology group information	610
Metro Ring Protocol	611
Metro Ring Protocol configuration notes	613
MRP rings without shared interfaces (MRP Phase 1)	613
MRP rings with shared interfaces (MRP Phase 2)	614
Ring initialization	616
How ring breaks are detected and healed	621
Master VLANs and customer VLANs	623
Metro Ring Protocol configuration	625
Metro Ring Protocol diagnostics	628
Displaying MRP information	629
MRP CLI example	631
VSRP	633
VSRP configuration notes and feature limitations	634
Layer 2 and Layer 3 redundancy	635
Master election and failover	635
VSRP-aware security features	639
VSRP parameters	640
Configuring basic VSRP parameters	642
Configuring optional VSRP parameters	643
Displaying VSRP information	651
VSRP fast start	654
VSRP and MRP signaling	656

Chapter 18

Power over Ethernet

Power over Ethernet overview	659
Power over Ethernet terms used in this chapter	660
Methods for delivering Power over Ethernet	660
PoE autodiscovery	662
Power class	662
Power specifications	663
Dynamic upgrade of PoE power supplies	663
Power over Ethernet cabling requirements	665
Supported powered devices	665
Installing PoE firmware	666
PoE and CPU utilization	668
Enabling and disabling Power over Ethernet	669
Disabling support for PoE legacy power-consuming devices	669

Enabling the detection of PoE power requirements advertised through CDP	670
Command syntax for PoE power requirements	670
Setting the maximum power level for a PoE power-consuming device	670
Setting power levels configuration note	671
Configuring power levels command syntax	671
Setting the power class for a PoE power-consuming device	672
Setting the power class configuration notes	672
Setting the power class command syntax	672
Setting the power budget for a PoE interface module	673
Setting the inline power priority for a PoE port	673
Command syntax for setting the inline power priority for a PoE port	674
Resetting PoE parameters	675
Displaying Power over Ethernet information	675
Displaying PoE operational status	675
Displaying detailed information about PoE power supplies ..	678

Chapter 19

UDLD and Protected Link Groups

UDLD overview	685
UDLD for tagged ports	686
Configuration notes and feature limitations for UDLD	686
Enabling UDLD	686
Enabling UDLD for tagged ports	687
Changing the Keepalive interval	688
Changing the Keepalive retries	688
Displaying UDLD information	688
Clearing UDLD statistics	690
Protected link groups	691
About active ports	691
Using UDLD with protected link groups	691
UDLD with protected link groups configuration notes	691
Creating a protected link group and assigning an active port	692

Chapter 20

Trunk Groups and Dynamic Link Aggregation

Trunk group overview	697
Trunk group connectivity to a server	698
Trunk group rules	699
Trunk group configuration examples	702
Support for flexible trunk group membership	704
Trunk group load sharing	705

Configuring a trunk group	707
CLI syntax for configuring consecutive ports in a trunk group	707
CLI syntax for configuring non-consecutive ports in a trunk group	708
Example 1: Configuring the trunk groups.	709
Example 2: Configuring a trunk group that spans two Ethernet modules in a chassis device.	709
Example 3: Configuring a multi-slot trunk group with one port per module	710
Example 4: Configuring a trunk group of 10 Gbps Ethernet ports	710
Additional trunking options	711
Displaying trunk group configuration information	716
Viewing the first and last ports in a trunk group	717
Dynamic link aggregation	718
IronStack LACP trunk group configuration example	719
Examples of valid LACP trunk groups	719
Configuration notes and limitations for configuring IronStack LACP trunk groups.	721
Adaptation to trunk disappearance	722
Flexible trunk eligibility	723
Enabling dynamic link aggregation.	724
How changing the VLAN membership of a port affects trunk groups and dynamic keys	725
Additional trunking options for LACP trunk ports.	725
Link aggregation parameters	725
Displaying and determining the status of aggregate links.	731
Events that affect the status of ports in an aggregate link.	731
Displaying link aggregation and port status information	731
Displaying LACP status information	734
Clearing the negotiated aggregate links table	734
Single instance LACP configuration	734
Configuration notes for single link LACP	735
CLI syntax for single link LACP	735

Chapter 21

VLANs

VLAN overview	738
Types of VLANs.	738
Modifying a port-based VLAN	744
Default VLAN	754
802.1Q tagging	755
Spanning Tree Protocol (STP)	758
Virtual routing interfaces.	759
VLAN and virtual routing interface groups.	760
Dynamic, static, and excluded port membership	761
Super aggregated VLANs.	763
Trunk group ports and VLAN membership.	763
Summary of VLAN configuration rules	764

Routing between VLANs	765
Virtual routing interfaces (Layer 2 switches only)	765
Routing between VLANs using virtual routing interfaces (Layer 3 switches only)	765
Dynamic port assignment (Layer 2 switches and Layer 3 switches)	766
Assigning a different VLAN ID to the default VLAN	766
Assigning different VLAN IDs to reserved VLANs 4091 and 4092	767
Assigning trunk group ports	768
Enable spanning tree on a VLAN	768
Configuring IP subnet, IPX network and protocol-based VLANs	770
IP subnet, IPX network, and protocol-based VLAN configuration example	770
IP subnet, IPX network, and protocol-based VLANs within port-based VLANs	772
IPv6 protocol VLAN configuration	776
Routing between VLANs using virtual routing interfaces (Layer 3 switches only)	776
Configuring protocol VLANs with dynamic ports	782
Aging of dynamic ports	783
Configuration guidelines for membership aging of dynamic VLAN ports	784
Configuring an IP, IPX, or AppleTalk Protocol VLAN with Dynamic Ports	784
Configuring an IP subnet VLAN with dynamic ports	785
Configuring an IPX network VLAN with dynamic ports	785
Configuring uplink ports within a port-based VLAN	787
Configuration considerations for uplink ports within a port-based VLAN	787
Configuration syntax for uplink ports within a port-based VLAN	787
IP subnet address on multiple port-based VLAN configuration	788
VLAN groups and virtual routing interface group	791
Configuring a VLAN group	791
Configuring a virtual routing interface group	793
Displaying the VLAN group and virtual routing interface group information	794
Allocating memory for more VLANs or virtual routing interfaces	795
Super-aggregated VLAN configuration	796
Configuration notes for aggregated VLANs	799
Configuring aggregated VLANs	799
Verifying the aggregated VLAN configuration	801
Complete CLI examples for aggregated VLANs	801

802.1ad tagging configuration	804
Configuration rules for 802.1ad tagging	805
Enabling 802.1ad tagging	805
Example 802.1ad configuration	806
Configuring 802.1ad tag profiles	808
Private VLAN configuration	809
Configuration notes for PVLANS and standard VLANs	813
Enabling broadcast or unknown unicast traffic to the PVLAN	816
CLI example for a general PVLAN network	816
CLI example for a PVLAN network with switch-switch link ports	817
Dual-mode VLAN ports	818
Displaying VLAN information	821
Displaying VLANs in alphanumeric order	821
Displaying system-wide VLAN information	822
Displaying global VLAN information	823
Displaying VLAN information for specific ports	823
Displaying a port VLAN membership	824
Displaying a port dual-mode VLAN membership	824
Displaying port default VLAN IDs (PVIDs)	825
Displaying PVLAN information	825

Chapter 22

Multi-Chassis Trunking

Multi-Chassis Trunking Overview	827
How MCT works	828
MCT terminology	829
MCT data flow	830
MCT and VLANs	834
Cluster client automatic configuration	835
MCT feature interaction	835
Basic MCT configuration	837
MCT configuration considerations	837
Differences in configuring MCT for the switch and router image	838
Configuring MCT	838
Setting up cluster client automatic configuration	841
MCT failover scenarios	843
Layer 2 behavior with MCT	846
MAC operations	846
Port loop detection	849
MCT Layer 2 protocols	850
Protocol-based VLANs	850
Uplink switch	851
Layer 2 multicast snooping over MCT	851
Layer 3 behavior with MCT	854
Layer 3 unicast over MCT	855
MCT for VRRP or VRRP-E	857

Displaying MCT information	860
Displaying peer and client states	860
Displaying state machine information	861
Displaying cluster, peer, and client states	862
Displaying information about Ethernet interfaces.....	862
Displaying STP information	864
Displaying information for multicast snooping	864
MCT configuration examples	867
Single-level MCT example	867
Two-level MCT example	871
MCT configuration with VRRP-E example.....	876
Multicast snooping configuration example	879

Chapter 23

GVRP

GVRP overview.....	883
GVRP application examples	884
Dynamic core and fixed edge	885
Dynamic core and dynamic edge	886
Fixed core and dynamic edge	886
Fixed core and fixed edge	886
VLAN names created by GVRP	886
Configuration notes for GVRP.....	886
GVRP configuration	888
Changing the GVRP base VLAN ID	888
Increasing the maximum configurable value of the Leaveall timer	888
Enabling GVRP	889
Disabling VLAN advertising.....	889
Disabling VLAN learning	890
Changing the GVRP timers	890
Converting a VLAN created by GVRP into a statically-configured VLAN	892
Displaying GVRP information	893
Displaying GVRP configuration information	894
Displaying GVRP VLAN information.....	896
Displaying GVRP statistics.....	898
Displaying CPU utilization statistics	901
Clearing GVRP statistics	902
GVRP CLI examples	902
Dynamic core and fixed edge	902
Dynamic core and dynamic edge	903
Fixed core and dynamic edge	904
Fixed core and fixed edge	904

Chapter 24

MAC-based VLANs

MAC-based VLAN overview	905
Static and dynamic hosts	905
MAC-based VLAN feature structure	906
Dynamic MAC-based VLAN	907
Configuration notes and feature limitations for dynamic MAC-based VLAN	907
Dynamic MAC-based VLAN CLI commands	907
Dynamic MAC-based VLAN configuration example	908
MAC-based VLAN configuration	909
Using MAC-based VLANs and 802.1X security on the same port	910
Configuring generic and Brocade vendor-specific attributes on the RADIUS server	910
Aging for MAC-based VLAN	911
Disabling aging for MAC-based VLAN sessions	912
Configuring the maximum MAC addresses per port	913
Configuring a MAC-based VLAN for a static host	913
Configuring MAC-based VLAN for a dynamic host	914
Configuring dynamic MAC-based VLAN	914
Configuring MAC-based VLANs using SNMP	915
Displaying Information about MAC-based VLANs	915
Displaying the MAC-VLAN table	915
Displaying the MAC-VLAN table for a specific MAC address	916
Displaying allowed MAC addresses	916
Displaying denied MAC addresses	917
Displaying detailed MAC-VLAN data	918
Displaying MAC-VLAN information for a specific interface	919
Displaying MAC addresses in a MAC-based VLAN	920
Displaying MAC-based VLAN logging	921
Clearing MAC-VLAN information	921
Sample MAC-based VLAN application	921

Chapter 25

Port mirroring and Monitoring

Port mirroring and monitoring overview	925
Port mirroring and monitoring configuration	927
Configuration notes for port mirroring and monitoring	927
Command syntax for port mirroring and monitoring	929
mirroring configuration on an IronStack	930
Configuration notes for Ironstack mirroring	930
ACL-based inbound mirroring	931
Creating an ACL-based inbound mirror clause for FWS and FCX devices	931
ACL-based inbound mirror clauses for FastIron X Series devices	932
Destination mirror port	932

MAC address filter-based mirroring	936
Configuring MAC address filter-based mirroring	936
VLAN-based mirroring	937
VLAN-based mirroring on FastIron X Series devices	939

Chapter 26

IP Configuration

Basic IP configuration	946
IP configuration overview	947
Edge Layer 3 support	947
Full Layer 3 support	947
IP interfaces	948
IP packet flow through a Layer 3 Switch	949
IP route exchange protocols	954
IP multicast protocols	954
IP interface redundancy protocols	954
ACLs and IP access policies	955
Basic IP parameters and defaults – Layer 3 Switches	955
When parameter changes take effect	956
IP global parameters – Layer 3 Switches	956
IP interface parameters – Layer 3 Switches	959
Basic IP parameters and defaults – Layer 2 Switches	962
IP global parameters – Layer 2 Switches	962
Interface IP parameters – Layer 2 Switches	964
Configuring IP parameters – Layer 3 Switches	964
Configuring IP addresses	964
Configuring 31-bit subnet masks on point-to-point networks	968
Configuring DNS resolver	970
Configuring packet parameters	973
Changing the router ID	976
Specifying a single source interface for specified packet types	977
ARP parameter configuration	981
Configuring forwarding parameters	986
Disabling ICMP messages	989
Disabling ICMP Redirect Messages	990
Static routes configuration	991
Configuring a default network route	1000
Configuring IP load sharing	1001
ICMP Router Discovery Protocol configuration	1004
IRDP parameters	1005
Reverse Address Resolution Protocol configuration	1007
Configuring UDP broadcast and IP helper parameters	1009
BootP and DHCP relay parameter configuration	1011
DHCP Server	1013
Displaying DHCP Server information	1024
DHCP Client-Based Auto-Configuration and Flash image update	1028

Configuring IP parameters – Layer 2 Switches.....	1037
Configuring the management IP address and specifying the default gateway	1037
Configuring Domain Name Server (DNS) resolver.....	1038
Changing the TTL threshold	1039
DHCP Assist configuration.....	1040
IPv4 point-to-point GRE tunnels.....	1043
IPv4 GRE tunnel overview	1044
GRE packet structure and header format	1044
Path MTU Discovery (PMTUD) support.....	1045
Configuration considerations for PMTUD support.....	1045
Tunnel loopback ports for GRE tunnels	1046
Support for IPv4 multicast routing over GRE tunnels	1046
GRE support with other features	1047
Configuration considerations for GRE IP tunnels	1048
Configuration tasks for GRE tunnels	1050
Example point-to-point GRE tunnel configuration	1058
Displaying GRE tunneling information	1059
Clearing GRE statistics	1064
Displaying IP configuration information and statistics.....	1064
Changing the network mask display to prefix format	1065
Displaying IP information – Layer 3 Switches	1065
Displaying IP information – Layer 2 Switches	1080
Disabling IP checksum check.....	1084

Chapter 27

Spanning Tree Protocol

STP overview	1087
Standard STP parameter configuration.....	1088
STP parameters and defaults	1088
Enabling or disabling the Spanning Tree Protocol (STP) ...	1089
Changing STP bridge and port parameters	1091
STP protection enhancement	1092
Displaying STP information	1094
STP feature configuration.....	1103
Fast port span	1103
Fast Uplink Span	1106
802.1W Rapid Spanning Tree (RSTP).....	1109
802.1W Draft 3	1146
Single Spanning Tree (SSTP).....	1151
STP per VLAN group.....	1153
PVST/PVST+ compatibility	1157
Overview of PVST and PVST+	1157
VLAN tags and dual mode.....	1158
Configuring PVST+ support.....	1159
Displaying PVST+ support information.....	1160
PVST+ configuration examples	1160
PVRST compatibility	1163

BPDU guard	1163
Enabling BPDU protection by port	1163
Re-enabling ports disabled by BPDU guard	1164
Displaying the BPDU guard status	1164
BPDU guard status example console messages	1165
Root guard	1166
Enabling STP root guard	1166
Displaying the STP root guard	1166
Displaying the root guard by VLAN	1167
Error disable recovery	1167
Enabling error disable recovery	1167
Setting the recovery interval	1168
Displaying the error disable recovery state by interface ...	1168
Displaying the recovery state for all conditions	1168
Displaying the recovery state by port number and cause..	1169
Errdisable Syslog messages	1169
802.1s Multiple Spanning Tree Protocol	1169
Multiple spanning-tree regions	1170
Configuration notes	1171
Configuring MSTP mode and scope	1171
Reduced occurrences of MSTP reconvergence	1172
Configuring additional MSTP parameters	1174

Chapter 28

Base Layer 3 and Routing Protocols

TCAM entries in FWS devices	1186
Adding a static IP route	1186
Adding a static ARP entry	1187
Modifying and displaying Layer 3 system parameter limits ...	1188
Layer 3 configuration notes	1188
FWS with base Layer 3	1188
FastIron first generation modules	1189
FastIron second generation modules	1190
FastIron third generation modules	1191
Displaying Layer 3 system parameter limits	1191
Configuring RIP	1193
Enabling RIP	1193
Enabling redistribution of IP static routes into RIP	1193
Configuring a redistribution filter	1195
Enabling redistribution	1195
Enabling learning of default routes	1196
Changing the route loop prevention method	1196
Other Layer 3 protocols	1196
Enabling or disabling routing protocols	1196
Enabling or disabling Layer 2 switching	1197
Configuration notes and feature limitations for	
Layer 2 switching	1197
Command syntax for Layer 2 switching	1197

Chapter 29

RIP (IPv4)

RIP overview	1199
ICMP host unreachable message for undeliverable ARPs ..	1200
RIP parameters and defaults	1200
RIP global parameters	1200
RIP interface parameters	1201
RIP parameter configuration	1202
Enabling RIP	1202
Enabling ECMP for routes in RIP	1203
Configuring metric parameters	1203
Changing the administrative distance	1204
Configuring redistribution	1205
Route learning and advertising parameters	1207
Denying route advertisements for connected routes	1208
Changing the route loop prevention method	1209
Suppressing RIP route advertisement on a VRRP or VRRP-E backup interface	1209
Configuring RIP route filters	1210
Displaying RIP filters	1211
Displaying CPU utilization statistics	1212

Chapter 30

RIP (IPv6)

RIPng overview	1215
Summary of configuration tasks	1216
RIPng configuration	1216
RIPng timers	1217
Route learning and advertising parameters	1218
Configuring default route learning and advertising	1218
Advertising IPv6 address summaries	1218
Changing the metric of routes learned and advertised on an interface	1219
Redistributing routes into RIPng	1220
Controlling distribution of routes through RIPng	1220
Configuring poison reverse parameters	1221
Clearing RIPng routes from the IPv6 route table	1221
Displaying the RIPng configuration	1222
Displaying RIPng routing table	1223

Chapter 31

OSPF version 2 (IPv4)

OSPF overview	1226
OSPF point-to-point links	1228
Designated routers in multi-access networks	1228
Designated router election in multi-access networks	1228
OSPF RFC 1583 and 2178 compliance	1230
Reduction of equivalent AS External LSAs	1230
Support for OSPF RFC 2328 Appendix E	1232
Dynamic OSPF activation and configuration	1233
Dynamic OSPF memory	1234
OSPF graceful restart	1234
Configuring OSPF	1235
OSPF configuration rules	1235
OSPF parameters	1235
Enabling OSPF on the router	1236
Assigning OSPF areas	1237
Assigning an area range (optional)	1241
Assigning interfaces to an area	1242
Modifying interface defaults	1242
Changing the timer for OSPF authentication changes	1245
Block flooding of outbound LSAs on specific	
OSPF interfaces	1245
Configuring an OSPF non-broadcast interface	1246
Assigning virtual links	1247
Modifying virtual link parameters	1249
Changing the reference bandwidth for the cost	
on OSPF interfaces	1251
Defining redistribution filters	1252
Preventing specific OSPF routes from being installed	
in the IP route table	1255
Modifying the default metric for redistribution	1258
Enabling route redistribution	1258
Disabling or re-enabling load sharing	1260
Configuring external route summarization	1262
Configuring default route origination	1263
Modifying SPF timers	1264
Modifying the redistribution metric type	1265
Administrative distance	1265
Configuring OSPF group Link State Advertisement	
(LSA) pacing	1266
Modifying OSPF traps generated	1266
Specifying the types of OSPF Syslog messages to log	1267
Modifying the OSPF standard compliance setting	1268
Modifying the exit overflow interval	1268
Configuring an OSPF point-to-point link	1269
Configuring OSPF graceful restart	1269

Clearing OSPF information	1270
Clearing OSPF neighbor information	1270
Clearing OSPF topology information	1271
Clearing redistributed routes from the OSPF routing table ..	1271
Clearing information for OSPF areas	1271
Displaying OSPF information	1272
Displaying general OSPF configuration information	1272
Displaying CPU utilization statistics	1273
Displaying OSPF area information	1275
Displaying OSPF neighbor information	1275
Displaying OSPF interface information	1278
Displaying OSPF route information	1279
Displaying OSPF external link state information	1281
Displaying OSPF link state information	1282
Displaying the data in an LSA	1283
Displaying OSPF virtual neighbor information	1283
Displaying OSPF virtual link information	1284
Displaying OSPF ABR and ASBR information	1284
Displaying OSPF trap status	1284
Displaying OSPF graceful restart information	1285

Chapter 32

OSPF version 3 (IPv6)

OSPF (IPv6) overview	1287
Differences between OSPF V2 and OSPF V3	1288
Link state advertisement types for OSPF V3	1288
OSPF V3 configuration	1289
Enabling OSPF V3	1289
Assigning OSPF V3 areas	1290
Assigning interfaces to an area	1291
Configuring virtual links	1291
Changing the reference bandwidth for the cost on	
OSPF V3 interfaces	1294
Redistributing routes into OSPF V3	1295
External route summarization	1298
Filtering OSPF V3 routes	1299
Default route origination	1302
Shortest path first timers	1303
Administrative distance	1304
Configuring the OSPF V3 LSA pacing interval	1305
Modifying exit overflow interval	1305
Modifying external link state database limit	1305
Modifying OSPF V3 interface defaults	1306
Disabling or re-enabling event logging	1307
IPSec for OSPFv3	1307
IPsec for OSPFv3 configuration	1308

Displaying OSPF V3 Information	1314
Displaying OSPF V3 area information.....	1315
Displaying OSPF V3 database information.....	1316
Displaying OSPF V3 interface information.....	1321
Displaying OSPF V3 memory usage	1324
Displaying OSPF V3 neighbor information	1325
Displaying routes redistributed into OSPF V3	1327
Displaying OSPF V3 route information	1328
Displaying OSPF V3 SPF information	1330
Displaying IPv6 OSPF virtual link information	1333
Displaying OSPF V3 virtual neighbor information	1333
IPsec examples	1334

Chapter 33

BGP (IPv4)

BGP4 overview	1342
Relationship between the BGP4 route table and the IP route table	1343
How BGP4 selects a path for a route	1344
BGP4 message types.....	1345
BGP4 graceful restart.....	1347
Basic configuration and activation for BGP4	1348
Note regarding disabling BGP4.....	1348
BGP4 parameters	1349
BGP4 parameter changes.....	1350
BGP4 memory considerations	1351
Memory configuration options obsoleted by dynamic memory	1352
Basic configuration tasks required for BGP4	1352
Enabling BGP4 on the router	1352
Changing the router ID.....	1352
Setting the local AS number	1353
Adding a loopback interface	1353
Adding BGP4 neighbors.....	1354
Adding a BGP4 peer group	1361

Optional BGP4 configuration tasks	1365
Changing the Keep Alive Time and Hold Time	1365
Changing the BGP4 next-hop update timer	1365
Enabling fast external fallover.	1366
Changing the maximum number of paths for BGP4 load sharing.	1366
Customizing BGP4 load sharing	1368
Specifying a list of networks to advertise.	1368
Changing the default local preference	1370
Using the IP default route as a valid next hop for a BGP4 route	1370
Advertising the default route.	1371
Changing the default MED (Metric) used for route redistribution	1371
Enabling next-hop recursion	1371
Changing administrative distances	1374
Requiring the first AS to be the neighbor AS	1376
Disabling or re-enabling comparison of the AS-Path length	1376
Enabling or disabling comparison of the router IDs	1376
Configuring the Layer 3 switch to always compare Multi-Exit Discriminators (MEDs)	1377
Treating missing MEDs as the worst MEDs	1377
Route reflection parameter configuration	1378
Configuration notes for BGP4 autonomous systems	1381
Aggregating routes advertised to BGP4 neighbors	1384
Configuring BGP4 graceful restart.	1385
Configuring BGP4 graceful restart	1385
Configuring timers for BGP4 graceful restart (optional) ...	1385
BGP null0 routing	1386
Configuration steps for BGP null 0 routing.	1387
Configuration examples for BGP null 0 routing	1388
Show commands for BGP null 0 routing.	1389
Modifying redistribution parameters	1391
Redistributing connected routes.	1391
Redistributing RIP routes.	1392
Redistributing OSPF external routes.	1392
Redistributing static routes.	1393
Disabling or re-enabling re-advertisement of all learned BGP4 routes to all BGP4 neighbors	1393
Redistributing IBGP routes into RIP and OSPF.	1393
Filtering	1394
Specific IP address filtering.	1394
AS-path filtering	1395
BGP4 filtering communities	1399
Defining IP prefix lists	1401
Defining neighbor distribute lists	1402
Defining route maps	1403
Using a table map to set the rag value.	1411
Configuring cooperative BGP4 route filtering.	1411

Route flap dampening configuration	1414
Globally configuring route flap dampening	1415
Using a route map to configure route flap dampening for specific routes	1416
Using a route map to configure route flap dampening for a specific neighbor.	1417
Removing route dampening from a route.	1418
Removing route dampening from a neighbor routes suppressed due to aggregation	1418
Displaying and clearing route flap dampening statistics ..	1419
Generating traps for BGP	1421
Displaying BGP4 information	1421
Displaying summary BGP4 information	1422
Displaying the active BGP4 configuration	1425
Displaying CPU utilization statistics	1425
Displaying summary neighbor information	1427
Displaying BGP4 neighbor information.	1428
Displaying peer group information	1439
Displaying summary route information	1440
Displaying the BGP4 route table.	1441
Displaying BGP4 route-attribute entries.	1447
Displaying the routes BGP4 has placed in the IP route table	1448
Displaying route flap dampening statistics	1449
Displaying the active route map configuration	1450
Displaying BGP4 graceful restart neighbor information ...	1451
Updating route information and resetting a neighbor session ..	1451
Using soft reconfiguration.	1452
Dynamically requesting a route refresh from a BGP4 neighbor	1454
Closing or resetting a neighbor session.	1457
Clearing and resetting BGP4 routes in the IP route table ..	1458
Clearing traffic counters.	1458
Clearing route flap dampening statistics.	1459
Removing route flap dampening	1459
Clearing diagnostic buffers.	1460

IGMP snooping overview	1461
MAC-based implementation on	
FastIron X Series devices	1462
Queriers and non-queriers	1463
VLAN-specific configuration	1463
Tracking and fast leave	1463
Support for IGMP snooping and Layer 3 multicast routing	
together on the same device	1464
Configuration notes and feature limitations	
for IGMP snooping and Layer 3 multicast routing	1464
PIM SM traffic snooping overview	1465
Application examples of PIM SM traffic snooping	1465
Configuration notes and limitations for	
PIM SM snooping	1466
IGMP snooping configuration	1467
Enabling IGMP snooping globally on the device	1469
Configuring the IGMP mode	1469
Configuring the IGMP version	1470
Disabling IGMP snooping on a VLAN	1471
Disabling transmission and receipt of IGMP packets	
on a port	1471
Modifying the age interval for group membership entries	1471
Modifying the query interval (active IGMP snooping	
mode only)	1472
Modifying the maximum response time	1472
Configuring report control	1472
Modifying the wait time before stopping traffic when	
receiving a leave message	1473
Modifying the multicast cache age time	1473
Enabling or disabling error and warning messages	1473
Configuring static router ports	1473
Turning off static group proxy	1474
Enabling IGMP V3 membership tracking and	
fast leave for the VLAN	1474
Enabling fast leave for IGMP V2	1475
Enabling fast convergence	1475
PIM SM snooping configuration	1476
Enabling or disabling PIM SM snooping	1476
Enabling PIM SM snooping on a VLAN	1476
Disabling PIM SM snooping on a VLAN	1477
IGMP snooping show commands	1477
Displaying the IGMP snooping configuration	1477
Displaying IGMP snooping errors	1478
Displaying IGMP group information	1478
Displaying IGMP snooping mcache information	1480
Displaying software resource usage for VLANs	1480
Displaying the status of IGMP snooping traffic	1481
Displaying querier information	1482

	PIM SM snooping show commands	1486
	Displaying PIM SM snooping information.	1486
	Displaying PIM SM snooping information on a Layer 2 switch	1486
	Displaying PIM SM snooping information for a specific group or source group pair	1488
	Clear commands for IGMP snooping	1489
	Clearing the IGMP mcache	1489
	Clearing the mcache on a specific VLAN	1489
	Clearing traffic on a specific VLAN	1489
	Clearing IGMP counters on VLANs	1489
Chapter 35	IP Multicast Traffic Reduction for FastIron WS and Brocade FCX and ICX Switches	
	IGMP snooping overview.	1491
	Configuration notes for IGMP snooping	1493
	Querier and non-querier configuration	1494
	VLAN specific configuration	1495
	IGMPv2 with IGMPv3	1495
	PIM SM traffic snooping overview	1495
	Application example of PIM SM traffic snooping.	1495
	IGMP snooping configuration	1497
	Displaying IGMP snooping information	1505
	Displaying querier information	1510
	Clear IGMP snooping commands	1514
Chapter 36	IP Multicast Protocols	
	IP multicast overview	1516
	IPv4 multicast group addresses	1516
	Mapping of IPv4 Multicast group addresses to Ethernet MAC addresses.	1516
	Supported Layer 3 multicast routing protocols	1517
	Suppression of unregistered multicast packets	1517
	Multicast terms	1517
	Global IP multicast parameters	1517
	Changing dynamic memory allocation for IP multicast groups	1518
	Changing IGMP V1 and V2 parameters	1519

PIM Dense	1522
Initiating PIM multicasts on a network	1522
Pruning a multicast tree	1522
Grafts to a multicast Tree	1524
PIM DM versions	1524
PIM DM configuration	1525
Failover time in a multi-path topology	1529
Modifying the TTL	1529
Displaying basic PIM Dense configuration information	1530
Displaying all multicast cache entries in a pruned state	1531
PIM Sparse	1531
PIM Sparse switch types	1532
RP paths and SPT paths	1533
PIM Sparse configuration	1533
ACL based RP assignment	1539
Anycast RP	1540
Displaying PIM Sparse configuration information and statistics	1545
PIM Passive	1558
Multicast Source Discovery Protocol (MSDP)	1558
Peer Reverse Path Forwarding (RPF) flooding	1561
Source active caching	1561
MSDP configuration	1562
Designating an interface IP address as the RP IP address	1563
Filtering MSDP source-group pairs	1563
MSDP mesh groups	1567
Displaying MSDP information	1573
Clearing MSDP information	1578
Passive multicast route insertion	1579
DVMRP overview	1580
Initiating DVMRP multicasts on a network	1580
Pruning a multicast tree	1581
Grafts to a multicast tree	1582
DVMRP configuration on the Layer 3 switch and interface	1582
Modifying DVMRP global parameters	1583
Modifying DVMRP interface parameters	1586
Displaying information about an upstream neighbor device	1587
IP tunnel configuration	1587
Using ACLs to control multicast features	1588
Using ACLs to limit static RP groups	1588
Using ACLs to limit PIM RP candidate advertisement	1590

Disabling CPU processing for select multicast groups	1591
CLI command syntax to disable CPU processing	1592
Viewing disabled multicast addresses	1592
Configuring a static multicast route	1593
Displaying the multicast configuration for another multicast router	1595
IGMP V3	1596
Default IGMP version	1597
Compatibility with IGMP V1 and V2	1597
Globally enabling the IGMP version	1597
Enabling the IGMP version per interface setting	1598
Enabling the IGMP version on a physical port within a virtual routing interface	1598
Enabling membership tracking and fast leave	1598
Setting the query interval	1599
Setting the group membership time	1599
Setting the maximum response time	1600
IGMP V3 and source-specific multicast protocols	1600
Displaying IGMP V3 information on Layer 3 switches	1600
Clearing IGMP statistics	1604
IGMP Proxy	1605
IGMP proxy configuration notes	1605
IGMP proxy limitations	1605
Configuring IGMP Proxy	1605
Displaying IGMP Proxy traffic	1606
IP multicast protocols and IGMP snooping on the same device	1606
IP multicast protocols and IGMP snooping configuration example	1607
IP multicast protocols and IGMP snooping CLI commands	1608

Chapter 37

MLD Snooping on FastIron X Series Switches

MLD Snooping Overview	1611
How MLD snooping uses MAC addresses to forward multicast packets	1612
MLD snooping configuration notes and feature limitations	1613
MLD snooping-enabled queriers and non-queriers	1614
MLD and VLAN configuration	1615
MLDv1 with MLDv2	1615

MLD snooping configuration	1615
Configuring the hardware and software resource limits...	1616
Disabling transmission and receipt of MLD packets on a port	1616
Configuring the global MLD mode	1617
Modifying the age interval	1617
Modifying the query interval (active MLD snooping mode only)	1617
Configuring the global MLD version	1618
Configuring report control	1618
Modifying the wait time before stopping traffic when receiving a leave message	1618
Modifying the multicast cache (mcache) aging time	1619
Disabling error and warning messages	1619
Configuring the MLD mode for a VLAN	1619
Disabling MLD snooping for the VLAN	1620
Configuring the MLD version for the VLAN	1620
Configuring the MLD version for individual ports	1620
Configuring static groups to the entire VLAN or to individual ports	1620
Configuring static router ports	1621
Disabling static group proxy	1621
Enabling MLDv2 membership tracking and fast leave for the VLAN	1621
Configuring fast leave for MLDv1	1622
Enabling fast convergence	1622
Displaying MLD snooping information	1623
Clearing MLD snooping counters and mcache	1628

Chapter 38

MLD Snooping on FastIron WS and Brocade FCX and ICX Switches

MLD snooping overview	1631
Configuration notes for MLD snooping	1633
Configuring queriers and non-queriers on MLD snooping-enabled devices	1634
MLD snooping configuration on VLANs	1634
MLD snooping using MLDv1 with MLDv2	1635

MLD snooping configuration	1635
Global tasks for MLD Snooping	1635
VLAN-specific tasks for MLD Snooping	1635
Configuring the hardware and software resource limits...	1636
Disabling transmission and receipt of MLD packets on a port	1636
Configuring the global MLD mode	1636
Modifying the age interval	1637
Modifying the query interval (Active MLD snooping mode only)	1637
Configuring the global MLD version	1637
Configuring report control	1637
Modifying the wait time before stopping traffic when receiving a leave message	1638
Modifying the multicast cache (mcache) aging time	1638
Disabling error and warning messages	1638
Configuring the MLD mode for a VLAN	1639
Disabling MLD snooping for the VLAN	1639
Configuring the MLD version for the VLAN	1639
Configuring the MLD version for individual ports	1639
Configuring static groups to the entire VLAN or to individual ports	1640
Configuring static router ports	1640
Turning off static group proxy	1640
Enabling MLDv2 membership tracking and fast leave for the VLAN	1641
Configuring fast leave for MLDv1	1641
Enabling fast convergence	1642
Displaying MLD snooping information	1642
Clear MLD snooping commands	1647

Chapter 39

VRRP and VRRP-E

VRRP and VRRP-E overview	1650
VRRP overview	1650
VRRP-E overview	1655
ARP behavior with VRRP-E	1658
Comparison of VRRP and VRRP-E	1658
VRRP	1658
VRRP-E	1658
Architectural differences between VRRP and VRRP-E	1659
VRRP and VRRP-E parameters	1660
Note regarding disabling VRRP or VRRP-E	1663
Basic VRRP parameter configuration	1664
Configuration rules for VRRP	1664
Configuring the Owner for IPv4 VRRP	1664
Configuring the Owner for IPv6 VRRP	1665
Configuring a Backup for IPv4 VRRP	1666
Configuring a Backup for IPv6 VRRP	1666
Configuration considerations for IPv6 VRRP v3 and IPv6 VRRP-E v3 support on Brocade devices	1667

Basic VRRP-E parameter configuration	1668
Configuration rules for VRRP-E	1668
Configuring IPv4 VRRP-E	1668
Configuring IPv6 VRRP-E	1669
Additional VRRP and VRRP-E parameter configuration	1670
VRRP and VRRP-E authentication types	1671
VRRP router type	1673
Suppression of RIP advertisements	1674
Hello interval configuration	1675
Dead interval configuration	1676
Backup Hello message state and interval	1676
Track port configuration	1677
Track priority configuration	1677
Backup preempt configuration	1678
Changing the timer scale	1678
VRRP-E slow start timer	1679
VRRP-E Extension for Server Virtualization	1680
Forcing a Master router to abdicate to a Backup router	1683
Displaying VRRP and VRRP-E information	1684
Displaying summary information	1684
Displaying detailed information	1686
Displaying statistics	1692
Clearing VRRP or VRRP-E statistics	1694
Displaying CPU utilization statistics	1694
Displaying VRRP and VRRP-E information for IPv6	1696
Displaying detailed information for IPv6 VRRP v3 and IPv6 VRRP-E v3	1696
Configuration examples	1698
VRRP example	1698
VRRP-E example	1700

Chapter 40

Rule-Based IP ACLs

ACL overview	1705
Types of IP ACLs	1705
ACL IDs and entries	1706
Numbered and named ACLs	1706
Default ACL action	1707
How hardware-based ACLs work	1707
How fragmented packets are processed	1707
Hardware aging of Layer 4 CAM entries	1708
ACL configuration considerations	1708
Configuring standard numbered ACLs	1709
Standard numbered ACL syntax	1709
Configuration example for standard numbered ACLs	1711
Standard named ACL configuration	1711
Standard named ACL syntax	1711
Configuration example for standard named ACLs	1713

Extended numbered ACL configuration	1714
Extended numbered ACL syntax	1714
Configuration examples for extended numbered ACLs	1718
Extended named ACL configuration	1720
Extended named ACL syntax	1721
Configuration example for extended named ACLs	1725
Applying egress ACLs to Control (CPU) traffic	1725
Preserving user input for ACL TCP/UDP port numbers	1725
ACL comment text management	1726
Adding a comment to an entry in a numbered ACL	1726
Adding a comment to an entry in a named ACL	1727
Deleting a comment from an ACL entry	1727
Viewing comments in an ACL	1727
Applying an ACL to a virtual interface in a protocol- or subnet-based VLAN	1728
ACL logging	1729
Enabling strict control of ACL filtering of fragmented packets ..	1732
Enabling ACL support for switched traffic in the router image ..	1733
Enabling ACL filtering based on VLAN membership or VE port membership	1734
Configuration notes for ACL filtering	1734
Applying an IPv4 ACL to specific VLAN members on a port (Layer 2 devices only)	1734
Applying an IPv4 ACL to a subset of ports on a virtual interface (Layer 3 devices only)	1735
ACLs to filter ARP packets	1736
Configuration considerations for filtering ARP packets	1737
Configuring ACLs for ARP filtering	1737
Displaying ACL filters for ARP	1738
Clearing the filter count	1738
Filtering on IP precedence and ToS values	1739
TCP flags - edge port security	1739
QoS options for IP ACLs	1740
Configuration notes for QoS options on FCX and ICX devices	1740
Using an ACL to map the DSCP value (DSCP CoS mapping)	1741
Using an IP ACL to mark DSCP values (DSCP marking)	1742
DSCP matching	1744
ACL-based rate limiting	1744
ACL statistics	1745
ACLs to control multicast features	1745
Enabling and viewing hardware usage statistics for an ACL	1745
Displaying ACL information	1746
Troubleshooting ACLs	1747

	Policy-based routing (PBR)	1747
Chapter 41	IPv6 ACLs	
	IPv6 ACL overview	1755
	IPv6 ACL traffic filtering criteria	1756
	IPv6 protocol names and numbers.	1756
	IPv6 ACL configuration notes	1757
	Configuring an IPv6 ACL	1758
	Example IPv6 configurations.	1758
	Default and implicit IPv6 ACL action.	1759
	Creating an IPv6 ACL.	1761
	Syntax for creating an IPv6 ACL	1761
	Enabling IPv6 on an interface to which an ACL will be applied . .	1768
	Syntax for enabling IPv6 on an interface	1768
	Applying an IPv6 ACL to an interface.	1768
	Syntax for applying an IPv6 ACL	1769
	Applying an IPv6 ACL to a trunk group	1769
	Applying an IPv6 ACL to a virtual interface in a protocol-based or subnet-based VLAN	1769
	Adding a comment to an IPv6 ACL entry	1769
	Deleting a comment from an IPv6 ACL entry	1770
	Support for ACL logging	1770
	Displaying IPv6 ACLs.	1770
Chapter 42	Traffic Policies	
	Traffic policies overview	1773
	Configuration notes and feature limitations for traffic policies.	1774
	Maximum number of traffic policies supported on a device	1775
	Setting the maximum number of traffic policies supported on a Layer 3 device	1776
	ACL-based rate limiting using traffic policies.	1776
	Support for fixed rate limiting and adaptive rate limiting . .	1777
	Configuring ACL-based fixed rate limiting.	1777
	ACL-based adaptive rate limiting configuration	1778
	Specifying the action to be taken for packets that are over the limit.	1781
	ACL statistics and rate limit counting	1782
	Enabling ACL statistics	1783
	Enabling ACL statistics with rate limiting traffic policies.	1784
	Viewing ACL and rate limit counters	1784
	Clearing ACL and rate limit counters	1785
	Viewing traffic policies	1786

	CPU rate-limiting	1787
Chapter 43	802.1X Port Security	
	IETF RFC support.	1789
	How 802.1X port security works	1790
	Device roles in an 802.1X configuration	1790
	Communication between the devices	1791
	Controlled and uncontrolled ports	1791
	Message exchange during authentication.....	1793
	Authenticating multiple hosts connected to the same port .	1795
	802.1X port security and sFlow	1798
	802.1X accounting.....	1799
	802.1X port security configuration	1799
	Configuring an authentication method list for 802.1X	1800
	Setting RADIUS parameters	1800
	Dynamic VLAN assignment for 802.1X port configuration .	1802
	Dynamically applying IP ACLs and MAC address filters	
	to 802.1X ports	1806
	Enabling 802.1X port security.....	1810
	Setting the port control	1810
	Configuring periodic re-authentication.....	1811
	Re-authenticating a port manually.....	1812
	Setting the quiet period.....	1812
	Specifying the wait interval and number of EAP-request/	
	identity frame retransmissions from the Brocade device..	1812
	Wait interval and number of EAP-request/	
	identity frame retransmissions from the RADIUS server ..	1813
	Specifying a timeout for retransmission of messages	
	to the authentication server	1814
	Initializing 802.1X on a port	1814
	Allowing access to multiple hosts.....	1815
	MAC address filters for EAP frames	1817
	Configuring VLAN access for non-EAP-capable clients	1818
	802.1X accounting configuration.....	1818
	802.1X Accounting attributes for RADIUS	1819
	Enabling 802.1X accounting.....	1819
	Displaying 802.1X information.....	1821
	Displaying 802.1X configuration information	1821
	Displaying 802.1X statistics	1824
	Clearing 802.1X statistics.....	1825
	Displaying dynamically-assigned VLAN information	1825
	Displaying information about dynamically applied	
	MAC address filters and IP ACLs.....	1826
	Displaying 802.1X multiple-host	
	authentication information	1829
	Sample 802.1X configurations.....	1833
	Point-to-point configuration.....	1833
	Hub configuration	1834
	802.1X Authentication with dynamic VLAN assignment...	1835

	Multi-device port authentication and 802.1X security on the same port.	1836
Chapter 44	MAC Port Security	
	MAC port security overview	1838
	Local and global resources used for MAC port security ...	1838
	Configuration notes and feature limitations for MAC port security	1838
	MAC port security configuration.	1839
	Enabling the MAC port security feature	1839
	Setting the maximum number of secure MAC addresses for an interface	1840
	Setting the port security age timer	1840
	Specifying secure MAC addresses	1841
	Autosaving secure MAC addresses to the startup configuration	1841
	Specifying the action taken when a security violation occurs	1842
	Clearing port security statistics	1843
	Clearing restricted MAC addresses	1843
	Clearing violation statistics	1843
	Displaying port security information	1844
	Displaying port security settings	1844
	Displaying the secure MAC addresses	1844
	Displaying port security statistics	1845
	Displaying restricted MAC addresses on a port	1847
Chapter 45	Multi-Device Port Authentication	
	How multi-device port authentication works	1850
	RADIUS authentication	1850
	Authentication-failure actions	1851
	Supported RADIUS attributes	1851
	Support for dynamic VLAN assignment	1852
	Support for dynamic ACLs	1852
	Support for authenticating multiple MAC addresses on an interface	1852
	Support for dynamic ARP inspection with dynamic ACLs ..	1852
	Support for DHCP snooping with dynamic ACLs	1852
	Support for source guard protection	1853
	Multi-device port authentication and 802.1X security on the same port.	1853
	Configuring Brocade-specific attributes on the RADIUS server	1854

Multi-device port authentication configuration.....	1855
Enabling multi-device port authentication.....	1855
Specifying the format of the MAC addresses sent to the RADIUS server	1856
Specifying the authentication-failure action	1856
Generating traps for multi-device port authentication	1857
Defining MAC address filters.....	1857
Configuring dynamic VLAN assignment	1858
Dynamically applying IP ACLs to authenticated MAC addresses	1861
Enabling denial of service attack protection	1864
Enabling source guard protection.....	1864
Clearing authenticated MAC addresses.....	1866
Disabling aging for authenticated MAC addresses	1866
Changing the hardware aging period for blocked MAC addresses	1867
Specifying the aging time for blocked MAC addresses	1868
Specifying the RADIUS timeout action	1868
Multi-device port authentication password override	1870
Limiting the number of authenticated MAC addresses....	1870
Displaying multi-device port authentication information	1870
Displaying authenticated MAC address information	1871
Displaying multi-device port authentication configuration information	1871
Displaying multi-device port authentication information for a specific MAC address or port	1872
Displaying the authenticated MAC addresses.....	1873
Displaying the non-authenticated MAC addresses	1874
Displaying multi-device port authentication information for a port.....	1874
Displaying multi-device port authentication settings and authenticated MAC addresses	1875
Displaying the MAC authentication table for FCX devices..	1878
Example port authentication configurations.....	1879
Multi-device port authentication with dynamic VLAN assignment.....	1879
Examples of multi-device port authentication and 802.1X authentication configuration on the same port.....	1882

Chapter 46

Web Authentication

Web authentication overview.....	1887
Web authentication configuration considerations	1888
Web authentication configuration tasks	1890
Enabling and disabling web authentication	1891
Web authentication mode configuration.....	1892
Using local user databases.....	1892
Passcodes for user authentication.....	1896
Automatic authentication	1900

Web authentication options configuration	1901
Enabling RADIUS accounting for web authentication	1901
Changing the login mode (HTTPS or HTTP)	1901
Specifying trusted ports.	1901
Specifying hosts that are permanently authenticated	1902
Configuring the re-authentication period	1903
Defining the web authentication cycle	1903
Limiting the number of web authentication attempts.	1903
Clearing authenticated hosts from the web authentication table	1903
Setting and clearing the block duration for web authentication attempts	1904
Manually blocking and unblocking a specific host	1904
Limiting the number of authenticated hosts	1904
Filtering DNS queries.	1905
Forcing re-authentication when ports are down	1905
Forcing re-authentication after an inactive period	1906
Defining the web authorization redirect address	1906
Deleting a web authentication VLAN	1906
Web authentication pages	1908
Displaying web authentication information.	1915
Displaying the web authentication configuration	1915
Displaying a list of authenticated hosts	1917
Displaying a list of hosts attempting to authenticate	1918
Displaying a list of blocked hosts	1919
Displaying a list of local user databases	1919
Displaying a list of users in a local user database	1920
Displaying passcodes	1920

Chapter 47

DoS Attack Protection

Smurf attacks	1921
Avoiding being an intermediary in a Smurf attack.	1922
Avoiding being a victim in a Smurf attack	1922
TCP SYN attacks	1923
TCP security enhancement	1925
Displaying statistics about packets dropped because of DoS attacks	1926

Chapter 48

DHCP

Dynamic ARP inspection.	1927
ARP poisoning	1927
About Dynamic ARP Inspection.	1928
Configuration notes and feature limitations for DAI	1929
Dynamic ARP inspection configuration	1930
Displaying ARP inspection status and ports	1931
Displaying the ARP table	1931

DHCP snooping	1931
How DHCP snooping works	1932
System reboot and the binding database	1933
Configuration notes and feature limitations for DHCP snooping	1933
Configuring DHCP snooping	1934
Clearing the DHCP binding database	1935
Displaying DHCP snooping status and ports	1935
Displaying the DHCP snooping binding database	1935
Displaying DHCP binding entry and status	1935
DHCP snooping configuration example	1936
DHCP relay agent information	1936
Configuration notes for DHCP option 82	1937
DHCP Option 82 sub-options	1937
DHCP option 82 configuration	1939
Viewing information about DHCP option 82 processing	1941
IP source guard	1942
Configuration notes and feature limitations for IP source guard	1943
Enabling IP source guard on a port	1944
Defining static IP source bindings	1944
Enabling IP source guard per-port-per-VLAN	1945
Enabling IP source guard on a VE	1945
Displaying learned IP addresses	1945

Chapter 49

Rate Limiting and Rate Shaping on FastIron X Series and FCX and ICX Series Switches

Rate limiting overview	1947
Rate limiting in hardware	1948
How Fixed rate limiting works	1948
Configuration notes for rate limiting	1949
Configuring a port-based rate limiting policy	1949
Configuring an ACL-based rate limiting policy	1950
Displaying the fixed rate limiting configuration	1951
Rate shaping overview	1952
Configuration notes for rate shaping	1952
Configuring outbound rate shaping for a port	1953
Configuring outbound rate shaping for a specific priority ..	1953
Configuring outbound rate shaping for a trunk port	1953
Displaying rate shaping configurations	1954

Chapter 50

Rate Limiting on FastIron WS Series Switches

Rate limiting overview	1955
Rate limiting in hardware	1956
How fixed rate limiting works	1956

Fixed rate limiting on inbound port configuration	1957
Minimum and maximum inbound rate limits	1957
Configuration notes for fixed rate limiting	1957
Configuration syntax for fixed rate limiting	1957
Fixed rate limiting on outbound port configuration	1958
Minimum and maximum outbound rate limits	1958
Configuration notes for outbound rate limiting	1958
Configuring port-based fixed rate limiting	1959
Configuring port- and priority-based rate limiting	1959
ACL-based rate limiting policy configuration	1960
Displaying the fixed rate limiting configuration	1960
Inbound ports	1960
Displaying fixed rate limiting configuration on outbound ports	1961

Chapter 51

Quality of Service

QoS overview	1963
Processing of classified traffic	1964
QoS for Brocade stackable devices	1970
QoS profile restrictions in an IronStack	1970
QoS behavior for trusting Layer 2 (802.1p) in an IronStack .	1971
QoS behavior for trusting Layer 3 (DSCP) in an IronStack .	1971
QoS behavior on port priority and VLAN priority in an IronStack	1971
QoS behavior for 802.1p marking in an IronStack	1971
QoS queues	1971
Queues for the SX-FI48GPP interface module	1972
Queues for the SX-FI-8XG interface module	1973
Queues for the ICX 6430 switch	1973
User-configurable scheduler profile	1975
QoS priorities-to-traffic assignment	1978
Changing a port priority	1978
Assigning static MAC entries to priority queues	1978
Buffer allocation and threshold for QoS queues	1979
802.1p priority override	1979
Configuration notes and feature limitations	1980
Enabling 802.1p priority override	1980
Marking	1980
DSCP-based QoS configuration	1981
Application notes for DSCP-based QoS	1981
Using ACLs to honor DSCP-based QoS	1981
Trust DSCP for the SX-FI48GPP, SX-FI-24GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI8XG modules	1982

Configuring QoS mapping configuration	1982
Default DSCP to internal forwarding priority mappings. . . .	1983
Changing the DSCP to internal forwarding priority mappings	1984
Changing the VLAN priority 802.1p to hardware forwarding queue mappings	1985
Default scheduling configuration for the SX-FI48GPP module.	1985
Default scheduling configuration for the ICX 6430.	1986
Scheduling QoS information.	1987
Scheduling for the SX-FI48GPP module.	1987
QoS queuing methods	1987
Selecting the QoS queuing method	1988
Configuring the QoS queues	1989
Viewing QoS settings.	1992
Viewing DSCP-based QoS settings.	1992

Appendix A Syslog messages

Appendix B NIAP-CCEVS Certification

NIAP-CCEVS certified Brocade equipment and Ironware releases	2021
Web-Management access to NIAP-CCEVS certified Brocade equipment.	2022
Local user password changes	2022

Index

About This Guide

Introduction

This guide describes the following product families from Brocade:

- FastIron X Series devices:
 - FastIron Edge Switch X Series (FESX) Layer 2/Layer 3 switch
 - FastIron Edge Switch X Series Expanded (FESXE) Layer 2/Layer 3 switch
 - FastIron SX 800 and 1600 Layer 2/Layer 3 switches
- FastIron WS (FWS) Layer 2, base Layer 3, and EPREM devices
- Brocade FCX Series (FCX) Stackable Switch
- Brocade ICX™ 6610 (ICX 6610) Stackable Switch
- Brocade ICX 6430 Series (ICX 6430)
- Brocade ICX 6450 Series (ICX 6450)

This guide includes procedures for configuring the software. The software procedures show how to perform tasks using the CLI. This guide also describes how to monitor Brocade products using statistics and summary screens.

This guide applies to the FastIron models listed in [Table 1](#).

Device nomenclature

[Table 1](#) lists the terms (product names) contained in this guide and the specific set of devices to which each term refers.

TABLE 1 FastIron family of switches

This name	Refers to these devices
FastIron X Series Devices	
NOTE: The FastIron X Series product family includes compact switch models and chassis models. The compact models are referred to as FESX switches. The chassis models are referred to as the FastIron SX switches. Chassis systems have these models: FastIron SX 800 and FastIron SX 1600.	
FastIron Edge Switch X Series (FESX)	FESX624, FESX624HF, FESX624-PREM, FESX624-PREM6, FESX624HF-PREM, FESX624HF-PREM6, FESX648, FESX648-PREM, FESX648-PREM6
FastIron SX Management Modules	FastIron SX 800/1600 Management modules with: <ul style="list-style-type: none">• 667MHz / 512MB NOTE: For a complete list of the SX 800/1600 Management modules and their part numbers, see the <i>Brocade FastIron X Series Chassis Hardware Installation Guide</i> .

Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Brocade Layer 3 Switch, you should be familiar with the following protocols if applicable to your network – IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, DVMRP, and VRRP.

What's new in this document

This document includes the information from IronWare software release 07.4.00.

Summary of enhancements in FastIron release 07.4.00a

[Table 2](#) lists the enhancements for FastIron release 07.4.00a.

TABLE 2 Summary of Enhancements in FastIron release 07.4.00a

Feature	Description	Described in
Stacking configuration for ICX 6430	This release supports including ICX 6430 48-port units and ICX 6430 24-port stacking units in an IronStack.	“Connecting ICX 6450 and ICX 6430 devices in a stack” on page 244 “Configuring an ICX 6430 and ICX 6450 IronStack” on page 266
IPv6 routing features	This release adds the following IPv6 routing features: <ul style="list-style-type: none">• IPv6 static routes• ECMP load sharing	“Static IPv6 route configuration” on page 373 “ECMP load sharing for IPv6” on page 381

Summary of enhancements in FastIron release 07.4.00

Table 3 lists the enhancements for FastIron release 07.4.00.

TABLE 3 Summary of Enhancements in FastIron release 07.4.00

Feature	Description	Described in
New hardware	This release introduces the new Brocade ICX 6430 and ICX 6450 Series stackable switches.	<i>Brocade ICX 6430 and ICX 6450 Stackable Switches Hardware Installation Guide</i>
Optics for ICX 6430 and ICX 6450	<p>The following optics are supported on the Brocade ICX 6430 and ICX 6450 Series stackable switches:</p> <ul style="list-style-type: none"> • E1MG-TX • E1MG-SX-OM 1000Base-SX SFP optic • E1MG-LX-OM 1000Base-LX SFP optic • E1MG-LHA-OM 1000Base-LHA SFP optic • E1MG-LHB 1000Base-LHB SFP optic • 10G-SFPP-SR 10GBASE-SR, SFP+ optic (LC) • 10G-SFPP-LR 10GBASE-LR, SFP+ optic (LC) • 10G-SFPP-LRM 10GBASE-LRM, 1310NM SFP+ OPTIC (LC), • 10G-SFPP-ER 10GBASE-ER SFP+ optic (LC) • 10G-SFPP-USR • 1G-SFP-TWX-0101, 1G-SFP-TWX-0501 • 10G-SFPP-TWX-0101, 10G-SFPP-TWX-0301, 10G-SFPP-TWX-0501 	http://www.brocade.com/downloads/documents/data_sheets/product_data_sheets/Optics_DS.pdf
New optic	<p>The 10G-SFPP-USR is also supported on the following devices:</p> <ul style="list-style-type: none"> • FCX • FSX devices that have SX-FI-8XG or SX-FI-2XG modules installed • ICX 6610 • ICX 6450 • Turbolron 24X 	http://www.brocade.com/downloads/documents/data_sheets/product_data_sheets/Optics_DS.pdf
Multi-Chassis Trunking	<p>Multi-Chassis Trunking (MCT) is technology that allows multiple switches to appear as single logical switch connecting to another switch using a standard LAG. MCT is supported on FSX devices that have any of the following modules installed.</p> <ul style="list-style-type: none"> • SX-FI-24GPP • SX-FI-24HF • SX-FI-2XG • SX-FI-8XG • SX-FI48GPP 	“Multi-Chassis Trunking” on page 827
MAC address move notification	<p>MAC address movement notification allows you to monitor the movement of MAC addresses that migrate from port to port. It enables you to distinguish between legitimate movement and malicious movement by allowing you to define malicious use as a threshold number of times a MAC address moves within a specific interval.</p>	“Monitoring MAC address movement” on page 574

TABLE 3 Summary of Enhancements in FastIron release 07.4.00 (Continued)

Feature	Description	Described in
Maximum Trunk support and Maximum trunk port member support on FastIron SX interface modules	<p>The following Interface modules support up to 12 ports per trunk group on the FastIron SX chassis for both static and LACP trunks. A maximum of 256 trunks are supported when the following modules are installed on the chassis:</p> <ul style="list-style-type: none"> • SX-FI48GPP—48-port 10/100/1000 Mbps Ethernet PoE interface module • SX-FI24GPP—24-port Gigabit Ethernet copper interface module • SX-FI24HF—24-port Gigabit Ethernet fiber interface module • SX-FI2XG—2-port 10 Gigabit Ethernet interface module • SX-FI8XG—8-port 10 Gigabit Ethernet interface module 	“Trunk Groups and Dynamic Link Aggregation” on page 697
VRRP-E short-path forwarding and revertible option	Also known as ‘VRRP-E Extension for Server Virtualization,’ enables the Brocade device to bypass the master router and directly forward packets to their destination through interfaces on the backup router. If enabled, the traffic travels through the short-path forwarding (SPF) path to reach the client.	“VRRP and VRRP-E” on page 1649
CPU rate-limiting	Increases CPU efficiency by preventing unnecessary traffic from being sent to the CPU and by limiting the rate at which some packet types are delivered to the CPU.	“CPU rate-limiting” on page 1787.
Support for 16K Nexthops	<p>16,000 nexthops are supported on the following modules:</p> <ul style="list-style-type: none"> • SX-FI-24GPP • SX-FI-24HF • SX-FI-2XG • SX-FI-8XG • SX-FI48GPP 	Chapter 28, “Base Layer 3 and Routing Protocols”
Quality of Service (QoS) support for ICX 6430 and 6450 devices	<p>Differences are detailed for ICX 6430 devices in the following areas:</p> <ul style="list-style-type: none"> • Priority to hardware forwarding queue mapping • Default QoS mappings • Default values for scheduling type • Default scheduling configuration 	Chapter 51, “Quality of Service”
Dynamic buffer support for 6430 and 6450 devices	<p>Specific information is provided for the ICX 6430 and 6450 devices in the following areas:</p> <ul style="list-style-type: none"> • Port and buffer descriptor values • Buffer sharing levels • Configuring values for ICX devices 	Chapter 16, “Basic Layer 2 Features”

TABLE 3 Summary of Enhancements in FastIron release 07.4.00 (Continued)

Feature	Description	Described in
Stacking configuration for ICX 6430 and ICX 6450 devices	ICX 6430 and ICX 6450 devices have four ports on the front panel for stacking configuration. ICX 6430 and ICX 6450 devices ship with two default stacking ports configured. ICX 6430 and ICX 6450 devices support linear and ring stack topologies, and can also operate as standalone devices. When stacking is enabled, ports 1 and 3 are dedicated to stacking and cannot be used for data ports. If stacking is not enabled on the ports, then all four stacking ports can be used for data or uplink ports.	“Connecting ICX 6450 and ICX 6430 devices in a stack” on page 244 “Configuring an ICX 6430 and ICX 6450 IronStack” on page 266
Auto Image Copy for stack units	The Auto Image Copy feature ensures that all units in a stack are running the same flash image after a stack merge. This feature is introduced on the ICX and FCX devices.	“Auto Image Copy for stack units” on page 321
Software licensing enhancements	Software-based licensing is introduced on the ICX 6450 devices. The premium license is available on the ICX 6450 devices.	“Software-based Licensing” on page 199
Licensing for Ports on Demand	Licensing for Ports on Demand (POD) is introduced on the ICX 6450 devices. The ICX 6450 device has four active uplink and stacking ports on slot 2. By default, regardless of what SFP+ media optic is used, ports 1 and 3 are 10 Gbps ports. By default, without a license at bootup, ports 2 and 4 come up in 10 Gbps port speed in an error disabled state. To enable ports 2 and 4 to 10 Gbps port speed, purchase the ICX6450-2X10G-LIC-POD license. The PoD feature is not applicable to ICX 6430 devices because there are no 10 Gbps ports on the device.	“Licensed features and part numbers” on page 202 “Licensing for Ports on Demand” on page 208
31-bit subnet mask	31-bit subnet masks can be configured on point-to-point interfaces on FSX, FCX, and ICX 6610 devices running full layer 3 image	“Configuring 31-bit subnet masks on point-to-point networks” on page 968
Web Management Interface	Support for the Web Management Interface on the ICX 6430 and ICX 6450 devices.	<i>Brocade FCX, Brocade FastIron SX, Brocade ICX Web Management Interface User Guide</i>
MIB enhancement	Support for the registration MIBs on the ICX 6430 and ICX 6450 devices.	<i>Unified IP MIB Reference</i>
Diagnostic information	Support for “supportsave” CLI commands and RAS requirement debug commands.	<i>Brocade FastIron, FCX, ICX, Turbolron Diagnostic and Troubleshooting Reference</i>
VLAN-based mirroring	Support for VLAN-based mirroring for FastIron X Series modules.	“VLAN-based mirroring on FastIron X Series devices” on page 939
802.1x user name support	802.1x user name support for RADIUS accounting messages.	“Support for RADIUS user-name attribute in access-accept messages” on page 1795

Unsupported features

Features that are not documented in this guide are not supported. [Table 4](#) lists the features that are not supported on Brocade FastIron devices. If required, these features are available on other Brocade devices.

TABLE 4 Unsupported Features

Unsupported features
System-level features not supported:
<ul style="list-style-type: none">• ACL logging of permitted packets• Broadcast and multicast MAC address filters• Outbound ACLs on FWS devices
Layer 2 features not supported:
<ul style="list-style-type: none">• SuperSpan• VLAN-based priority
Layer 3 features not supported:
<ul style="list-style-type: none">• AppleTalk Routing• Foundry Standby Router Protocol (FSRP)• IPv6 Multicast Routing• IPX Routing• IS-IS• Multiprotocol Border Gateway Protocol (MBGP)• Multiprotocol Label Switching (MPLS)• Network Address Translation (NAT)

Document conventions

This section describes text formatting conventions and important notice formats used in this document.

Text formatting

The narrative-text formatting conventions that are used are as follows:

bold text	Identifies command names
	Identifies the names of user-manipulated GUI elements
	Identifies keywords
	Identifies text to enter at the GUI or CLI
<i>italic text</i>	Provides emphasis
	Identifies variables
	Identifies document titles
<code>code text</code>	Identifies CLI output

For readability, command names in the narrative portions of this guide are presented in bold: for example, **show version**.

Command syntax conventions

Command syntax in this manual follows these conventions:

command and parameters	Commands and parameters are printed in bold.
[]	Optional parameter.
<i>variable</i>	Variables are printed in italics enclosed in angled brackets < >.
...	Repeat the previous element, for example “member[;member...]”
	Choose from one of the parameters.

Notes, cautions, and danger notices

The following notices and statements are used in this manual. They are listed below in order of increasing severity of potential hazards.

NOTE

A note provides a tip, guidance or advice, emphasizes important information, or provides a reference to related information.



CAUTION

A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.



DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Related publications

The following Brocade documents supplement the information in this guide:

- *Brocade FCX, Brocade FastIron SX, Brocade ICX Web Management Interface User Guide*
- *Brocade ICX 6610 Series Hardware Installation Guide*
- *Brocade FastIron CX Hardware Installation Guide*
- *Brocade FastIron X Series Chassis Hardware Installation Guide*
- *Brocade FastIron Compact Switch Hardware Installation Guide*
- *Brocade FastIron Edge X-Series Hardware Installation Guide*
- *Brocade FastIron WS Switch Hardware Installation Guide*
- *Brocade ICX 6450, ICX 6430 Switch Hardware Installation Guide*

- *Brocade FCX and Brocade ICX 6610 Debug Guide*
- *Brocade FCX Series Hardware Installation Guide*
- *Unified IP MIB Reference*

The latest version of these guides are posted at <http://www.brocade.com/ethernetproducts>.

If you find errors in the guides, send an email to documentation@brocade.com

Getting technical help

To contact Technical Support, go to <http://www.brocade.com/services-support/index.page> for the latest e-mail and telephone contact information.

Table 5 lists the individual Brocade FastIron switches and the management application features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images.

TABLE 5 Supported management application features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Management port	Yes (FSX 800 and FSX 1600 only)	No	Yes	Yes	Yes
industry-standard Command Line Interface (CLI), including support for: <ul style="list-style-type: none">• Serial and Telnet access• Alias command• On-line help• Command completion• Scroll control• Line editing• Searching and filtering output• Special characters	Yes	Yes	Yes	Yes	Yes
Web-based GUI <ul style="list-style-type: none">• Web Management Interface	Yes	Yes	Yes	Yes	Yes

Management port overview

NOTE

The management port applies to FCX, SX 800, SX 1600, ICX 6430, and ICX 6450 devices.

The management port is an out-of-band port that customers can use to manage their devices without interfering with the in-band ports. The management port is widely used to download images and configurations, for Telnet sessions, and for Web management.

For FCX devices, the MAC address for the management port is derived from the base MAC address of the unit, plus the number of ports in the base module. For example, on a 48-port FCX standalone device, the base MAC address is 0000.1234.2200. The management port MAC address for this device would be 0000.1234.2200 plus 0x30, or 0000.1234.2230. The 0x30 in this case equals the 48 ports on the base module.

For SX 800 and SX 1600 devices, the MAC address for the management port is derived as if the management port is the last port on the management module where it is located. For example, on a 2 X 10G management module, the MAC address of the management port is that of the third port on that module.

How the management port works

The following rules apply to management ports:

- Only packets that are specifically addressed to the management port MAC address or the broadcast MAC address are processed by the Layer 2 switch or Layer 3 switch. All other packets are filtered out.
- No packet received on a management port is sent to any in-band ports, and no packets received on in-band ports are sent to a management port.
- A management port is not part of any VLAN
- Protocols are not supported on the management port.
- Creating a management VLAN disables the management port on the device.
- For FCX devices, all features that can be configured from the global configuration mode can also be configured from the interface level of the management port. Features that are configured through the management port take effect globally, not on the management port itself.

For switches, any in-band port may be used for management purposes. A router sends Layer 3 packets using the MAC address of the port as the source MAC address.

For stacking devices, (for example, an FCX stack) each stack unit has one out-of band management port. Only the management port on the Active Controller will actively send and receive packets. If a new Active Controller is elected, the new Active Controller management port will become the active management port. In this situation, the MAC address of the old Active Controller and the MAC address of the new controller will be different.

CLI Commands for use with the management port

The following CLI commands can be used with a management port.

To display the current configuration, use the **show running-config interface management** command.

Syntax: **show running-config interface management** <num>

```
Brocade(config-if-mgmt)#ip addr 10.44.9.64/24
Brocade(config)#show running-config interface management 1
interface management 1
ip address 10.44.9.64 255.255.255.0
```

To display the current configuration, use the **show interfaces management** command.

Syntax: **show interfaces management** <num>

```
Brocade(config)#show interfaces management 1
GigEthernetmgmt1 is up, line protocol is up
Hardware is GigEthernet, address is 0000.9876.544a (bia 0000.9876.544a)
Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
Configured mdi mode AUTO, actual none
BPRU guard is disabled, ROOT protect is disabled
```



```

Link Error Dampening is Disabled
STP configured to OFF, priority is level0, mac-learning is enabled
Flow Control is config disabled, oper enabled
Mirror disabled, Monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 0 bits-time, IPG GMII 0 bits-time
IP MTU 1500 bytes
300 second input rate: 83728 bits/sec, 130 packets/sec, 0.01% utilization
300 second output rate: 24 bits/sec, 0 packets/sec, 0.00% utilization
39926 packets input, 3210077 bytes, 0 no buffer
Received 4353 broadcasts, 32503 multicasts, 370 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
22 packets output, 1540 bytes, 0 underruns
Transmitted 0 broadcasts, 6 multicasts, 16 unicasts
0 output errors, 0 collisions

```

To display the management interface information in brief form, enter the **show interfaces brief management** command.

Syntax: **show interfaces brief management** <num>

```

Brocade(config)#show interfaces brief management 1
Port  Link  State  Dupl  Speed  Trunk  Tag  Pri  MAC  Name
mgmt1 Up    None   Full  1G     None   No   0    0000.9876.544a

```

To display management port statistics, enter the **show statistics management** command.

Syntax: **show statistics management** <num>

```

Brocade(config)#show statistics management 1
Port  Link  State  Dupl  Speed  Trunk  Tag  Pri  MAC  Name
mgmt1 Up    None   Full  1G     None   No   0    0000.9876.544a

```

```

Port mgmt1 Counters:
  InOctets3210941OutOctets1540
  InPkts39939OutPackets22
  InBroadcastPkts4355OutBroadcastPkts0
  InMulticastPkts35214OutMulticastPkts6
  InUnicastPkts370OutUnicastPkts16
  InBadPkts0
  InFragments0
  InDiscards0OutErrors0
  CRC 0 Collisions0
  InErrors0 LateCollisions0
  InGiantPkts0
  InShortPkts0
  InJabber0
  InFlowCtrlPkts0OutFlowCtrlPkts0
  InBitsPerSec83728OutBitsPerSec24
  InPktsPerSec130OutPktsPerSec0
  InUtilization0.01%OutUtilization0.00%

```

To display the management interface statistics in brief form, enter the **show statistics brief management** command.

Syntax: **show statistics brief management** <num>

```
Brocade(config)#show statistics brief management 1
PortIn PacketsOut PacketsTrunkIn ErrorsOut Errors
mgmt1399462200

Total1399452200
```

Logging on through the CLI

Once an IP address is assigned to a Brocade device running Layer 2 software or to an interface on the Brocade device running Layer 3 software, you can access the CLI either through the direct serial connection to the device or through a local or remote Telnet session.

You can initiate a local Telnet or SNMP connection by attaching a cable to a port and specifying the assigned management station IP address.

The commands in the CLI are organized into the following levels:

- **User EXEC** – Lets you display information and perform basic tasks such as pings and traceroutes.
- **Privileged EXEC** – Lets you use the same commands as those at the User EXEC level plus configuration commands that do not require saving the changes to the system-config file.
- **CONFIG** – Lets you make configuration changes to the device. To save the changes across reboots, you need to save them to the system-config file. The CONFIG level contains sub-levels for individual ports, for VLANs, for routing protocols, and other configuration areas.

NOTE

By default, any user who can open a serial or Telnet connection to the Brocade device can access all these CLI levels. To secure access, you can configure Enable passwords or local user accounts, or you can configure the device to use a RADIUS or TACACS/TACACS+ server for authentication. Refer to [Chapter 4, “Security Access”](#).

Online help

To display a list of available commands or command options, enter “?” or press Tab. If you have not entered part of a command at the command prompt, all the commands supported at the current CLI level are listed. If you enter part of a command, then enter “?” or press Tab, the CLI lists the options you can enter at this point in the command string.

If you enter an invalid command followed by ?, a message appears indicating the command was unrecognized. An example is given below.

```
Brocade(config)#rooter ip
Unrecognized command
```

Command completion

The CLI supports command completion, so you do not need to enter the entire name of a command or option. As long as you enter enough characters of the command or option name to avoid ambiguity with other commands or options, the CLI understands what you are typing. This feature is not available in the boot loader prompt of ICX 6430 and ICX 6450 devices.

Scroll control

By default, the CLI uses a page mode to paginate displays that are longer than the number of rows in your terminal emulation window. For example, if you display a list of all the commands at the global CONFIG level but your terminal emulation window does not have enough rows to display them all at once, the page mode stops the display and lists your choices for continuing the display. An example is given below.

```
aaa
all-client
appletalk
arp
boot
some lines omitted for brevity...

ipx
lock-address
logging
mac
--More--, next page: Space, next line:
Return key, quit: Control-c
```

The software provides the following scrolling options:

- Press the **Space bar** to display the next page (one screen at a time).
- Press the **Return** or **Enter** key to display the next line (one line at a time).
- Press **Ctrl+C** or **Ctrl+Q** to cancel the display.

Line editing commands

The CLI supports the following line editing commands. To enter a line-editing command, use the CTRL+key combination for the command by pressing and holding the CTRL key, then pressing the letter associated with the command.

TABLE 6 CLI line editing commands

Ctrl+Key combination	Description
Ctrl+A	Moves to the first character on the command line.
Ctrl+B	Moves the cursor back one character.
Ctrl+C	Escapes and terminates command prompts and ongoing tasks (such as lengthy displays), and displays a fresh command prompt.
Ctrl+D	Deletes the character at the cursor.
Ctrl+E	Moves to the end of the current command line.
Ctrl+F	Moves the cursor forward one character.
Ctrl+K	Deletes all characters from the cursor to the end of the command line.
Ctrl+L; Ctrl+R	Repeats the current command line on a new line.
Ctrl+N	Enters the next command line in the history buffer.
Ctrl+P	Enters the previous command line in the history buffer.
Ctrl+U; Ctrl+X	Deletes all characters from the cursor to the beginning of the command line.

TABLE 6 CLI line editing commands (Continued)

Ctrl+Key combination	Description
Ctrl+W	Deletes the last word you typed.
Ctrl+Z	Moves from any CONFIG level of the CLI to the Privileged EXEC level; at the Privileged EXEC level, moves to the User EXEC level.

Using stack-unit, slot number, and port number with CLI commands

Many CLI commands require users to enter port numbers as part of the command syntax, and many **show** command outputs display port numbers. The port numbers are entered and displayed in one of the following formats:

- port number only
- slot number and port number
- stack-unit, slot number, and port number

The following sections show which format is supported on which devices. The ports are labelled on the front panels of the devices.

CLI nomenclature on Chassis-based models

Chassis-based models (FSX 800 and FSX 1600) use port numbering that consists of a slot number and a port number. When you enter CLI commands on these devices, you must specify both the slot number and the port number. The slot numbers used in the FSX CLI examples apply only to Chassis devices.

Here is an example. The following commands change the CLI from the global CONFIG level to the configuration level for the first port on the device:

- FSX commands

```
Brocade(config)#interface e 1/1
Brocade(config-if-1/1)#
```

Syntax: **ethernet** <slotnum>/<portnum>

CLI nomenclature on FESX Compact devices

The FESX compact devices use port numbers only. When you enter CLI commands that require port numbers as part of the syntax, just specify the port number.

Here are some examples. The following commands change the CLI from the global CONFIG level to the configuration level for the first port on the device:

```
Brocade(config)#interface e 1
Brocade(config-if-e1000-1)#
```

Syntax: **ethernet** <portnum>

CLI nomenclature on Stackable devices

Stackable devices (FCX and ICX) use the **stack-unit/slot/port** nomenclature. When you enter CLI commands that include the port number as part of the syntax, you must use the **stack-unit/slot/port** number format. For example, the following commands change the CLI from the global CONFIG level to the configuration level for the first port on the device:

```
Brocade(config)#interface e 1/1/1
Brocade(config-if-e1000-1/1/1)#
```

Syntax: **ethernet** <stack-unit>/<slotnum>/<portnum>

Refer to [Chapter 7, “Brocade Stackable Devices”](#) for more information about these devices.

Searching and filtering output from CLI commands

You can filter CLI output from **show** commands and at the **–More–** prompt. You can search for individual characters, strings, or construct complex regular expressions to filter the output.

Searching and filtering output from Show commands

You can filter output from **show** commands to display lines containing a specified string, lines that do not contain a specified string, or output starting with a line containing a specified string. The search string is a regular expression consisting of a single character or string of characters. You can use special characters to construct complex regular expressions. Refer to [“Using special characters in regular expressions”](#) on page 9 for information on special characters used with regular expressions.

Displaying lines containing a specified string

The following command filters the output of the **show interface** command for port 3/11 so it displays only lines containing the word “Internet”. This command can be used to display the IP address of the interface.

```
Brocade#show interface e 3/11 | include Internet
Internet address is 192.168.1.11/24, MTU 1518 bytes, encapsulation ethernet
```

Syntax: <show-command> | **include** <regular-expression>

NOTE

The vertical bar (|) is part of the command.

Note that the regular expression specified as the search string is case sensitive. In the example above, a search string of “Internet” would match the line containing the IP address, but a search string of “internet” would not.

Displaying lines that do not contain a specified string

The following command filters the output of the **show who** command so it displays only lines that do not contain the word “closed”. This command can be used to display open connections to the Brocade device.

```
Brocade#show who | exclude closed
Console connections:
    established
    you are connecting to this session
    2 seconds in idle
Telnet connections (inbound):
    1      established, client ip address 192.168.9.37
    27 seconds in idle
Telnet connection (outbound):
SSH connections:
```

Syntax: <show-command> | **exclude** <regular-expression>

Displaying lines starting with a specified string

The following command filters the output of the **show who** command so it displays output starting with the first line that contains the word “SSH”. This command can be used to display information about SSH connections to the Brocade device.

```
Brocade#show who | begin SSH
SSH connections:
    1      established, client ip address 192.168.9.210
    7 seconds in idle
    2      closed
    3      closed
    4      closed
    5      closed
```

Syntax: <show-command> | **begin** <regular-expression>

Searching and filtering output at the --More-- prompt

The --More-- prompt displays when output extends beyond a single page. From this prompt, you can press the Space bar to display the next page, the Return or Enter key to display the next line, or Ctrl+C or Q to cancel the display. In addition, you can search and filter output from this prompt.

At the --More-- prompt, you can press the forward slash key (/) and then enter a search string. The Brocade device displays output starting from the first line that contains the search string, similar to the **begin** option for **show** commands. An example is given below.

```
--More--, next page: Space, next line: Return key, quit: Control-c
/telnet
```

The results of the search are displayed.

```
searching...
telnet          Telnet by name or IP address
temperature     temperature sensor commands
terminal        display syslog
traceroute      TraceRoute to IP node
undebg          Disable debugging functions (see also 'debug')
undetele        Undetele flash card files
whois           WHOIS lookup
write           Write running configuration to flash or terminal
```

To display lines containing only a specified search string (similar to the **include** option for **show** commands) press the plus sign key (+) at the --More-- prompt and then enter the search string.

```
--More--, next page: Space, next line: Return key, quit: Control-c
+telnet
```

The filtered results are displayed.

```
filtering...
telnet                               Telnet by name or IP address
```

To display lines that do not contain a specified search string (similar to the **exclude** option for **show** commands) press the minus sign key (-) at the **--More--** prompt and then enter the search string.

```
--More--, next page: Space, next line: Return key, quit: Control-c
-telnet
```

The filtered results are displayed.

```
filtering...
temperature                         temperature sensor commands
terminal                           display syslog
traceroute                         TraceRoute to IP node
undebg                             Disable debugging functions (see also 'debug')
undete                             Undelete flash card files
whois                              WHOIS lookup
write                              Write running configuration to flash or terminal
```

As with the commands for filtering output from **show** commands, the search string is a regular expression consisting of a single character or string of characters. You can use special characters to construct complex regular expressions. See the next section for information on special characters used with regular expressions.

Using special characters in regular expressions

You use a regular expression to specify a single character or multiple characters as a search string. In addition, you can include special characters that influence the way the software matches the output against the search string. These special characters are listed in the following table.

TABLE 7 Special characters for regular expressions

Character	Operation
.	The period matches on any single character, including a blank space. For example, the following regular expression matches "aaz", "abz", "acz", and so on, but not just "az": a.z
*	The asterisk matches on zero or more sequential instances of a pattern. For example, the following regular expression matches output that contains the string "abc", followed by zero or more Xs: abcX*
+	The plus sign matches on one or more sequential instances of a pattern. For example, the following regular expression matches output that contains "de", followed by a sequence of "g"s, such as "deg", "degg", "deggg", and so on: deg+

TABLE 7 Special characters for regular expressions (Continued)

Character	Operation
?	<p>The question mark matches on zero occurrences or one occurrence of a pattern. For example, the following regular expression matches output that contains "dg" or "deg": de?g</p> <p>NOTE: Normally when you type a question mark, the CLI lists the commands or options at that CLI level that begin with the character or string you entered. However, if you enter Ctrl+V and then type a question mark, the question mark is inserted into the command line, allowing you to use it as part of a regular expression.</p>
^	<p>A caret (when not used within brackets) matches on the beginning of an input string. For example, the following regular expression matches output that begins with "deg": ^deg</p>
\$	<p>A dollar sign matches on the end of an input string. For example, the following regular expression matches output that ends with "deg": deg\$</p>
_	<p>An underscore matches on one or more of the following:</p> <ul style="list-style-type: none"> • , (comma) • { (left curly brace) • } (right curly brace) • ((left parenthesis) •) (right parenthesis) • The beginning of the input string • The end of the input string • A blank space <p>For example, the following regular expression matches on "100" but not on "1002", "2100", and so on. _100_</p>
[]	<p>Square brackets enclose a range of single-character patterns. For example, the following regular expression matches output that contains "1", "2", "3", "4", or "5": [1-5]</p> <p>You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets.</p> <ul style="list-style-type: none"> • ^ – The caret matches on any characters except the ones in the brackets. For example, the following regular expression matches output that does not contain "1", "2", "3", "4", or "5": [^1-5] • - The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. See the example above.
	<p>A vertical bar separates two alternative values or sets of values. The output can match one or the other value. For example, the following regular expression matches output that contains either "abc" or "defg": abc defg</p>
()	<p>Parentheses allow you to create complex expressions. For example, the following complex expression matches on "abc", "abcabc", or "defg", but not on "abcdefgdefg": ((abc)+) ((defg)?)</p>

If you want to filter for a special character instead of using the special character as described in the table above, enter "\ " (backslash) in front of the character. For example, to filter on output containing an asterisk, enter the asterisk portion of the regular expression as "*".


```
Brocade#show ip route bgp | include \*
```

Creating an alias for a CLI command

You can create **aliases** for CLI commands. An alias serves as a shorthand version of a longer CLI command. For example, you can create an alias called **shoro** for the CLI command **show ip route**. Then when you enter **shoro** at the command prompt, the **show ip route** command is executed.

To create an alias called **shoro** for the CLI command **show ip route**, enter the **alias shoro = show ip route** command.

```
Brocade(config)#alias shoro = show ip route
```

Syntax: [no] **alias** <alias-name> = <cli-command>

The <alias-name> must be a single word, without spaces.

After the alias is configured, entering **shoro** at either the Privileged EXEC or CONFIG levels of the CLI, executes the **show ip route** command.

To create an alias called **wrsbc** for the CLI command **copy running-config tftp 10.10.10.10 test.cfg**, enter the following command.

```
Brocade(config)#alias wrsbc = copy running-config tftp 10.10.10.10 test.cfg
```

To remove the **wrsbc** alias from the configuration, enter one of the following commands.

```
Brocade(config)#no alias wrsbc
```

or

```
Brocade(config)#unalias wrsbc
```

Syntax: **unalias** <alias-name>

The specified <alias-name> must be the name of an alias already configured on the Brocade device.

To display the aliases currently configured on the Brocade device, enter the following command at either the Privileged EXEC or CONFIG levels of the CLI.

```
Brocade#alias
      wrsbc      copy running-config tftp 10.10.10.10 test.cfg
      shoro      show ip route
```

Syntax: **alias**

Configuration notes for creating a command alias

The following configuration notes apply to this feature:

- You cannot include additional parameters with the alias at the command prompt. For example, after you create the **shoro** alias, **shoro bgp** would not be a valid command.
- If configured on the Brocade device, authentication, authorization, and accounting is performed on the actual command, not on the alias for the command.
- To save an alias definition to the startup-config file, use the **write memory** command.

Logging on through the Web Management Interface

To use the Web Management Interface, open a Web browser and enter the IP address of the management port on the Brocade device in the Location or Address field. The Web browser contacts the Brocade device and displays a Login panel, such as the one shown below.

FIGURE 1 Web Management Interface login panel

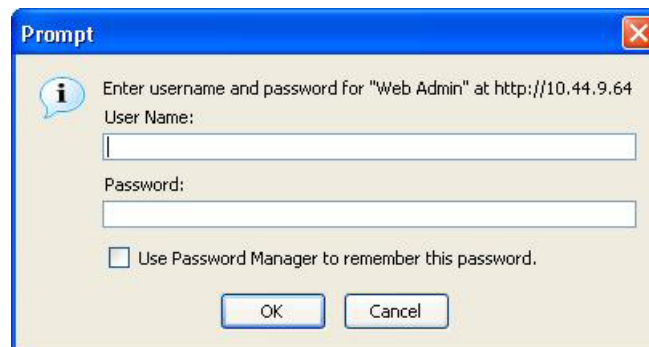


NOTE

If you are unable to connect with the device through a Web browser due to a proxy problem, it may be necessary to set your Web browser to direct Internet access instead of using a proxy. For information on how to change a proxy setting, refer to the on-line help provided with your Web browser.

To log in, click on the Login link. The following dialog box is displayed.

FIGURE 2 Web Management Interface login dialog



The login username and password you enter depends on whether your device is configured with AAA authentication for SNMP. If AAA authentication for SNMP is not configured, you can use the user name "get" and the default read-only password "public" for read-only access. However, for read-write access, you must enter "set" for the user name, and enter a read-write community string you have configured on the device for the password. There is no default read-write community string. You must add one using the CLI.

As an alternative to using the SNMP community strings to log in, you can configure the Brocade device to secure Web management access using local user accounts or Access Control Lists (ACLs).

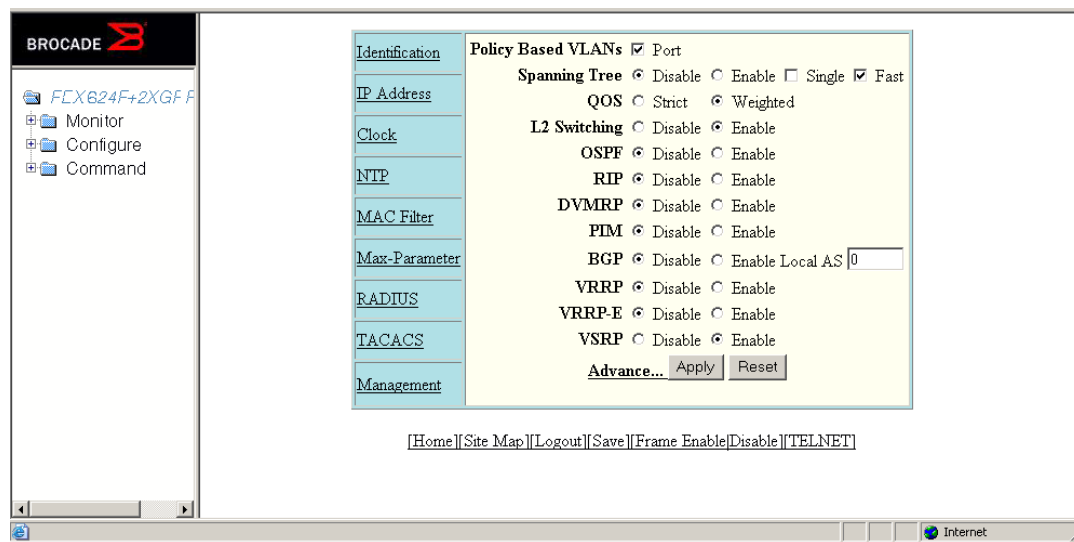
Navigating the Web Management Interface

When you log into a device, the System configuration panel is displayed. This panel allows you to enable or disable major system features. You can return to this panel from any other panel by selecting the Home link.

The Site Map link gives you a view of all available options on a single screen.

Figure 3 displays the first Web Management Interface panel for Layer 3 Switch features, while Figure 4 displays the first panel for Layer 2 switch features. These panels allow you to configure the features supported by the Layer 3 switch and Layer 2 switch software.

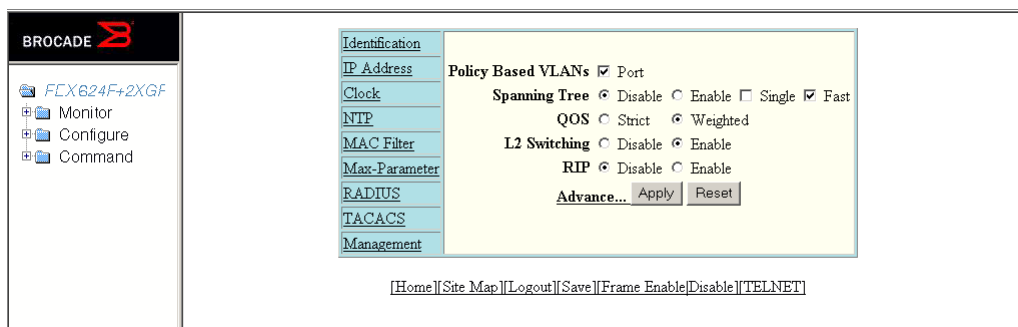
FIGURE 3 First panel for Layer 3 switch features



NOTE

If you are using Internet Explorer 6.0 to view the Web Management Interface, make sure the version you are running includes the latest service packs. Otherwise, the navigation tree (the left-most pane in Figure 3) will not display properly. For information on how to load the latest service packs, refer to the on-line help provided with your Web browser.

FIGURE 4 First panel for Layer 2 switch features



NOTE

If you are using Internet Explorer 6.0 to view the Web Management Interface, make sure the version you are running includes the latest service packs. Otherwise, the navigation tree (the left-most pane in [Figure 3](#)) will not display properly. For information on how to load the latest service packs, refer to the on-line help provided with your Web browser.

The left pane of the Web Management Interface window contains a “tree view,” similar to the one found in Windows Explorer. Configuration options are grouped into folders in the tree view. These folders, when expanded, reveal additional options. To expand a folder, click on the plus sign to the left of the folder icon.

You can configure the appearance of the Web Management Interface by using one of the following methods.

Using the CLI, you can modify the appearance of the Web Management Interface with the **web-management** command.

To cause the Web Management Interface to display the List view by default, enter the **web-management list-menu** command.

```
Brocade(config)#web-management list-menu
```

To disable the front panel frame, enter the **no web-management front-panel** command.

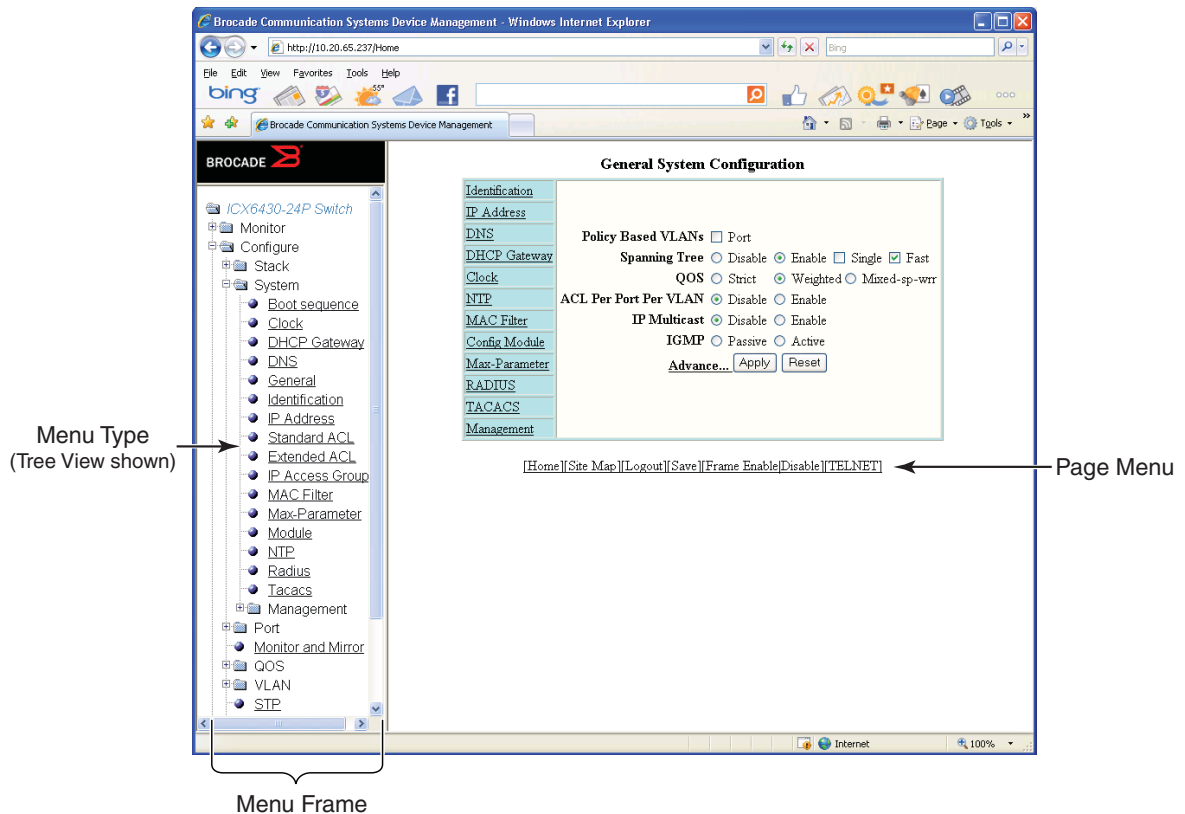
```
Brocade(config)#no web-management front-panel
```

When you save the configuration with the **write memory** command, the changes will take place the next time you start the Web Management Interface, or if you are currently running the Web Management Interface, the changes will take place when you click the Refresh button on your browser.

Using the Web Management Interface

1. Click on the **plus sign** next to Configure in the tree view to expand the list of configuration options.
2. Click on the **plus sign** next to System in the tree view to expand the list of system configuration links.
3. Click on the **plus sign** next to Management in the tree view to expand the list of system management links.
4. Click on the **Web Preference** link to display the Web Management Preferences panel.
5. **Enable** or **disable** elements on the Web Management Interface by clicking on the appropriate radio buttons on the panel. The following figure identifies the elements you can change.

Logging on through the Web Management Interface



NOTE

The tree view is available when you use the Web Management Interface with Netscape 4.0 or higher or Internet Explorer 4.0 or higher browsers. If you use the Web Management Interface with an older browser, the Web Management Interface displays the List view only, and the Web Management Preferences panel does not include an option to display the tree view.

- When you have finished, click the Apply button on the panel, then click the Refresh button on your browser to activate the changes.
- To save the configuration, click the plus sign next to the Command folder, then click the Save to Flash link.

NOTE

The only changes that become permanent are the settings to the Menu Type and the Front Panel Frame. Any other elements you enable or disable will go back to their default settings the next time you start the Web Management Interface.

Logging on through the Web Management Interface

Basic Software Features

Table 8 lists the individual Brocade FastIron switches and the basic software features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 8 Supported basic software features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Basic System Parameters					
System name, contact, and location	Yes	Yes	Yes	Yes	Yes
SNMP trap receiver and trap source address	Yes	Yes	Yes	Yes	Yes
Virtual routing interface statistics via SNMP	Yes	No	No	No	No
Disable Syslog messages and traps for CLI access	Yes	Yes	Yes	Yes	Yes
Cancelling an outbound Telnet session	Yes	Yes	Yes	Yes	Yes
System time using a Simple Network Time Protocol (SNTP) server or local system counter	Yes	Yes	Yes	Yes	Yes
Enabling broadcast mode for SNTP client	No	No	Yes	Yes	Yes
System clock	Yes	Yes	Yes	Yes	Yes
Byte-based broadcast, multicast, and unknown-unicast limits	Yes	No	No	No	No
Packet-based broadcast, multicast, and unknown-unicast limits	Yes	Yes	Yes	Yes	Yes
CLI banners	Yes	Yes	Yes	Yes	Yes
Local MAC address for Layer 2 management traffic	No	Yes	Yes	Yes	Yes
Basic Port Parameters					
Port name	Yes	Yes	Yes	Yes	Yes
10/100/1000 port speed	Yes	Yes	Yes	Yes	Yes
Auto-negotiation	Yes	Yes	Yes	Yes	Yes
Auto-negotiation maximum port speed advertisement and down-shift	Yes	Yes	Yes	Yes	Yes
Duplex mode	Yes	Yes	Yes	Yes	Yes
Auto MDI/MDIX detection	Yes	Yes	Yes	Yes	Yes

TABLE 8 Supported basic software features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Port status (enable or disable)	Yes	Yes	Yes	Yes	Yes
Flow control: <ul style="list-style-type: none"> Responds to flow control packets, but does not generate them 	Yes	Yes	Yes	Yes	Yes
Symmetric flow control <ul style="list-style-type: none"> Can transmit and receive 802.3x PAUSE frames 	No	No	Yes	Yes	Yes
Auto-negotiation and advertisement of flow control	Yes	Yes	Yes	Yes	Yes
PHY FIFO Rx and TX Depth	No	Yes	Yes	Yes	Yes
Interpacket Gap (IPG) adjustment	Yes	Yes	Yes	Yes	Yes
CLI support for 100BaseTX and 100BaseFX	Yes	Yes	Yes	Yes	No
Gbps fiber negotiate mode	Yes	Yes	Yes	Yes	No
QoS priority	Yes	Yes	Yes	Yes	Yes
VOIP autoconfiguration and CDP	Yes	Yes	Yes	Yes	Yes
Port flap dampening	Yes	Yes	Yes	Yes	Yes
Port loop detection	Yes	Yes	Yes	Yes	Yes

Basic system parameter configuration

Brocade devices are configured at the factory with default parameters that allow you to begin using the basic features of the system immediately. However, many of the advanced features such as VLANs or routing protocols for the device must first be enabled at the system (global) level before they can be configured. If you use the Command Line Interface (CLI) to configure system parameters, you can find these system level parameters at the Global CONFIG level of the CLI.

NOTE

Before assigning or modifying any router parameters, you must assign the IP subnet (interface) addresses for each port.

NOTE

For information about configuring IP addresses, DNS resolver, DHCP assist, and other IP-related parameters, refer to [Chapter 26, “IP Configuration”](#).

NOTE

For information about the Syslog buffer and messages, refer to [Appendix A, “Syslog messages”](#).

The procedures in this section describe how to configure the basic system parameters listed in [Table 8](#).

Entering system administration information

You can configure a system name, contact, and location for a Brocade device and save the information locally in the configuration file for future reference. This information is not required for system operation but is suggested. When you configure a system name, the name replaces the default system name in the CLI command prompt.

The name, contact, and location each can be up to 255 alphanumeric characters.

Here is an example of how to configure a system name, system contact, and location.

```
Brocade(config)# hostname zappa
zappa(config)# snmp-server contact Support Services
zappa(config)# snmp-server location Centerville
zappa(config)# end
zappa# write memory
```

Syntax: `hostname <string>`

Syntax: `snmp-server contact <string>`

Syntax: `snmp-server location <string>`

The text strings can contain blanks. The SNMP text strings do not require quotation marks when they contain blanks but the host name does.

NOTE

The **chassis name** command does not change the CLI prompt. Instead, the command assigns an administrative ID to the device.

SNMP parameter configuration

Use the procedures in this section to perform the following configuration tasks:

- Specify a Simple Network Management Protocol (SNMP) trap receiver.
- Specify a source address and community string for all traps sent by the device.
- Change the holddown time for SNMP traps
- Disable individual SNMP traps. (All traps are enabled by default.)
- Disable traps for CLI access that is authenticated by a local user account, a RADIUS server, or a TACACS/TACACS+ server.

NOTE

To add and modify “get” (read-only) and “set” (read-write) community strings, refer to [Chapter 4, “Security Access”](#).

Specifying an SNMP trap receiver

You can specify a trap receiver to ensure that all SNMP traps sent by the Brocade device go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. When you specify the host, you also specify a community string. The Brocade device sends all the SNMP traps to the specified hosts and includes the specified community string. Administrators can therefore filter for traps from a Brocade device based on IP address or community string.

When you add a trap receiver, the software automatically encrypts the community string you associate with the receiver when the string is displayed by the CLI or Web Management Interface. If you want the software to show the community string in the clear, you must explicitly specify this when you add a trap receiver. In either case, the software does not encrypt the string in the SNMP traps sent to the receiver.

To specify the host to which the device sends all SNMP traps, use one of the following methods.

To add a trap receiver and encrypt the display of the community string, enter commands such as the following.

To specify an SNMP trap receiver and change the UDP port that will be used to receive traps, enter a command such as the following.

```
Brocade(config)# snmp-server host 2.2.2.2 0 mypublic port 200
Brocade(config)# write memory
```

Syntax: `snmp-server host <ip-addr> [0 | 1] <string> [port <value>]`

The `<ip-addr>` parameter specifies the IP address of the trap receiver.

The `0 | 1` parameter specifies whether you want the software to encrypt the string (`1`) or show the string in the clear (`0`). The default is `0`.

The `<string>` parameter specifies an SNMP community string configured on the Brocade device. The string can be a read-only string or a read-write string. The string is not used to authenticate access to the trap host but is instead a useful method for filtering traps on the host. For example, if you configure each of your Brocade devices that use the trap host to send a different community string, you can easily distinguish among the traps from different Brocade devices based on the community strings.

The command in the example above adds trap receiver 2.2.2.2 and configures the software to encrypt display of the community string. When you save the new community string to the startup-config file (using the **write memory** command), the software adds the following command to the file.

```
snmp-server host 2.2.2.2 1 <encrypted-string>
```

To add a trap receiver and configure the software to encrypt display of the community string in the CLI and Web Management Interface, enter commands such as the following.

```
Brocade(config)# snmp-server host 2.2.2.2 0 FastIron-12
Brocade(config)# write memory
```

The **port <value>** parameter allows you to specify which UDP port will be used by the trap receiver. This parameter allows you to configure several trap receivers in a system. With this parameter, a network management application can coexist in the same system. Brocade devices can be configured to send copies of traps to more than one network management application.

Specifying a single trap source

You can specify a single trap source to ensure that all SNMP traps sent by the Layer 3 switch use the same source IP address. For configuration details, refer to [“Specifying a single source interface for specified packet types”](#) on page 977

Setting the SNMP trap holddown time

When a Brocade device starts up, the software waits for Layer 2 convergence (STP) and Layer 3 convergence (OSPF) before beginning to send SNMP traps to external SNMP servers. Until convergence occurs, the device might not be able to reach the servers, in which case the messages are lost.

By default, a Brocade device uses a one-minute holddown time to wait for the convergence to occur before starting to send SNMP traps. After the holddown time expires, the device sends the traps, including traps such as “cold start” or “warm start” that occur before the holddown time expires.

You can change the holddown time to a value from one second to ten minutes.

To change the holddown time for SNMP traps, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# snmp-server enable traps holddown-time 30
```

The command in this example changes the holddown time for SNMP traps to 30 seconds. The device waits 30 seconds to allow convergence in STP and OSPF before sending traps to the SNMP trap receiver.

Syntax: [no] snmp-server enable traps holddown-time <secs>

The <secs> parameter specifies the number of seconds and can be from 1 – 600 (ten minutes). The default is 60 seconds.

Disabling SNMP traps

Brocade devices come with SNMP trap generation enabled by default for all traps. You can selectively disable one or more of the following traps.

NOTE

By default, all SNMP traps are enabled at system startup.

SNMP Layer 2 traps

The following traps are generated on devices running Layer 2 software:

- SNMP authentication keys
- Power supply failure
- Fan failure
- Cold start
- Link up
- Link down
- Bridge new root
- Bridge topology change
- Locked address violation

SNMP Layer 3 traps

The following traps are generated on devices running Layer 3 software:

- SNMP authentication key
- Power supply failure
- Fan failure
- Cold start
- Link up
- Link down
- Bridge new root
- Bridge topology change
- Locked address violation
- BGP4
- OSPF
- VRRP
- VRRP-E

To stop link down occurrences from being reported, enter the following.

```
Brocade(config)# no snmp-server enable traps link-down
```

Syntax: [no] snmp-server enable traps <trap-type>

Displaying virtual routing interface statistics

NOTE

This feature is supported on FastIron X Series devices only.

You can enable SNMP to extract and display virtual routing interface statistics from the ifXTable (64-bit counters).

The following describes the limitations of this feature:

- The Brocade device counts traffic from all virtual interfaces (VEs). For example, in a configuration with two VLANs (VLAN 1 and VLAN 20) on port 1, when traffic is sent on VLAN 1, the counters (VE statistics) increase for both VE 1 and VE 20.
- The counters include all traffic on each virtual interface, even if the virtual interface is disabled.
- The counters include traffic that is denied by ACLs or MAC address filters.

To enable SNMP to display VE statistics, enter the **enable snmp ve-statistics** command.

```
Brocade(config)# enable snmp ve-statistics
```

Syntax: [no] enable snmp ve-statistics

Use the **no** form of the command to disable this feature once it is enabled.

Note that the above CLI command enables SNMP to display virtual interface statistics. It does not enable the CLI or Web Management Interface to display the statistics.

Disabling Syslog messages and traps for CLI access

Brocade devices send Syslog messages and SNMP traps when a user logs into or out of the User EXEC or Privileged EXEC level of the CLI. The feature applies to users whose access is authenticated by an authentication-method list based on a local user account, RADIUS server, or TACACS/TACACS+ server.

NOTE

The Privileged EXEC level is sometimes called the “Enable” level, because the command for accessing this level is **enable**.

The feature is enabled by default.

Examples of Syslog messages for CLI access

When a user whose access is authenticated by a local user account, a RADIUS server, or a TACACS or TACACS+ server logs into or out of the CLI User EXEC or Privileged EXEC mode, the software generates a Syslog message and trap containing the following information:

- The time stamp
- The user name
- Whether the user logged in or out
- The CLI level the user logged into or out of (User EXEC or Privileged EXEC level)

NOTE

Messages for accessing the User EXEC level apply only to access through Telnet. The device does not authenticate initial access through serial connections but does authenticate serial access to the Privileged EXEC level. Messages for accessing the Privileged EXEC level apply to access through the serial connection or Telnet.

The following examples show login and logout messages for the User EXEC and Privileged EXEC levels of the CLI.

```
Brocade# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Buffer logging: level ACDMEINW, 12 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning
Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Oct 15 18:01:11:info:dg logout from USER EXEC mode
Oct 15 17:59:22:info:dg logout from PRIVILEGE EXEC mode
Oct 15 17:38:07:info:dg login to PRIVILEGE EXEC mode
Oct 15 17:38:03:info:dg login to USER EXEC mode
```

Syntax: show logging

The first message (the one on the bottom) indicates that user “dg” logged in to the CLI User EXEC level on October 15 at 5:38 PM and 3 seconds (Oct 15 17:38:03). The same user logged into the Privileged EXEC level four seconds later.

The user remained in the Privileged EXEC mode until 5:59 PM and 22 seconds. (The user could have used the CONFIG modes as well. Once you access the Privileged EXEC level, no further authentication is required to access the CONFIG levels.) At 6:01 PM and 11 seconds, the user ended the CLI session.

Disabling the Syslog messages and traps

Logging of CLI access is enabled by default. If you want to disable the logging, enter the following commands.

```
Brocade(config)# no logging enable user-login
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload
```

Syntax: [no] logging enable user-login

Cancelling an outbound Telnet session

If you want to cancel a Telnet session from the console to a remote Telnet server (for example, if the connection is frozen), you can terminate the Telnet session by doing the following.

1. At the console, press **Ctrl+^** (Ctrl+Shift-6).
2. Press the **X** key to terminate the Telnet session.

Pressing **Ctrl+^** twice in a row causes a single **Ctrl+^** character to be sent to the Telnet server. After you press **Ctrl+^**, pressing any key other than **X** or **Ctrl+^** returns you to the Telnet session.

Specifying an SNTP server

The Brocade device can be configured as a Simple Network Time Protocol (SNTP) client. You can configure the Brocade device to consult up to three SNTP servers for the current system time and date. The first server configured will be used unless it becomes unreachable, in which case the Brocade device will attempt to synchronize with the other SNTP servers (if any) in the order in which they were configured.

NOTE

Brocade devices do not retain time and date information across power cycles. Unless you want to reconfigure the system time counter each time the system is reset, Brocade recommends that you use the SNTP feature as described below.

To identify an SNTP server with IP address 208.99.8.95 to act as the clock reference for a Brocade device, enter the following.

```
Brocade(config)# sntp server 208.99.8.95
```

Syntax: [no] sntp server { <ip-address> | <hostname> | ipv6 <ipv6-address> } [<sntp-version>] [authentication-key <key-ID> <key-string>]

The <sntp-version> parameter specifies the SNTP version the server is running and can be from 1 – 4. The default is 4. The SNTP version is automatically set to 4, unless a different SNTP version is specified in the device startup configuration. You can configure up to three SNTP servers by entering three separate **sntp server** commands.

The order in which the SNTP servers are configured is the order in which they are consulted. The server that was configured first is the first server consulted after the poll cycle; the next server will be consulted only if a positive ACK is not received from the first one.

To specify an IPv6 address for the SNTP server, use the **ipv6** option.

The **authentication-key** option allows you to configure an authentication key for communication with the SNTP server. When the authentication key is configured for an SNTP client, it is used only for an SNTP unicast client. You must assign a unique server *<key-ID>* and pre-share *<key-string>*. The *<key-ID>* and pre-share *<key-string>* are used together to create the MD5 checksum. The MD5 checksum is used for authentication for request and reply messages with the SNTP server. The *<key-ID>* is the symmetric key shared with the upstream server, and accepts values from 1 to 4,294,967,295. The *<key-string>* is the authentication string itself, and can take up to 16 characters. If the *<key-string>* variable consists of only numerical characters, you must enclose the numerical characters in double quotes.

Modification of the authentication key fields is not supported. To change the key ID or key string, remove the time server using the **no sntp server...** command, then reconfigure the server with the new key.

By default, the Brocade device polls its SNTP server every 30 minutes (1800 seconds). To configure the Brocade device to poll for clock updates from a SNTP server every 15 minutes, enter the following.

```
Brocade(config)# sntp poll-interval 900
```

Syntax: [no] sntp poll-interval <16-131072>

To display information about SNTP associations, enter the **show sntp associations** command.

```
Brocade# show sntp associations
  address      ref clock      st  when  poll  delay  disp
~207.95.6.102  0.0.0.0         16  202    4    0.0    5.45
~207.95.6.101  0.0.0.0         16  202    0    0.0    0.0
* synced, ~ configured
```

Syntax: show sntp associations

The following table describes the information displayed by the **show sntp associations** command.

TABLE 9 Output from the **show sntp associations** command

Field	Description
(leading character)	One or both of the following: * Synchronized to this peer ~ Peer is statically configured
address	IP address of the peer
ref clock	IP address of the peer reference clock, or the reference ID of the external clock source if the peer is stratum 1. Examples of external clock source IDs: GPS, CDMA, WWV (Ft.Collins US Radio 2.5, 5, 10, 15 MHz), CESM (calibrated Cesium clock), etc.
st	NTP stratum level of the peer
when	Amount of time since the last NTP packet was received from the peer. A negative number indicates the system has never received any synchronization message from the specified server.
poll	The poll interval of the peer relative to the server.

TABLE 9 Output from the **show sntp associations** command (Continued)

Field	Description
delay	The total delay time in milliseconds along the path to the root clock.
disp	The dispersion of the root path in milliseconds.

To display detailed information about SNTP associations, enter the **show sntp associations details** command.

```
Brocade# show sntp associations details
208.99.8.95 configured,insane, unsynched,invalid, stratum 16
ref ID 0.0.0.0,time 0.0 (Jan 1 00:00:00)
our mode client, peer mode unspec, our poll intvl 15, peer poll intvl 0
root delay 0.0 msec, root disp 0.0
delay 0 msec, offset 0 msec
precision 2**0, version 0
org time 0.0 (Jan 1 00:00:00)
rcv time 0.0 (Jan 1 00:00:00)
xmt time 0.0 (Jan 1 00:00:00)
```

Syntax: show sntp associations details

The following table describes the information displayed by the **show sntp associations details** command.

TABLE 10 Output from the **show sntp associations details** command

Field	Description
IP address	The IP address of the SNTP server. The IP address is an IPv4 or an IPv6 address.
configured or dynamic	The SNTP server is either configured, or the last responsive broadcast server that is found dynamically.
authenticated	If MD5 authentication is enabled for the peer.
sane or insane	If the SNTP server passes sanity checks.
synched or unsynched	If the system is synchronized or unsynchronized to the NTP peer.
valid or invalid	If the peer time is valid or invalid.
stratum	The NTP stratum level of the peer.
reference ID	The IP address of the peer (if any) to which the unit is synchronized. The reference ID can also refer to the external clock source if the peer is stratum 1. Examples of external clock source IDs: GPS, CDMA, WWV (Ft.Collins US Radio 2.5, 5, 10, 15 MHz), CESM (calibrated Cesium clock), etc.
time	The reference time stamp.
our mode	The mode relative to the peer. The mode can be a client or a broadcast client.
peer mode	Peer mode relative to us.
our poll intvl	The system poll interval relative to the peer.
peer poll intv	The poll interval of the peer relative to the server.

Field	Description
root delay	The total delay time in milliseconds along the path to the root clock.
root disp	The dispersion of the root path in milliseconds.
delay	The round trip delay to the peer in milliseconds.
offset	The offset of the peer clock relative to the system clock.
precision	The precision of the system clock in Hz.
version	The NTP version of the peer. The version can be from 1 - 4.
org time	The original timestamp of the system clock. The original timestamp is what the client has sent to the server.
rcv time	The receive timestamp of the system clock.
xmt time	The transmit timestamp of the system clock.

To display information about SNTP status, enter the **show sntp status** command.

```
Brocade# show sntp status
Clock is synchronized, stratum = 4, reference clock = 10.70.20.23
precision is 2**-20
reference time is 3489354594.3780510747
clock offset is 0.0000 msec, root delay is 0.41 msec
root dispersion is 0.11 msec, peer dispersion is 0.00 msec
sntp poll-interval is 10 secs
```

Syntax: show sntp status

The following table describes the information displayed by the **show sntp status** command.

TABLE 11 Output from the **show sntp status** command

Field	Description
unsynchronized	System is not synchronized to an NTP peer.
synchronized	System is synchronized to an NTP peer.
stratum	NTP stratum level of the upstream time server.
reference clock	IP address of the peer reference clock, or the reference ID of the external clock source if the peer is stratum 1. Examples of external clock source IDs: GPS, CDMA, WWV (Ft. Collins US Radio 2.5, 5, 10, 15 MHz), CESM (calibrated Cesium clock), etc.
precision	Precision of this system's clock (in Hz)
reference time	Reference time stamp
clock offset	Offset of clock to synchronized peer
root delay	Total delay along the path to the root clock
root dispersion	Dispersion of the root path
peer dispersion	Dispersion of the synchronized peer
sntp poll-interval	Shows how often the Brocade device polls for clock updates from an SNTP server.

Configuring the device as an SNTP server

You can configure the Brocade device to function as an SNTP server to its downstream clients. When using the device as an SNTP server, you can also set it to use its own internal clock as the reference source if an upstream server becomes unavailable.

To use the device as a an SNTP server, enter a command such as the following at the Privileged EXEC level.

```
Brocade(config)# sntp server-mode use-local-clock authentication-key abc123
Brocade(config)# write memory
```

The above example configures the device to operate as an SNTP server with the local clock as a reference backup and an authentication key of “abc123” and writes the configuration changes to memory.

Syntax: [no] sntp server-mode [use-local-clock [stratum <stratum-number>]] [authentication-key <key-string>]

- The **use-local-clock** option causes the Brocade device to use the local clock as a reference source if an upstream reference source becomes unavailable. The SNTP stratum number is set to 1 by default. You may specify a different stratum number using the **stratum** option; <stratum-number> must be between 1 and 15. When the internal clock is serving as the SNTP reference source, the Brocade device will use the specified stratum number (or the default value of 1). When it is synchronized with the upstream server, the Brocade device will use the upstream server’s stratum number plus 1.
If you do not include the **use-local-clock** option the Brocade device will function as specified by RFC 4330: when the Brocade device loses upstream synchronization, it will respond to client SNTP requests with a “kiss-of-death” response (stratum value=0).

NOTE

To enable the **use-local-clock** option, you must set the internal clock of the Brocade device either by SNTP synchronization (see [“Specifying an SNTP server”](#) on page 24) or by using the **clock set** command (see [“Setting the system clock”](#) on page 30). Until the internal clock is set, the Brocade device will continue to rely exclusively on an upstream SNTP server if one is reachable. If none, the SNTP server of the Brocade device is disabled (down).

- To require a code string for authentication of SNTP communication from clients, use the **authentication-key** option and enter a key string of up to 16 characters. When this option is used, authentication parameters are required in clients’ SNTP request messages. If authentication fails, the Brocade device will reply with stratum 0 and a reference ID code of “CRYP” (cryptographic authentication or identification failed), and messages received without the required parameters will be dropped.

NOTE

Once entered, the authentication key cannot be viewed. Using the **show running-config** command will show output similar to the following when an authentication key has been set:

```
sntp server-mode authentication-key 2 $QHMiR3NzQA=
```

The **2** indicates that the key is encrypted using base-64 encryption; the characters following the **2** are the encrypted authentication string.

NOTE

You cannot enable or disable the **use-local-clock** option (or its stratum number) or change the authentication string when the SNTP server is up. To change these settings after enabling SNTP server mode, you must disable server mode using the command **no sntp server-mode**, then re-enable it with the new parameters.

Displaying SNTP server information

Use the **show sntp server-mode** command to display the status of the SNTP server and its configuration.

```
Brocade# show sntp server-mode
Status           : up
Stratum          : 4
Authentication    : md5
Clock source     : 10.50.2.121
Last upstream sync: 15:55:00 Pacific Sun Jul 5 2009

Last 5 unique responses sent to downstream clients :
Client Address      Reference Time
10.1.50.23          16:10:32 Pacific Sun Jul 5 2009
10.1.52.34          15:50:40 Pacific Sun Jul 5 2009
10.1.50.41          10:22:08 Pacific Fri Jul 3 2009
10.1.50.10          06:21:03 Pacific Fri Jul 3 2009
10.1.50.29          21:17:39 Pacific Fri Jul 2 2009
```

Syntax: show sntp server-mode

TABLE 12 Output from the **show sntp server-mode** command

Field	Description
status	The operational state of the SNTP server. “ Up ” means that the SNTP port is open; “ down ” means that the SNTP port is closed. (If sntp server-mode is disabled, the show sntp server-mode command will display the message “SNTP server is not operational.”)
stratum	Stratum number of this server. The range is from 1 through 15. If the device is synchronized to an upstream SNTP server, this will show that server’s stratum number +1. If the device is unsynchronized and using the use-local-clock option, this will show the user-specified stratum number (or the default value of “ 1 ” if no stratum has been configured).
authentication	Authentication key used. If authentication has been configured successfully, this displays “ md5 .” If not, it displays “ none .”
clock source	The source of the reference time. When the reference source is an upstream SNTP server, this will show the IP address of the upstream server. When the internal clock of the device is being used as the reference, this will show “ local-clock .”
last upstream sync	The last upstream time-server synchronization, displayed in timestamp format. This field is not displayed if the time source is the local clock.
last responses sent to clients	The last responses sent to downstream clients (maximum of five unique clients), displayed in reverse chronological order. Each entry shows the IP address of the client and the timestamp sent.

Enabling broadcast mode for an SNTP client

The Brocade device can be configured as an SNTP client. You can enable an SNTP client to function in a broadcast mode when the NTP server is within the same LAN, and the expected delay in response to calibrate the system clock is minimal. In a broadcast mode, the SNTP client will not send queries to the NTP server. The SNTP client will listen to any number of NTP servers on the network until the last message is received from the system clock. To update the system clock with the last message received, you can enable the SNTP client to either listen to all NTP broadcast servers on any interface, or enable the SNTP client to listen to only one specific NTP broadcast server.

To enable an SNTP client in a broadcast mode to listen to all NTP servers on any interface, enter the **sntp broadcast client** command.

```
Brocade(config)#sntp broadcast client
```

Syntax: sntp broadcast client

The **sntp broadcast client** command enables an SNTP client to listen to all NTP servers, and update the client's clock with the last message received from any NTP server.

To enable an SNTP client to listen to only one specific IPv4 NTP broadcast server, enter the following commands.

```
Brocade(config)#sntp broadcast client
Brocade(config)#sntp broadcast server 1.1.1.1
```

To enable an SNTP client to listen to only one specific IPv6 NTP broadcast server, enter the following commands.

```
Brocade(config)#sntp broadcast client
Brocade(config)#sntp broadcast server ipv6 2001:179:2:1::1
```

Syntax: sntp broadcast server [<ip-address> | ipv6 <ipv6-address>]

The **sntp broadcast client** command must be configured with the **sntp broadcast server** command to allow for an SNTP client to listen to only one specific NTP server.

When both unicast and broadcast modes are enabled for an SNTP client, the priority by which the NTP server is used to update the client's clock is as follows.

1. The last responsive unicast server.
2. The broadcast server on any interface.

Setting the system clock

In addition to SNTP support, Brocade switches and routers also allow you to set the system time counter. Using the **clock set** command starts the system clock with the time and date you specify.

NOTE

The time counter setting is not retained across power cycles. For more details about SNTP, refer to [“Specifying an SNTP server”](#) on page 24.

To set the system time and date to 10:15:05 on October 15, 2003, enter the following command.

```
Brocade# clock set 10:15:05 10-15-2003
```

Syntax: [no] clock set <hh:mm:ss> <mm-dd-yy> | <mm-dd-yyyy>

To synchronize the time counter with your SNTP server time, enter the following command.

```
Brocade# sntp sync
```

Syntax: sntp sync

By default, Brocade switches and routers do not change the system time for daylight saving time. To enable daylight saving time, enter the **clock summer-time** command.

```
Brocade(config)# clock summer-time
```

Syntax: [no] clock summer-time

Although SNTP servers typically deliver the time and date in Greenwich Mean Time (GMT), you can configure the Brocade device to adjust the time for any one-hour offset from GMT or for one of the following U.S. time zones:

- US Pacific (default)
- Alaska
- Aleutian
- Arizona
- Central
- East-Indiana
- Eastern
- Hawaii
- Michigan
- Mountain
- Pacific
- Samoa

The default is US Pacific.

To change the time zone to Australian East Coast time (which is normally 10 hours ahead of GMT), enter the **clock timezone gmt** command.

```
Brocade(config)# clock timezone gmt gmt+10
```

Syntax: [no] clock timezone gmt | us <time-zone>

You can enter one of the following values for <time-zone>:

- US time zones (**us**): alaska, aleutian, arizona, central, east-indiana, eastern, hawaii, michigan, mountain, pacific, samoa.
- GMT time zones (**gmt**): gmt+0:00 to gmt+12:00 in increments of 1, and gmt-0:00 to gmt-12:00 in decrements of 1 are supported.

New start and end dates for US daylight saving time

NOTE

This feature applies to US time zones only.

The system will automatically change the system clock to Daylight Saving Time (DST), in compliance with the new federally mandated start of daylight saving time, which is extended one month beginning in 2007. The DST will start at 2:00am on the second Sunday in March and will end at 2:00am on the first Sunday in November.

The DST feature is automatic, but to trigger the device to the correct time, the device must be configured to the US time zone, not the GMT offset. To configure your device to use the US time zone, enter the **clock timezone us pacific** command.

```
Brocade(config)# clock timezone us pacific
```

Syntax: [no] **clock timezone us** <timezone-type>

Enter pacific, eastern, central, or mountain for <timezone-type>.

This command must be configured on every device that follows the US DST.

To verify the change, run a **show clock** command.

```
Brocade# show clock
```

Refer to October 19, 2006 - Daylight Saving Time 2007 Advisory, posted on kp.foundrynet.com for more information.

Limiting broadcast, multicast, and unknown unicast traffic

Brocade devices can forward all flooded traffic at wire speed within a VLAN. However, some third-party networking devices cannot handle high rates of broadcast, multicast, or unknown-unicast traffic. If high rates of traffic are being received by the Brocade device on a given port of that VLAN, you can limit the number of broadcast, multicast, or unknown-unicast packets or bytes received each second on that port. This can help to control the number of such packets or bytes that are flooded on the VLAN to other devices.

Broadcast, multicast, and unknown unicast configuration notes and feature limitations

The following describes feature differences on FastIron devices:

- **FastIron X Series devices, except for the SX-FI48GPP interface module**
 - Unknown unicast limiting is independent of broadcast and multicast limiting. To enable multicast limiting, enable it after enabling broadcast limiting. Multicast limiting uses the limit defined in broadcast limiting. You cannot set a separate limit for multicast limiting.
 - FastIron X Series devices support packet-based and byte-based limiting per port, as well as simultaneously on the same port. For example, you can configure the broadcast limit in packet-based mode and the unknown unicast limit in the byte-based mode on the same port.
 - On FastIron X Series devices, when you configure unknown-unicast limiting, the rate applies to all ports in the **port range** for which unknown unicast is enabled. Also, when you enable multicast limiting, it is enabled on all the ports in the port range for which broadcast limiting is enabled. A 1-Gbps port range consists of 12 ports.
- **SX-FI48GPP interface module**
 - ❑ To enable multicast or unknown-unicast limiting, enable it after enabling broadcast limiting. Multicast and unknown-unicast limiting use the limit defined in broadcast limiting. You cannot set a separate limit for unknown-unicast limiting and multicast limiting.
 - ❑ The SX-FI48GPP module supports packet-based limiting only. It does not support byte-based limiting.
 - ❑ Each port on the SX-FI48GPP module can be configured individually.
- **FastIron WS, Brocade FCX Series, and ICX 6430 devices**
 - To enable unknown-unicast limiting or multicast limiting, enable it after enabling broadcast limiting. Unknown-unicast limiting and multicast limiting use the limit defined in broadcast limiting. You cannot set a separate limit for unknown-unicast limiting and multicast limiting.
 - FastIron WS, Brocade FCX Series, and ICX 6430 devices support packet-based limiting only.
- **Brocade ICX 6610 and ICX 6450 devices support byte based limiting only**

Command syntax for packet-based limiting on FastIron X-Series devices

To enable broadcast limiting on a group of ports by counting the number of packets received, enter commands such as the following.

```
Brocade(config)# interface ethernet 1 to 8
Brocade(config-mif-e1000-1-8)# broadcast limit 65536
```

These commands configure packet-based broadcast limiting on ports 1 – 8. On each port, the maximum number of broadcast packets per second cannot exceed 65,536 packets per second.

To include multicasts in the 65536 packets per second limit on each of the ports, enter the **multicast limit** command after enabling broadcast limiting.

```
Brocade(config-mif-e1000-1-8)# multicast limit
```

To enable unknown unicast limiting by counting the number of packets received, enter commands such as the following.

```
Brocade(config)# interface ethernet 1
Brocade(config-if-e1000-1)# unknown-unicast limit 65536
The combined number of inbound Unknown Unicast packets permitted
    for ports 1 to 12 is now set to 65536
Brocade(config-if-e1000-1)#
```

NOTE

On the SX-FI48GPP module, multicast and unknown-unicast limiting use the value defined in broadcast limiting. You cannot set a separate limit for unknown-unicast limiting and multicast limiting.

Syntax: [no] broadcast limit <num>

Syntax: [no] multicast limit

Syntax: [no] unknown-unicast limit <num>

The <num> variable specifies the maximum number of packets per second. It can be any number that is a multiple of 8192, up to a maximum value of 2147418112. If you enter the **multicast limit** or **unknown-unicast limit** command, multicast packets or unknown-unicast limit are included in the corresponding limit. If you specify 0, limiting is disabled. If you specify a number that is not a multiple of 8192, the software rounds the number to the next multiple of 8192. Limiting is disabled by default.

Command syntax for packet-based limiting on FastIron WS and Brocade FCX Series and ICX 6430 devices

To enable broadcast limiting on a group of ports by counting the number of packets received, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/1 to 1/1/8
Brocade(config-mif-e1000-1/1/1-1/1/8)# broadcast limit 65536
```

To include unknown unicast limiting by counting the number of packets received, enter commands such as the following.

```
Brocade(config-mif-e1000-1/1/1-1/1/8)# unknown-unicast limit
```

To include multicasts limiting, enter the **multicast limit** command after enabling broadcast limiting.

```
Brocade(config-mif-e1000-1-8)# multicast limit
```

Syntax: [no]broadcast limit <num>

Syntax: [no] multicast limit

Syntax: [no] unknown-unicast limit

The <num> variable specifies the maximum number of packets per second. It can be any number that is a multiple of 65536, up to a maximum value of 2147418112. If you enter the **multicast limit** command, multicast packets are included in the corresponding limit. If you specify 0, limiting is disabled. If you specify a number that is not a multiple of 65536, the software rounds the number to the next multiple of 65536. Limiting is disabled by default.

Command syntax for packet-based limiting on Brocade ICX 6610 and 6450 devices

To enable broadcast limiting on a group of ports by counting the number of bytes received, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/1 to 1/1/8
Brocade(config-mif-e1000-1/1/1-1/1/8)# broadcast limit 8192
```

To include unknown-unicast limiting, enter the unknown-unicast limit command after enabling broadcast limiting.

```
Brocade(config-mif-e1000-1/1/1-1/1/8)# unknown-unicast limit
```

To include multicasts limiting, enter the multicast limit command after enabling broadcast limiting.

```
Brocade(config-mif-e1000-1-8)# multicast limit
```

Syntax: [no]broadcast limit <num>

Syntax: [no] multicast limit

Syntax: [no] unknown-unicast limit

The <num> variable specifies the maximum number of Kilo bytes per second. It can be any number that is a multiple of 8192, up to a maximum value of 2147418112. If you enter the **multicast limit** or **unknown-unicast limit** command, multicast or unknown-unicast packets are included in the corresponding limit. If you specify 0, limiting is disabled. If you specify a number that is not a multiple of 8192, the software rounds the number to the next multiple of 8192. Limiting is disabled by default.

Command syntax for byte-based limiting

NOTE

Byte-based limiting is supported on FSX and FESX devices only. Byte-based limiting is not supported on the FastIron WS, Brocade FCX Series, ICX 6430 devices and the SX-FI48GPP module.

Byte-based limiting provides the ability to rate limit traffic based on byte count. When the byte mode is enabled, packets will be received on a port as long as the number of bytes received per second is less than the corresponding limit. Once the limit is reached, further packets will be dropped.

To enable broadcast limiting on a group of ports by counting the number of bytes received, enter commands such as the following.

```
Brocade(config)# interface ethernet 9 to 10
Brocade(config-mif-e1000-9-10)# broadcast limit 131072 bytes
```

These commands configure byte-based broadcast limiting on ports 9 and 10. On each port, the total number of bytes received from broadcast packets cannot exceed 131,072 per second.

To include multicasts in the 131072 bytes per second limit on each of the ports, enter the **multicast limit** command after enabling broadcast limiting.

```
Brocade(config-mif-e1000-1-8)# multicast limit
```

To enable unknown unicast limiting, enter commands such as the following.

Configuring the device as an SNMP server

```
Brocade# config terminal
Brocade(config)# interface ethernet 13
Brocade(config-if-e1000-13)# unknown-unicast limit 65536 bytes
The combined number of bytes of inbound Unknown Unicast packets
    permitted for ports 13 to 24 is now set to 65536
Brocade((config-if-e1000-13)#
```

Syntax: [no] broadcast limit <num> bytes

Syntax: [no] multicast limit

Syntax: [no] unknown-unicast limit <num> bytes

The <num> variable can be any number that is a multiple of 65536, up to a maximum value of 2147418112. If you enter the **multicast limit** command, multicast packets are included in the limit you specify. If you specify 0, limiting is disabled. If you specify a number that is not a multiple of 65536, the software rounds the number to the next multiple of 65536. Limiting is disabled by default. The **unknown-unicast limit <num> bytes** command is supported on FSX and FESX devices.

Viewing broadcast, multicast, and unknown unicast limits

You can use the **show run interface** command to display the broadcast, multicast, and unknown-unicast limits configured on the device.

You can use the following commands, in addition to the **show run interface** command, to display the broadcast, multicast, and unknown-unicast limits configured on the device:

- **show rate-limit unknown-unicast**
- **show rate-limit broadcast**

NOTE

The **show rate-limit unknown-unicast** command is only supported on FSX and FESX devices.

Use the **show run interface** command to view the broadcast, multicast, and unknown-unicast limit configured on each port.

Example

```
Brocade# show run interface
interface ethernet 4
broadcast limit 1245184 bytes
multicast limit
!
interface ethernet 5
broadcast limit 1245184 bytes
multicast limit
!
interface ethernet 12
unknown-unicast limit 524288
!
interface ethernet 13
unknown-unicast limit 65536 bytes
!
interface ethernet 14
broadcast limit 65536
!
interface ethernet 23
broadcast limit 131072
multicast limit
```

Syntax: show run interface

Use the **show rate-limit unknown-unicast** command to display the unknown unicast limit for each port region to which it applies.

Example

```
Brocade# show rate-limit unknown-unicast
Unknown Unicast Limit Settings:
Port Region Combined Limit Packets/Bytes
  1 - 12          524288      Packets
 13 - 24          65536       Bytes
```

Syntax: show rate-limit unknown-unicast

Use the **show rate-limit broadcast** command to display the broadcast limit or broadcast and multicast limit for each port to which it applies.

Example

```
Brocade# show rate-limit broadcast
Broadcast/Multicast Limit Settings:
Port    Limit    Packets/Bytes    Packet Type(s)
  4      1245184      Bytes    Broadcast + Multicast
  5      1245184      Bytes    Broadcast + Multicast
 14        65536      Packets    Broadcast only
 23      131072      Packets    Broadcast + Multicast
```

Syntax: show rate-limit broadcast

CLI banner configuration

Brocade devices can be configured to display a greeting message on users' terminals when they enter the Privileged EXEC CLI level or access the device through Telnet. In addition, a Brocade device can display a message on the Console when an incoming Telnet CLI session is detected.

Setting a message of the day banner

You can configure the Brocade device to display a message on a user terminal when he or she establishes a Telnet CLI session. For example, to display the message "Welcome to FESX!" when a Telnet CLI session is established.

```
Brocade(config)# banner motd $ (Press Return)
Enter TEXT message, End with the character '$'.
Welcome to FESX! $
```

A delimiting character is established on the first line of the **banner motd** command. You begin and end the message with this delimiting character. The delimiting character can be any character except " (double-quotation mark) and cannot appear in the banner text. In this example, the delimiting character is \$ (dollar sign). The text in between the dollar signs is the contents of the banner. The banner text can be up to 4000 characters long, which can consist of multiple lines.

Syntax: [no] banner motd <delimiting-character>

To remove the banner, enter the **no banner motd** command.

NOTE

The **banner** <delimiting-character> command is equivalent to the **banner motd** <delimiting-character> command.

When you access the Web Management Interface, the banner is displayed.



Click the [Login] link to accept and continue the login process..

[\[Login\]](#)

NOTE

If you are using a Web client to view the message of the day, and your banners are very wide, with large borders, you may need to set your PC display resolution to a number greater than the width of your banner. For example, if your banner is 100 characters wide and the display is set to 80 characters, the banner may distort, or wrap, and be difficult to read. If you set your display resolution to 120 characters, the banner will display correctly.

Requiring users to press the Enter key after the message of the day banner

In earlier IronWare software releases, users were required to press the Enter key after the Message of the Day (MOTD) was displayed, prior to logging in to the Brocade device on a console or from a Telnet session. Now, this requirement is disabled by default. Unless configured, users do not have to press Enter after the MOTD banner is displayed.

For example, if the MOTD "Authorized Access Only" is configured, by default, the following messages are displayed when a user tries to access the Brocade device from a Telnet session.

```
Authorized Access Only ...
Username:
```

The user can then login to the device.

However, if the requirement to press the **Enter key** is enabled, the following messages are displayed when accessing the switch from Telnet.

```
Authorized Access Only ...
Press <Enter> to accept and continue the login process....
```

The user must press the **Enter key** before the login prompt is displayed.

Also, on the console, the following messages are displayed if the requirement to press the **Enter key** is disabled.

```
Press Enter key to login
Authorized Access Only ...
User Access Verification
Please Enter Login Name:
```

However, if the requirement to press the **Enter key** after a MOTD is enabled, the following messages are displayed when accessing the switch on the console.

```
Press Enter key to login
Authorized Access Only ...
Press <Enter> to accept and continue the login process....
```

The user must press the Enter key to continue to the login prompt.

To enable the requirement to press the **Enter key** after the MOTD is displayed, enter a command such as the following.

```
Brocade(config)# banner motd require-enter-key
```

Syntax: [no] banner motd require-enter-key

Use the **no** form of the command to disable the requirement.

Setting a privileged EXEC CLI level banner

You can configure the Brocade device to display a message when a user enters the Privileged EXEC CLI level.

Example

```
Brocade(config)# banner exec_mode # (Press Return)
Enter TEXT message, End with the character '#'.
You are entering Privileged EXEC level
Do not foul anything up! #
```

As with the **banner motd** command, you begin and end the message with a delimiting character; in this example, the delimiting character is #(pound sign). The delimiting character can be any character except “ (double-quotation mark) and cannot appear in the banner text. The text in between the pound signs is the contents of the banner. Banner text can be up to 4000 characters, which can consist of multiple lines.

Syntax: [no] banner exec_mode <delimiting-character>

To remove the banner, enter the **no banner exec_mode** command.

Displaying a console message when an incoming Telnet session is detected

You can configure the Brocade device to display a message on the Console when a user establishes a Telnet session. This message indicates where the user is connecting from and displays a configurable text message.

Example

```
Brocade(config)# banner incoming $ (Press Return)
Enter TEXT message, End with the character '$'.
Incoming Telnet Session!! $
```

When a user connects to the CLI using Telnet, the following message appears on the Console.

```
Telnet from 209.157.22.63
Incoming Telnet Session!!
```

As with the **banner motd** command, you begin and end the message with a delimiting character; in this example, the delimiting character is \$(dollar sign). The delimiting character can be any character except " (double-quotation mark) and cannot appear in the banner text. The text in between the dollar signs is the contents of the banner. Banner text can be up to 4000 characters, which can consist of multiple lines.

Syntax: [no] **banner incoming** <delimiting-character>

To remove the banner, enter the **no banner incoming** command.

Local MAC address for Layer 2 management traffic

By default, Brocade Layer 2 devices use the MAC address of the first port as the MAC address for Layer 2 management traffic. For example, when the Brocade device receives an ARP request for its management IP address, it responds with the first port MAC address. This may cause problems in some configurations where the Brocade device uses the same MAC address for management traffic as for switched traffic.

You can configure the Brocade device to use a different MAC address for Layer 2 management traffic than for switched traffic. When you issue the **use-local-management-mac**, the Brocade device changes a local bit in the first port MAC address and uses this MAC address for management traffic. The second bit of the first port MAC address is changed to 2. For example, if the MAC address is 00e0.5201.9900 after the feature is enabled, the switch uses 02e0.5201.9900 for management functions. Switched traffic will continue to use the first port MAC address without the local bit setting.

Example

```
Brocade(config)# use-local-management-mac
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload
```

Syntax: [no] **use-local-management-mac**

NOTE

You must save the configuration and reload the software to place the change into effect.

NOTE

This feature is only available for the switch code. It is not available for router code.

Basic port parameter configuration

The procedures in this section describe how to configure the port parameters shown in [Table 8](#).

All Brocade ports are pre-configured with default values that allow the device to be fully operational at initial startup without any additional configuration. However, in some cases, changes to the port parameters may be necessary to adjust to attached devices or other network requirements.

Assigning a port name

A port name can be assigned to help identify interfaces on the network. You can assign a port name to physical ports, virtual interfaces, and loopback interfaces.

To assign a name to a port.

```
Brocade(config)# interface ethernet 2
Brocade(config-if-e1000-2)# port-name Marsha
```

Syntax: `port-name <text>`

The `<text>` parameter is an alphanumeric string. The name can be up to 64 characters long. The name can contain blanks. You do not need to use quotation marks around the string, even when it contains blanks.

Port speed and duplex mode modification

The Gigabit Ethernet copper ports are designed to auto-sense and auto-negotiate the speed and duplex mode of the connected device. If the attached device does not support this operation, you can manually enter the port speed to operate at either 10, 100, or 1000 Mbps. The default and recommended setting is 10/100/1000 auto-sense.

NOTE

You can modify the port speed of copper ports only; this feature does not apply to fiber ports.

NOTE

For optimal link operation, copper ports on devices that do not support 803.3u must be configured with like parameters, such as speed (10,100,1000), duplex (half, full), MDI/MDIX, and Flow Control.

Port speed and duplex mode configuration syntax

The following commands change the port speed of copper interface 8 on a FastIron from the default of 10/100/1000 auto-sense, to 100 Mbps operating in full-duplex mode.

```
Brocade(config)# interface ethernet 8
Brocade(config-if-e1000-8)# speed-duplex 100-full
```

Syntax: `speed-duplex <value>`

where `<value>` can be one of the following:

- 10-full – 10 Mbps, full duplex
- 10-half – 10 Mbps, half duplex
- 100-full – 100 Mbps, full duplex
- 100-half – 100 Mbps, half duplex
- 1000-full-master – 1 Gbps, full duplex master
- 1000-full-slave – 1 Gbps, full duplex slave
- auto – auto-negotiation

The default is **auto** (auto-negotiation).

Use the **no** form of the command to restore the default.

NOTE

On FastIron devices, when setting the speed and duplex-mode of an interface to 1000-full, configure one side of the link as master (**1000-full-master**) and the other side as slave (**1000-full-slave**).

NOTE

Auto negotiated FESX combo ports may flap for a few seconds before the link is up.

NOTE

On Brocade ICX 6610 and Series devices, after you remove 10 Gbps speed from the running configuration, plugging in a 1G optic SFP transceiver into a 10 Gbps port causes the software to fail to revert the ports back from the default 10G LRM mode to 1 Gbps speed. Remove the 1G SFP transceiver and plug in the 10G optic SFP+transceiver so that the Brocade ICX 6610 devices go into default 10 Gbps LRM mode..

Enabling auto-negotiation maximum port speed advertisement and down-shift

NOTE

For optimal link operation, link ports on devices that do not support 803.3u must be configured with like parameters, such as speed (10,100,1000), duplex (half, full), MDI/MDIX, and Flow Control.

NOTE

Auto negotiated FESX combo ports may flap for a few seconds before the link is up.

Maximum Port speed advertisement and **Port speed down-shift** are enhancements to the auto-negotiation feature, a mechanism for accommodating multi-speed network devices by automatically configuring the highest performance mode of inter-operation between two connected devices.

Port speed down-shift enables Gbps copper ports on the Brocade device to establish a link at 1000 Mbps over a 4-pair wire when possible, or to down-shift to 100 Mbps if the medium is a 2-pair wire.

Maximum port speed advertisement enables you to configure an auto-negotiation maximum speed that Gbps copper ports on the Brocade device will advertise to the connected device. You can configure a port to advertise a maximum speed of either 100 Mbps or 10 Mbps. When the maximum port speed advertisement feature is configured on a port that is operating at 100 Mbps maximum speed, the port will advertise 10/100 Mbps capability to the connected device. Similarly, if a port is configured at 10 Mbps maximum speed, the port will advertise 10 Mbps capability to the connected device.

The port speed down-shift and maximum port speed advertisement features operate dynamically at the physical link layer between two connected network devices. They examine the cabling conditions and the physical capabilities of the remote link, then configure the speed of the link segment according to the highest physical-layer technology that both devices can accommodate.

The port speed down-shift and maximum port speed advertisement features operate dynamically at the physical link layer, independent of logical trunk group configurations. Although Brocade recommends that you use the same cable types and auto-negotiation configuration on all members of a trunk group, you could utilize the auto-negotiation features conducive to your cabling environment. For example, in certain circumstances, you could configure each port in a trunk group to have its own auto-negotiation maximum port speed advertisement or port speed down-shift configuration.

Maximum port speed application notes

- Port speed down-shift and maximum port speed advertisement work only when auto-negotiation is enabled (CLI command **speed-duplex auto**). If auto-negotiation is OFF, the device will reject the port speed down-shift and maximum port speed advertisement configuration.
- When port speed down-shift or maximum port speed advertisement is enabled on a port, the device will reject any configuration attempts to set the port to a forced speed mode (100 Mbps or 1000 Mbps).
- When the port speed down-shift feature is enabled on a combo port, the port will not support true media automatic detection, meaning the device will not be able to detect and select the fiber or copper connector based on link availability.

Enabling port speed down-shift

To enable port speed down-shift on a port that has auto-negotiation enabled, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)# link-config gig copper autoneg-control down-shift ethernet 1
ethernet 2
```

The above command configures Gbps copper ports 1 and 2 to establish a link at 1000 Mbps over a 4-pair wire when possible, or to down-shift (reduce the speed) to 100 Mbps when the medium is a 2-pair wire.

Syntax: [no] link-config gig copper autoneg-control down-shift ethernet <port> [ethernet <port>] | to <port>...

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both.

You can enable port speed down-shift on one or two ports at a time.

To disable port speed down-shift after it has been enabled, enter the **no** form of the command.

Configuring port speed down-shift and auto-negotiation for a range of ports

Port speed down-shift and auto-negotiation can be configured for an entire range of ports with a single command.

For example, to configure down-shift on ports 0/1/1 to 0/1/10 and 0/1/15 to 0/1/20 on the device, enter the following.

```
Brocade(config)# link-config gig copper autoneg-control down-shift ethernet 0/1/1
to 0/1/10 ethernet 0/1/15 to 0/1/20
```

To configure down-shift on ports 5 to 13 and 17 to 19 on a compact switch, enter the following.

```
Brocade(config)# link-config gig copper autoneg-control down-shift ethernet 5 to
13 ethernet 17 to 19
```

Syntax: [no] link-config gig copper autoneg-control [down-shift | 100m-auto | 10m-auto] ethernet <port-list>

The <port-list> is the list of ports to which the command will be applied.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

The output from the **show run** command for this configuration will resemble the following.

```
Brocade# show run
Current configuration:
!
ver 04.0.00b64T7e1
!
module 1 fgs-48-port-management-module
module 2 fgs-cx4-2-port-10g-module
!
link-config gig copper autoneg-control down-shift ethernet 0/1/1 to 0/1/10
ethernet 0/1/15 to 0/1/20
!
!
ip address 10.44.9.11 255.255.255.0
ip default-gateway 10.44.9.1
!
end
```

To disable selective auto-negotiation of 100m-auto on ports 0/1/21 to 0/1/25 and 0/1/30, enter the following.

```
Brocade(config)# no link-config gig copper autoneg-control 100m-auto ethernet
0/1/21 to 0/1/25 ethernet 0/1/30
```

Configuring maximum port speed advertisement

To configure a maximum port speed advertisement of 10 Mbps on a port that has auto-negotiation enabled, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)# link-config gig copper autoneg-control 10m ethernet 1
```

To configure a maximum port speed advertisement of 100 Mbps on a port that has auto-negotiation enabled, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)# link-config gig copper autoneg-control 100m ethernet 2
```

Syntax: [no] link-config gig copper autoneg-control 10m | 100m ethernet <port> [ethernet <port>]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*

- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both.

You can enable maximum port speed advertisement on one or two ports at a time.

To disable maximum port speed advertisement after it has been enabled, enter the **no** form of the command.

Modifying port duplex mode

You can manually configure a 10/100 Mbps port to accept either full-duplex (bi-directional) or half-duplex (uni-directional) traffic.

NOTE

You can modify the port duplex mode of copper ports only. This feature does not apply to fiber ports.

Port duplex mode and port speed are modified by the same command.

Port duplex mode configuration syntax

To change the port speed of interface 8 from the default of 10/100/1000 auto-sense to 10 Mbps operating at full-duplex, enter the following.

```
Brocade(config)# interface ethernet 8
Brocade(config-if-e1000-8)# speed-duplex 10-full
```

Syntax: **speed-duplex** <value>

The <value> can be one of the following:

- 10-full
- 10-half
- 100-full
- 100-half
- auto (default)

MDI and MDIX configuration

Brocade devices support automatic Media Dependent Interface (MDI) and Media Dependent Interface Crossover (MDIX) detection on all Gbps Ethernet Copper ports.

MDI/MDIX is a type of Ethernet port connection using twisted pair cabling. The standard wiring for end stations is MDI, whereas the standard wiring for hubs and switches is MDIX. MDI ports connect to MDIX ports using straight-through twisted pair cabling. For example, an end station connected to a hub or a switch uses a straight-through cable. MDI-to-MDI and MDIX-to-MDIX connections use crossover twisted pair cabling. So, two end stations connected to each other, or two hubs or switches connected to each other, use crossover cable.

The auto MDI/MDIX detection feature can automatically correct errors in cable selection, making the distinction between a straight-through cable and a crossover cable insignificant.

MDI and MDIX configuration notes

- This feature applies to copper ports only.
- The **mdi-mdix mdi** and **mdi-mdix mdix** commands work independently of auto-negotiation. Thus, these commands work whether auto-negotiation is turned ON or OFF.

MDI and MDIX configuration syntax

The auto MDI/MDIX detection feature is enabled on all Gbps copper ports by default. For each port, you can disable auto MDI/MDIX, designate the port as an MDI port, or designate the port as an MDIX port.

To turn off automatic MDI/MDIX detection and define a port as an MDI only port.

```
Brocade(config-if-e1000-2)# mdi-mdix mdi
```

To turn off automatic MDI/MDIX detection and define a port as an MDIX only port.

```
Brocade(config-if-e1000-2)# mdi-mdix mdix
```

To turn on automatic MDI/MDIX detection on a port that was previously set as an MDI or MDIX port.

```
Brocade(config-if-e1000-2)# mdi-mdix auto
```

Syntax: **mdi-mdix** <mdi | mdix | auto>

After you enter the **mdi-mdix** command, the Brocade device resets the port and applies the change.

To display the MDI/MDIX settings, including the configured value and the actual resolved setting (for mdi-mdix auto), enter the command **show interface** at any level of the CLI.

Disabling or re-enabling a port

A port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is enabled.

To disable port 8 of a Brocade device, enter the following.

```
Brocade(config)# interface ethernet 8
Brocade(config-if-e1000-8)# disable
```

You also can disable or re-enable a virtual interface. To do so, enter commands such as the following.

```
Brocade(config)# interface ve v1
Brocade(config-vif-1)# disable
```

Syntax: **disable**

To re-enable a virtual interface, enter the **enable** command at the Interface configuration level. For example, to re-enable virtual interface v1, enter the **enable** command.

```
Brocade(config-vif-1)# enable
```

Syntax: **enable**

Flow control configuration

Flow control (802.3x) is a QoS mechanism created to manage the flow of data between two full-duplex Ethernet devices. Specifically, a device that is oversubscribed (is receiving more traffic than it can handle) sends an 802.3x PAUSE frame to its link partner to temporarily reduce the amount of data the link partner is transmitting. Without flow control, buffers would overflow, packets would be dropped, and data retransmission would be required.

All FastIron devices support *asymmetric* flow control, meaning they can receive PAUSE frames but cannot transmit them. In addition, FCX devices also support *symmetric* flow control, meaning they can both receive and transmit 802.3x PAUSE frames. For details about symmetric flow control, refer to [“Symmetric flow control on FCX devices”](#) on page 49.

Flow control configuration notes

- Auto-negotiation of flow control is not supported on 10 Gbps ports, fiber ports, and copper or fiber combination ports.
- When any of the flow control commands are applied to a port that is up, the port will be disabled and re-enabled.
- For 10 Gbps ports, the **show interface <port>** display shows Flow Control is enabled or Flow Control is disabled, depending on the configuration.
- When flow-control is enabled, the hardware can only advertise PAUSE frames. It does not advertise Asym.

Disabling or re-enabling flow control

You can configure the Brocade device to operate with or without flow control. Flow control is enabled by default globally and on all full-duplex ports. You can disable and re-enable flow control at the Global CONFIG level for all ports. When enabled globally, you can disable and re-enable flow control on individual ports.

To disable flow control, enter the **no flow-control** command.

```
Brocade(config)# no flow-control
```

To turn the feature back on, enter the **flow-control** command.

```
Brocade(config)# flow-control
```

Syntax: [no] flow-control

NOTE

For optimal link operation, link ports on devices that do not support 803.3u must be configured with like parameters, such as speed (10,100,1000), duplex (half, full), MDI/MDIX, and Flow Control.

Negotiation and advertisement of flow control

By default, when flow control is enabled globally and auto-negotiation is ON, flow control is enabled and advertised on 10/100/1000M ports. If auto-negotiation is OFF or if the port speed was configured manually, then flow control is not negotiated with or advertised to the peer. For details about auto-negotiation, refer to [“Port speed and duplex mode modification”](#) on page 41.

To disable flow control capability on a port, enter the following commands.

Basic port parameter configuration

```
Brocade(config)# interface ethernet 0/1/21
Brocade(config-if-e1000-0/1/21)# no flow-control
```

To enable flow control negotiation, enter the following commands.

```
Brocade(config)# interface ethernet 0/1/21
Brocade(config-if-e1000-0/1/21)# flow-control neg-on
```

Syntax: [no] flow-control [neg-on]

- **flow-control** [default] - Enable flow control, flow control negotiation, and advertise flow control
- **no flow-control neg-on** - Disable flow control negotiation
- **no flow-control** - Disable flow control, flow control negotiation, and advertising of flow control

After flow control negotiation is enabled using the **flow-control neg-on** command option, flow control is enabled or disabled depending on the peer advertisement.

Commands may be entered in IF (single port) or MIF (multiple ports at once) mode.

Example

```
Brocade(config)# interface ethernet 0/1/21
Brocade(config-if-e1000-0/1/21)# no flow-control
```

This command disables flow control on port 0/1/21.

```
Brocade(config)# interface ethernet 0/1/11 to 0/1/15
Brocade(config-mif-0/1/11-0/1/15)# no flow-control
```

This command disables flow control on ports 0/1/11 to 0/1/15.

Displaying flow-control status

The **show interface <port>** command displays configuration, operation, and negotiation status where applicable.

For example, on a FastIron Stackable device, issuing the command for 10/100/1000M port 0/1/21 displays the following output.

```
Brocade# show interfaces ethernet 0/1/21
GigabitEthernet0/1/21 is up, line protocol is up
  Hardware is GigabitEthernet, address is 00e0.5204.4014 (bia 00e0.5204.4014)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 1, port is untagged, port state is LISTENING
  BPDU Guard is disabled, Root Protect is disabled
  STP configured to ON, priority is level0
  Flow Control is config enabled, oper enabled, negotiation disabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 96 bit times
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
```

```

0 runts, 0 giants
5 packets output, 320 bytes, 0 underruns
Transmitted 0 broadcasts, 5 multicasts, 0 unicasts
0 output errors, 0 collisions

```

Issuing the **show interface <port>** command on a FSX device displays the following output:

```

Brocade# show interface ethernet 18/1
GigabitEthernet18/1 is up, line protocol is up
  Hardware is GigabitEthernet, address is 0012.f228.0600 (bia 0012.f228.0798)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of 4 L2 VLANs, port is tagged, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0, flow control enabled
  Flow Control is config enabled, oper enabled, negotiation disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  IPG MII 96 bits-time, IPG GMII 96 bits-time
  IP MTU 1500 bytes, encapsulation ethernet
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 848 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  10251 packets output, 1526444 bytes, 0 underruns
  Transmitted 1929 broadcasts, 8293 multicasts, 29 unicasts
  0 output errors, 0 collisions

```

The line highlighted in bold will resemble one of the following, depending on the configuration:

- If flow control negotiation is enabled (and a neighbor advertises “Pause-Not Capable”), the display shows:
Flow Control is config enabled, oper disabled, negotiation enabled
- If flow control negotiation is enabled (and a neighbor advertises “Pause-Capable”), the display shows:
Flow Control is config enabled, oper enabled, negotiation enabled
- If flow control is enabled, and flow control negotiation is disabled, the display shows:
Flow Control is config enabled, oper enabled, negotiation disabled
- If flow control is disabled, the display shows:
Flow control is config disabled, oper disabled

Symmetric flow control on FCX devices

In addition to *asymmetric* flow control, FCX devices support *symmetric* flow control, meaning they can both receive and transmit 802.3x PAUSE frames.

By default on FCX devices, packets are dropped from the end of the queue at the egress port (tail drop mode), when the maximum queue limit is reached. Conversely, when symmetric flow control is enabled, packets are guaranteed delivery since they are managed at the ingress port and no packets are dropped.

Symmetric flow control addresses the requirements of a lossless service class in an Internet Small Computer System Interface (iSCSI) environment. It is supported on FCX standalone units as well as on all FCX units in an IronStack.

About XON and XOFF thresholds

An 802.3x PAUSE frame is generated when the buffer limit at the ingress port reaches or exceeds the port's upper watermark threshold (XOFF limit). The PAUSE frame requests that the sender stop transmitting traffic for a period of time. The time allotted enables the egress and ingress queues to be cleared. When the ingress queue falls below the port's lower watermark threshold (XON limit), an 802.3x PAUSE frame with a quanta of 0 (zero) is generated. The PAUSE frame requests that the sender resume sending traffic normally.

Each 1G and 10G port is configured with a default total number of buffers as well as a default XOFF and XON threshold. The defaults are different for 1G ports versus 10G ports. Also, the default XOFF and XON thresholds are different for jumbo mode versus non-jumbo mode. The defaults are shown in Table 13.

TABLE 13 XON and XOFF default thresholds

	Limit when Jumbo disabled / % of buffer limit	Limit when Jumbo enabled / % of buffer limit
1G ports		
Total buffers	272	272
XOFF	240 / 91%	216 / 82%
XON	200 / 75%	184 / 70%
10G ports		
Total buffers	416	416
XOFF	376 / 91%	336 / 82%
XON	312 / 75%	288 / 70%

If necessary, you can change the total buffer limits and the XON and XOFF default thresholds. Refer to [“Changing the total buffer limits”](#) on page 52 and [“Changing the XON and XOFF thresholds”](#) on page 51, respectively.

Configuration notes and feature limitations for symmetric flow control

Note the following configuration notes and feature limitations before enabling symmetric flow control.

- Symmetric flow control is supported on FCX and ICX devices only. It is not supported on other FastIron models.
- Symmetric flow control is supported on all 1G and 10G data ports on FCX and ICX devices.
- Symmetric flow control is not supported on stacking ports or across units in a stack.
- To use this feature, 802.3x flow control must be enabled globally and per interface on the FCX. By default, 802.3x flow control is enabled, but can be disabled with the **no flow-control** command.

- The following QoS features are not supported together with symmetric flow control:
 - Dynamic buffer allocation (CLI commands **qd-descriptor** and **qd-buffer**)
 - Buffer profiles (CLI command **buffer-profile port-region**)
 - DSCP-based QoS (CLI command **trust dscp**)

NOTE

Although the above QoS features are not supported with symmetric flow control, the CLI will still accept these commands. The last command issued will be the one placed into effect on the device. For example, if **trust dscp** is enabled after **symmetric-flow-control** is enabled, symmetric flow control will be disabled and trust dscp will be placed into effect. Make sure you do not enable incompatible QoS features when symmetric flow control is enabled on the device.

- Head of Line (HOL) blocking may occur when symmetric flow control is enabled. This means that a peer can stop transmitting traffic streams unrelated to the congestion stream.

Enabling and disabling symmetric flow control

By default, symmetric flow control is disabled and tail drop mode is enabled. However, because flow control is enabled by default on all full-duplex ports, these ports will always honor received 802.3x Pause frames, whether or not symmetric flow control is enabled.

To enable symmetric flow control globally on all full-duplex data ports of a standalone unit, enter the **symmetric-flow-control enable** command.

```
Brocade(config)# symmetric-flow-control enable
```

To enable symmetric flow control globally on all full-duplex data ports of a particular unit in an IronStack, enter the **symmetric-flow-control enable <stack-unit>** command.

```
Brocade(config)# symmetric-flow-control enable unit 4
```

Syntax: [no] **symmetric-flow-control enable** [unit <stack-unit>]

The <stack-unit> parameter specifies one of the units in a stacking system. Master/Standby/Members are examples of a stack-unit

To disable symmetric flow control once it has been enabled, use the **no** form of the command.

Changing the XON and XOFF thresholds

This section describes how to change the XON and XOFF thresholds described in [“About XON and XOFF thresholds”](#) on page 50.

To change the thresholds for all 1G ports, enter a command such as the following.

```
Brocade(config)# symmetric-flow-control set 1 xoff 91 xon 75
```

To change the thresholds for all 10G ports, enter a command such as the following.

```
Brocade(config)# symmetric-flow-control set 2 xoff 91 xon 75
```

In the above configuration examples, when the XOFF limit of 91% is reached or exceeded, the Brocade device will send PAUSE frames to the sender telling it to stop transmitting data temporarily. When the XON limit of 75% is reached, the Brocade device will send PAUSE frames to the sender telling it to resume sending data.

Syntax: `symmetric-flow-control set 1 | 2 xoff <%> xon <%>`

symmetric-flow-control set 1 sets the XOFF and XON limits for 1G ports.

symmetric-flow-control set 2 sets the XOFF and XON limits for 10G ports.

For **xoff <%>**, the <%> minimum value is 60% and the maximum value is 95%.

For **xon <%>**, the <%> minimum value is 50% and the maximum value is 90%.

Use the **show symmetric** command to view the default or configured XON and XOFF thresholds. Refer to [“Displaying symmetric flow control status”](#) on page 52.

Changing the total buffer limits

This section describes how to change the total buffer limits described in [“About XON and XOFF thresholds”](#) on page 50. You can change the limits for all 1G ports and for all 10G ports.

To change the total buffer limit for all 1G ports, enter a command such as the following.

```
Brocade(config)# symmetric-flow-control set 1 buffers 320
Total buffers modified, 1G: 320, 10G: 128
```

To change the total buffer limit for all 10G ports, enter a command such as the following.

```
Brocade(config)# symmetric-flow-control set 2 buffers 128
Total buffers modified, 1G: 320, 10G: 128
```

Syntax: `symmetric-flow-control set 1 | 2 buffers <value>`

symmetric-flow-control set 1 buffers <value> sets the total buffer limits for 1G ports. The default <value> is 272. You can specify a number from 64 – 320.

symmetric-flow-control set 2 buffers <value> sets the total buffer limits for 10G ports. The default <value> is 416. You can specify a number from 64 – 1632.

Use the **show symmetric** command to view the default or configured total buffer limits. Refer to [“Displaying symmetric flow control status”](#) on page 52.

Displaying symmetric flow control status

The **show symmetric-flow-control** command displays the status of symmetric flow control as well as the default or configured total buffer limits and XON and XOFF thresholds.

```

Brocade(config)# show symmetric
Symmetric Flow Control Information:
-----
Symmetric Flow Control is enabled on units: 2 3
Buffer parameters:
1G Ports:
    Total Buffers : 272
    XOFF Limit    : 240(91%)
    XON Limit     : 200(75%)
10G Ports:
    Total Buffers : 416
    XOFF Limit    : 376(91%)
    XON Limit     : 312(75%)

```

Syntax: `show symmetric-flow-control`

PHY FIFO Rx and Tx depth configuration

PHY devices on FWS devices contain transmit and receive synchronizing FIFOs to adjust for frequency differences between clocks. The **phy-fifo-depth** command allows you to configure the depth of the transmit and receive FIFOs. There are 4 settings (0-3) with 0 as the default. A higher setting indicates a deeper FIFO.

The default setting works for most connections. However, if the clock differences are greater than the default will handle, CRCs and errors will begin to appear on the ports. Raising the FIFO depth setting will adjust for clock differences.

Brocade recommends that you disable the port before applying this command, and re-enable the port. Applying the command while traffic is flowing through the port can cause CRC and other errors for any packets that are actually passing through the PHY while the command is being applied.

Syntax: `[no] phy-fifo-depth <setting>`

- `<setting>` is a value between 0 and 3. (0 is the default.)

This command can be issued for a single port from the IF config mode or for multiple ports from the MIF config mode.

NOTE

Higher settings give better tolerance for clock differences with the partner phy, but may marginally increase latency as well.

Interpacket Gap (IPG) on a FastIron X Series switch

IPG is the time delay, in bit time, between frames transmitted by the device. You configure IPG at the interface level. The command you use depends on the interface type on which IPG is being configured.

The default interpacket gap is 96 bits-time, which is 9.6 microseconds for 10 Mbps Ethernet, 960 nanoseconds for 100 Mbps Ethernet, 96 nanoseconds for 1 Gbps Ethernet, and 9.6 nanoseconds for 10 Gbps Ethernet.

IPG on a FastIron X series switch configuration notes

- The CLI syntax for IPG differs on FastIron X Series devices compared to FastIron Stackable devices. This section describes the configuration procedures for FastIron X Series devices. For FastIron Stackable devices, refer to [“IPG on FastIron Stackable devices”](#) on page 55.
- IPG configuration commands are based on "port regions". All ports within the same port region should have the same IPG configuration. If a port region contains two or more ports, changes to the IPG configuration for one port are applied to all ports in the same port region. When you enter a value for IPG, the CLI displays the ports to which the IPG configuration is applied.

Example

```
Brocade(config-if-e1000-7/1)# ipg-gmii 120
IPG 120(112) has been successfully configured for ports 7/1 to 7/12
```

- When you enter a value for IPG, the device applies the closest valid IPG value for the port mode to the interface. For example, if you specify 120 for a 1 Gbps Ethernet port in 1 Gbps mode, the device assigns 112 as the closest valid IPG value to program into hardware.

Configuring IPG on a Gbps Ethernet port

On a Gbps Ethernet port, you can configure IPG for 10/100 mode and for Gbps Ethernet mode.

10/100M mode

To configure IPG on a Gbps Ethernet port for 10/100M mode, enter the following command.

```
Brocade(config)# interface ethernet 7/1
Brocade(config-if-e1000-7/1)# ipg-mii 120
IPG 120(120) has been successfully configured for ports 7/1 to 7/12
```

Syntax: [no] ipg-mii <bit time>

Enter 12-124 for <bit time>. The default is 96 bit time.

1G mode

To configure IPG on a Gbps Ethernet port for 1-Gbps Ethernet mode, enter commands such as the following.

```
Brocade(config)# interface ethernet 7/1
Brocade(config-if-e1000-7/1)# ipg-gmii 120
IPG 120(112) has been successfully configured for ports 0/7/1 to 7/12
```

Syntax: [no] ipg-gmii <bit time>

Enter 48 - 112 for <bit time>. The default is 96 bit time.

Configuring IPG on a 10 Gbps Ethernet interface

To configure IPG on a 10 Gbps Ethernet interface, enter commands such as the following.

```
Brocade(config)# interface ethernet 9/1
Brocade(config-if-e10000-9/1)# ipg-xgmii 120
IPG 120(128) has been successfully configured for port 9/1
```

Syntax: [no] ipg-xgmii <bit time>

Enter 96-192 for <bit time>. The default is 96 bit time.

IPG on FastIron Stackable devices

On FWS, FCX, and ICX devices, you can configure an IPG for each port. An IPG is a configurable time delay between successive data packets.

You can configure an IPG with a range from 48-120 bit times in multiples of 8, with a default of 96. The IPG may be set from either the interface configuration level or the multiple interface level.

IPG configuration notes

- The CLI syntax for IPG differs on FastIron Stackable devices compared to FastIron X Series devices. This section describes the configuration procedures for FastIron Stackable devices. For FastIron X Series devices, refer to [“Interpacket Gap \(IPG\) on a FastIron X Series switch”](#) on page 53.
- When an IPG is applied to a trunk group, it applies to all ports in the trunk group. When you are creating a new trunk group, the IPG setting on the primary port is automatically applied to the secondary ports.
- This feature is supported on 10/100/1000M ports.

Configuring IPG on a 10/100/1000M port

To configure an IPG of 112 on Ethernet interface 0/1/21, for example, enter the following command.

```
Brocade(config)# interface ethernet 0/1/21
Brocade(config-if-e1000-0/1/21)# ipg 112
```

For multiple interface levels, to configure IPG for ports 0/1/11 and 0/1/14 through 0/1/17, enter the following commands.

```
Brocade(config)# interface ethernet 0/1/11 ethernet 0/1/14 to 0/1/17
Brocade(config-mif-0/1/11,0/1/14-0/1/17)# ipg 104
```

Syntax: [no] ipg <value>

For *value*, enter a number in the range from 48-120 bit times in multiples of 8. The default is 96.

As a result of the above configuration, the output from the show interface Ethernet 0/1/21 command is as follows.

```
Brocade# show interfaces ethernet 0/1/21
GigabitEthernet 0/1/21 is up, line protocol is up
  Hardware is GigabitEthernet, address is 00e0.5204.4014 (bia 00e0.5204.4014)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  BPDU Guard is disabled, Root Protect is disabled
  STP configured to ON, priority is level0
  Flow Control is config enabled, oper enabled, negotiation disabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 112 bit times
  IP MTU 10222 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 248 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  80 packets output, 5120 bytes, 0 underruns
  Transmitted 0 broadcasts, 80 multicasts, 0 unicasts
  0 output errors, 0 collisions
```

Enabling and disabling support for 100BaseTX

For FastIron X Series devices, you can configure a 1000Base-TX SFP (part number E1MG-TX) to operate at a speed of 100 Mbps. To do so, enter the **100-tx** command at the Interface level of the CLI.

```
Brocade(config-if-e1000-11)# 100-tx
```

After the link is up, it will be in 100M/full-duplex mode, as shown in the following example.

```
Brocade# show interface brief ethernet 11
Port Link State Dupl Speed Trunk Tag Priori MAC Name
11 Up Forward Full 100M None No level10 0000.d213.c74b
```

The **show media** command will display the SFP transceiver as **1G M-TX**.

Syntax: [no] **100-tx**

To disable support, enter the **no** form of the command.

100BaseTX configuration notes

- This feature requires that autonegotiation be enabled on the other end of the link.
- Although combo ports (ports 1 – 4) on Hybrid Fiber (HF) models support the 1000Base-TX SFP, they cannot be configured to operate at 100 Mbps. The 100 Mbps operating speed is supported only with non-combo ports (ports 5-24).
- The FCX624S-F is the only FCX model that supports the 1000Base-TX SFP module, and only on the non-combo ports (ports 5-24). The FCX624S-F does not have a specific command to enable the 1000Base-TX SFP optic at 100 Mbps. You must manually configure it with the **speed-duplex 100-full** command. Refer to “[Port speed and duplex mode configuration syntax](#)” on page 41.
- 1000Base-TX modules must be configured individually, one interface at a time.
- 1000Base-TX modules do not support Digital Optical Monitoring.
- This module requires a Cat5 cable and uses an RJ45 connector.
- Hotswap is supported for this module when it is configured in 100M mode.

Enabling and disabling support for 100BaseFX

Some Brocade devices support 100BaseFX fiber transceivers. After you physically install a 100BaseFX transceiver, you must enter a CLI command to enable it.

NOTE

The CLI syntax for enabling and disabling 100BaseFX support on a FESX Compact device differs from the syntax for Chassis-based and Stackable devices. Follow the appropriate instructions below.

Enabling and disabling support for 100BaseFX is not supported on ICX devices. For information about supported SFP and SFP+ transceivers on ICX devices, refer to the following Brocade website:

http://www.brocade.com/downloads/documents/data_sheets/product_data_sheets/Optics_DS.pdf

Enabling and disabling 100BaseFX on FESX Compact device

This section shows how to enable 100BaseFX on a FESX Compact device.

The Brocade device supports the following types of SFPs for 100BaseFX:

- Multimode SFP – maximum distance is 2 kilometers
- Long Reach (LR) – maximum distance is 40 kilometers
- Intermediate Reach (IR) – maximum distance is 15 kilometers

For information about supported SFP and SFP+ transceivers on FastIron devices, refer to the following Brocade website:

http://www.brocade.com/downloads/documents/data_sheets/product_data_sheets/Optics_DS.pdf

NOTE

Connect the 100BaseFX fiber transceiver *after* configuring both sides of the link. Otherwise, the link could become unstable, fluctuating between up and down states.

To enable 100BaseFX on a fiber port, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)# link-config gig fiber 100base-fx ethernet 4
```

The above command enables 100BaseFX on port 4.

The following command enables 100BaseFX on ports 3 and 4

```
Brocade(config)# link-config gig fiber 100base-fx ethernet 3 ethernet 4
```

Syntax: [no] link-config gig fiber 100base-fx ethernet [<port>] ethernet [<port>]

The <port> variable is a valid port number. You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both.

To disable 100BaseFX support on a fiber port, enter the **no** form of the command. Note that you must disable 100BaseFX support before inserting a different type of module in the same port. Otherwise, the device will not recognize traffic traversing the port.

Enabling and disabling 100BaseFX on Chassis-based and stackable devices

NOTE

The following procedure applies to Stackable devices and to Chassis-based 100/1000 Fiber interface modules only. The CLI syntax for enabling and disabling 100BaseFX support on these devices differs than on a Compact device. Make sure you refer to the appropriate procedures. These are not supported on ICX 6430 and ICX 6450 devices.

FastIron devices support the following types of SFPs for 100BaseFX:

- **Multimode SFP** – maximum distance is 2 kilometers
- **Long Reach (LR)** – maximum distance is 40 kilometers
- **Intermediate Reach (IR)** – maximum distance is 15 kilometers

For information about supported SFP and SFP+ transceivers on FastIron devices, refer to the following Brocade website:

http://www.brocade.com/downloads/documents/data_sheets/product_data_sheets/Optics_DS.pdf

NOTE

Connect the 100BaseFX fiber transceiver *after* configuring both sides of the link. Otherwise, the link could become unstable, fluctuating between up and down states.

To enable support for 100BaseFX on an FSX fiber port or on a Stackable switch, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/6
Brocade(config-if-1/6)# 100-fx
```

The above commands enable 100BaseFX on port 6 in slot 1.

Syntax: [no] 100-fx

To disable 100BaseFX support on a fiber port, enter the **no** form of the command. Note that you must disable 100BaseFX support before inserting a different type of module in the same port. Otherwise, the device will not recognize traffic traversing the port.

Changing the Gbps fiber negotiation mode

The globally configured Gbps negotiation mode is the default mode for all Gbps fiber ports. You can override the globally configured default and set individual ports to the following:

NOTE

Gbps negotiation is not supported on ICX 6430 and ICX 6450 devices.

- **Negotiate-full-auto** – The port first tries to perform a handshake with the other port to exchange capability information. If the other port does not respond to the handshake attempt, the port uses the manually configured configuration information (or the defaults if an administrator has not set the information). This is the default.
- **Auto-Gbps** – The port tries to perform a handshake with the other port to exchange capability information.
- **Negotiation-off** – The port does not try to perform a handshake. Instead, the port uses configuration information manually configured by an administrator.

To change the mode for individual ports, enter commands such as the following.

```
Brocade(config)# interface ethernet 1 to 4
Brocade(config-mif-1-4)# gig-default auto-gig
```

This command overrides the global setting and sets the negotiation mode to auto-Gbps for ports 1 – 4.

Syntax: gig-default neg-full-auto | auto-gig | neg-off

NOTE

When Gbps negotiation mode is turned off (CLI command **gig-default neg-off**), the Brocade device may inadvertently take down both ends of a link. This is a hardware limitation for which there is currently no workaround.

Port priority (QoS) modification

You can give preference to the inbound traffic on specific ports by changing the Quality of Service (QoS) level on those ports. For information and procedures, refer to [Chapter 51, “Quality of Service”](#).

Dynamic configuration of Voice over IP (VoIP) phones

You can configure a FastIron device to automatically detect and re-configure a VoIP phone when it is physically moved from one port to another within the same device. To do so, you must configure a **voice VLAN ID** on the port to which the VoIP phone is connected. The software stores the voice VLAN ID in the port database for retrieval by the VoIP phone.

The dynamic configuration of a VoIP phone works in conjunction with the VoIP phone discovery process. Upon installation, and sometimes periodically, a VoIP phone will query the Brocade device for VoIP information and will advertise information about itself, such as, device ID, port ID, and platform. When the Brocade device receives the VoIP phone query, it sends the voice VLAN ID in a reply packet back to the VoIP phone. The VoIP phone then configures itself within the voice VLAN.

As long as the port to which the VoIP phone is connected has a voice VLAN ID, the phone will configure itself into that voice VLAN. If you change the voice VLAN ID, the software will immediately send the new ID to the VoIP phone, and the VoIP phone will re-configure itself with the new voice VLAN.

VoIP configuration notes

- This feature works with any VoIP phone that:
 - Runs CDP
 - Sends a VoIP VLAN query message
 - Can configure its voice VLAN after receiving the VoIP VLAN reply
- Automatic configuration of a VoIP phone will not work if one of the following applies:
 - You do not configure a voice VLAN ID for a port with a VoIP phone
 - You remove the configured voice VLAN ID from a port without configuring a new one
 - You remove the port from the voice VLAN
- Make sure the port is able to intercept CDP packets (**cdp run** command).
- Some VoIP phones may require a reboot after configuring or re-configuring a voice VLAN ID. For example, if your VoIP phone queries for VLAN information only once upon boot up, you must reboot the VoIP phone before it can accept the VLAN configuration. If your phone is powered by a PoE device, you can reboot the phone by disabling then re-enabling the port.

Enabling dynamic configuration of a Voice over IP (VoIP) phone

You can create a voice VLAN ID for a port, or for a group of ports.

To create a voice VLAN ID for a port, enter commands such as the following.

```
Brocade(config)# interface ethernet 2
Brocade(config-if-e1000-2)# voice-vlan 1001
```

To create a voice VLAN ID for a group of ports, enter commands such as the following.

```
Brocade(config)# interface ethernet 1-8
Brocade(config-mif-1-8)# voice-vlan 1001
```

Syntax: [no] **voice-vlan** <voice-vlan-num>

where <voice-vlan-num> is a valid VLAN ID between 1 – 4095.

To remove a voice VLAN ID, use the **no** form of the command.

Viewing voice VLAN configurations

You can view the configuration of a voice VLAN for a particular port or for all ports.

To view the voice VLAN configuration for a port, specify the port number with the **show voice-vlan** command. The following example shows the command output results.

```
Brocade# show voice-vlan ethernet 2
Voice vlan ID for port 2: 1001
```

The following example shows the message that appears when the port does not have a configured voice VLAN.

```
Brocade# show voice-vlan ethernet 2
Voice vlan is not configured for port 2.
```

To view the voice VLAN for all ports, use the **show voice-vlan** command. The following example shows the command output results.

```
Brocade# show voice-vlan
Port ID      Voice-vlan
2            1001
8            150
15           200
```

Syntax: **show voice-vlan** [ethernet <port>]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both.

Port flap dampening configuration

Port Flap Dampening increases the resilience and availability of the network by limiting the number of port state transitions on an interface.

If the port link state toggles from up to down for a specified number of times within a specified period, the interface is physically disabled for the specified wait period. Once the wait period expires, the port link state is re-enabled. However, if the wait period is set to zero (0) seconds, the port link state will remain disabled until it is manually re-enabled.

Port flap dampening configuration notes

- When a flap dampening port becomes a member of a trunk group, that port, as well as all other member ports of that trunk group, will inherit the primary port configuration. This means that the member ports will inherit the primary port flap dampening configuration, regardless of any previous configuration.
- The Brocade device counts the number of times a port link state toggles from "up to down", and not from "down to up".
- The sampling time or window (the time during which the specified toggle threshold can occur before the wait period is activated) is triggered when the first "up to down" transition occurs.
- "Up to down" transitions include UDLD-based toggles, as well as the physical link state.

Configuring port flap dampening on an interface

This feature is configured at the interface level.

```
Brocade(config)# interface ethernet 2/1
Brocade(config-if-e10000-2/1)# link-error-disable 10 3 10
```

Syntax: [no] **link-error-disable** <toggle-threshold> <sampling-time-in-sec> <wait-time-in-sec>

The <toggle-threshold> is the number of times a port link state goes from up to down and down to up before the wait period is activated. Enter a value from 1 - 50.

The <sampling-time-in-sec> is the amount of time during which the specified toggle threshold can occur before the wait period is activated. The default is 0 seconds. Enter 1 – 65535 seconds.

The <wait-time-in-sec> is the amount of time the port remains disabled (down) before it becomes enabled. Enter a value from 0 – 65535 seconds; 0 indicates that the port will stay down until an administrative override occurs.

Configuring port flap dampening on a trunk

You can configure the port flap dampening feature on the primary port of a trunk using the **link-error-disable** command. Once configured on the primary port, the feature is enabled on all ports that are members of the trunk. You cannot configure port flap dampening on port members of the trunk.

Enter commands such as the following on the primary port of a trunk.

```
Brocade(config)# interface ethernet 2/1
Brocade(config-if-e10000-2/1)# link-error-disable 10 3 10
```

Re-enabling a port disabled by port flap dampening

A port disabled by port flap dampening is automatically re-enabled once the wait period expires; however, if the wait period is set to zero (0) seconds, you must re-enable the port by entering the following command on the disabled port.

```
Brocade(config)# interface ethernet 2/1
Brocade(config-if-e10000-2/1)# no link-error-disable 10 3 10
```

Displaying ports configured with port flap dampening

Ports that have been disabled due to the port flap dampening feature are identified in the output of the **show link-error-disable** command. The following shows an example output.

```
Brocade# show link-error-disable
Port 2/1 is forced down by link-error-disable.
```

Use the **show link-error-disable all** command to display the ports with the port flap dampening feature enabled.

For FastIron Stackable devices, the output of the command shows the following.

```
Brocade# show link-error-disable all
Port8/1 is configured for link-error-disable
      threshold:1, sampling_period:10, waiting_period:0
Port8/2 is configured for link-error-disable
      threshold:1, sampling_period:10, waiting_period:0
Port8/3 is configured for link-error-disable
      threshold:1, sampling_period:10, waiting_period:0
Port8/4 is configured for link-error-disable
      threshold:1, sampling_period:10, waiting_period:0
Port8/5 is configured for link-error-disable
      threshold:4, sampling_period:10, waiting_period:2
Port8/9 is configured for link-error-disable
      threshold:2, sampling_period:20, waiting_period:0
```

For FastIron X Series devices, the output of the command shows the following.

```
Brocade# show link-error-disable all
Port  # Threshold Sampling-Time Shutoff-Time State Counter
-----
  11           3           120           600   Idle   N/A
  12           3           120           500   Down   424
```

[Table 14](#) defines the port flap dampening statistics displayed by the **show link-error-disable all** command.

TABLE 14 Output of show link-error-disable

Column	Description
Port #	The port number.
Threshold	The number of times the port link state will go from up to down and down to up before the wait period is activated.
Sampling-Time	The number of seconds during which the specified toggle threshold can occur before the wait period is activated.
Shutoff-Time	The number of seconds the port will remain disabled (down) before it becomes enabled. A zero (0) indicates that the port will stay down until an administrative override occurs.

TABLE 14 Output of show link-error-disable (Continued)

Column	Description
State	<p>The port state can be one of the following:</p> <ul style="list-style-type: none"> • Idle – The link is normal and no link state toggles have been detected or sampled. • Down – The port is disabled because the number of sampled errors exceeded the configured threshold. • Err – The port sampled one or more errors.
Counter	<ul style="list-style-type: none"> • If the port state is Idle, this field displays N/A. • If the port state is Down, this field shows the remaining value of the shutoff timer. • If the port state is Err, this field shows the number of errors sampled.

Syntax: `show link-error-disable [all]`

Also, in FastIron X Series devices, the **show interface** command indicates if the port flap dampening feature is enabled on the port.

Example

```

Brocade# show interface ethernet 15
GigabitEthernet15 is up, line protocol is up
  Link Error Dampening is Enabled
  Hardware is GigabitEthernet, address is 00e0.5200.010e (bia 00e0.5200.010e)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX

Brocade# show interface ethernet 17
GigabitEthernet17 is ERR-DISABLED, line protocol is down
  Link Error Dampening is Enabled
  Hardware is GigabitEthernet, address is 00e0.5200.010e (bia 00e0.5200.010e)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown

```

The line “Link Error Dampening” displays “Enabled” if port flap dampening is enabled on the port or “Disabled” if the feature is disabled on the port. The feature is enabled on the ports in the two examples above. Also, the characters “ERR-DISABLED” is displayed for the “GbpsEthernet” line if the port is disabled because of link errors.

Syntax: `show interface ethernet <port-number>`

In addition to the show commands above, the output of the **show interface brief** command for FastIron X Series indicates if a port is down due to link errors.

Example

```

Brocade# show interface brief e17

Port  Link    State    Dupl Speed Trunk Tag Priori MAC          Name
17    ERR-DIS None    None None  15    Yes level0 00e0.5200.010e

```

The ERR-DIS entry under the “Link” column indicates the port is down due to link errors.

NOTE

If a port name is longer than five characters, the port name is truncated in the output of the **show interface brief** command.

Syslog messages for port flap dampening

The following Syslog messages are generated for port flap dampening.

- If the threshold for the number of times that a port link toggles from “up” to “down” then “down” to “up” has been exceeded, the following Syslog message is displayed.

```
0d00h02m10s:I:ERR_DISABLE: Link flaps on port ethernet 16 exceeded threshold;
port in err-disable state
```
- If the wait time (port is down) expires and the port is brought up the following Syslog message is displayed.

```
0d00h02m41s:I:ERR_DISABLE: Interface ethernet 16, err-disable recovery timeout
```

Port loop detection

This feature allows the Brocade device to disable a port that is on the receiving end of a loop by sending test packets. You can configure the time period during which test packets are sent.

Types of loop detection

There are two types of loop detection; Strict Mode and Loose Mode. In Strict Mode, a port is disabled only if a packet is looped back to that same port. Strict Mode overcomes specific hardware issues where packets are echoed back to the input port. In Strict Mode, loop detection must be configured on the physical port.

In Loose Mode, loop detection is configured on the VLAN of the receiving port. Loose Mode disables the receiving port if packets originate from any port or VLAN on the same device. The VLAN of the receiving port must be configured for loop detection in order to disable the port.

Recovering disabled ports

Once a loop is detected on a port, it is placed in Err-Disable state. The port will remain disabled until one of the following occurs:

- You manually disable and enable the port at the Interface Level of the CLI.
- You enter the command **clear loop-detection**. This command clears loop detection statistics and enables all Err-Disabled ports.
- The device automatically re-enables the port. To set your device to automatically re-enable Err-Disabled ports, refer to [“Configuring the device to automatically re-enable ports”](#) on page 67.

Port loopback detection configuration notes

- Loopback detection packets are sent and received on both tagged and untagged ports. Therefore, this feature cannot be used to detect a loop across separate devices.

The following information applies to Loose Mode loop detection:

- With Loose Mode, two ports of a loop are disabled.
- Different VLANs may disable different ports. A disabled port affects every VLAN using it.
- Loose Mode floods test packets to the entire VLAN. This can impact system performance if too many VLANs are configured for Loose Mode loop detection.

NOTE

Brocade recommends that you limit the use of Loose Mode. If you have a large number of VLANs, configuring loop detection on all of them can significantly affect system performance because of the flooding of test packets to all configured VLANs. An alternative to configuring loop detection in a VLAN-group of many VLANs is to configure a separate VLAN with the same tagged port and configuration, and enable loop detection on this VLAN only.

NOTE

When loop detection is used with L2 loop prevention protocols, such as spanning tree (STP), the L2 protocol takes higher priority. Loop detection cannot send or receive probe packets if ports are blocked by L2 protocols, so it does not detect L2 loops when STP is running because loops within a VLAN have been prevented by STP. Loop detection running in Loose Mode can detect and break L3 loops because STP cannot prevent loops across different VLANs. In these instances, the ports are not blocked and loop detection is able to send out probe packets in one VLAN and receive packets in another VLAN. In this way, loop detection running in Loose Mode disables both ingress and egress ports.

Enabling loop detection

Use the **loop-detection** command to enable loop detection on a physical port (Strict Mode) or a VLAN (Loose Mode). Loop detection is disabled by default. The following example shows a Strict Mode configuration.

```
Brocade(config)# interface ethernet 1/1
Brocade(config-if-e1000-1/1)# loop-detection
```

The following example shows a Loose Mode configuration.

```
Brocade(config)# vlan20
Brocade(config-vlan-20)# loop-detection
```

By default, the port will send test packets every one second, or the number of seconds specified by the **loop-detection-interval** command. Refer to [“Configuring a global loop detection interval”](#) on page 66.

Syntax: [no] loop-detection

Use the [no] form of the command to disable loop detection.

Configuring a global loop detection interval

The loop detection interval specifies how often a test packet is sent on a port. When loop detection is enabled, the loop detection time unit is 0.1 second, with a default of 10 (one second). The range is from 1 (one tenth of a second) to 100 (10 seconds). You can use the **show loop-detection status** command to view the loop detection interval.

To configure the global loop detection interval, enter a command similar to the following.

```
Brocade(config)# loop-detection-interval 50
```


This command sets the loop-detection interval to 5 seconds (50 x 0.1).

To revert to the default global loop detection interval of 10, enter one of the following.

```
Brocade(config)# loop-detection-interval 10
```

OR

```
Brocade(config)# no loop-detection-interval 50
```

Syntax: [no] **loop-detection-interval** <number>

where <number> is a value from 1 to 100. The system multiplies your entry by 0.1 to calculate the interval at which test packets will be sent.

Configuring the device to automatically re-enable ports

To configure the Brocade device to automatically re-enable ports that were disabled because of a loop detection, enter the **errdisable recovery cause loop-detection** command.

```
Brocade(config)# errdisable recovery cause loop-detection
```

The above command will cause the Brocade device to automatically re-enable ports that were disabled because of a loop detection. By default, the device will wait 300 seconds before re-enabling the ports. You can optionally change this interval to a value from 10 to 65535 seconds. Refer to [“Specifying the recovery time interval”](#) on page 67.

Syntax: [no] **errdisable recovery cause loop-detection**

Use the [no] form of the command to disable this feature.

Specifying the recovery time interval

The recovery time interval specifies the number of seconds the Brocade device will wait before automatically re-enabling ports that were disabled because of a loop detection. (Refer to [“Configuring the device to automatically re-enable ports”](#) on page 67.) By default, the device will wait 300 seconds. To change the recovery time interval, enter a command such as the following.

```
Brocade(config)# errdisable recovery interval 120
```

The above command configures the device to wait 120 seconds (2 minutes) before re-enabling the ports.

To revert back to the default recovery time interval of 300 seconds (5 minutes), enter one of the following commands.

```
Brocade(config)# errdisable recovery interval 300
```

OR

```
Brocade(config)# no errdisable recovery interval 120
```

Syntax: [no] **errdisable recovery interval** <seconds>

where <seconds> is a number from 10 to 65535.

Clearing loop-detection

To clear loop detection statistics and re-enable all ports that are in Err-Disable state because of a loop detection, enter the **clear loop-detection** command.

```
Brocade# clear loop-detection
```

Displaying loop-detection information

Use the **show loop-detection status** command to display loop detection status, as shown.

```
Brocade# show loop-detection status
loop detection packets interval: 10 (unit 0.1 sec)
Number of err-disabled ports: 3
You can re-enable err-disable ports one by one by "disable" then "enable"
under interface config, re-enable all by "clear loop-detect", or
configure "errdisable recovery cause loop-detection" for automatic recovery
```

index	port/vlan	status	#errdis	sent-pkts	recv-pkts
1	1/13	untag, LEARNING	0	0	0
2	1/15	untag, BLOCKING	0	0	0
3	1/17	untag, DISABLED	0	0	0
4	1/18	ERR-DISABLE by itself	1	6	1
5	1/19	ERR-DISABLE by vlan 12	0	0	0
6	vlan12	2 ERR-DISABLE ports	2	24	2

If a port is errdisabled in Strict mode, it shows "ERR-DISABLE by itself". If it is errdisabled due to its associated vlan, it shows "ERR-DISABLE by vlan ?"

The following command displays the current disabled ports, including the cause and the time.

```
Brocade# show loop-detection disable
Number of err-disabled ports: 3
You can re-enable err-disable ports one by one by "disable" then "enable"
under interface config, re-enable all by "clear loop-detect", or
configure "errdisable recovery cause loop-detection" for automatic recovery
```

index	port	caused-by	disabled-time
1	1/18	itself	00:13:30
2	1/19	vlan 12	00:13:30
3	1/20	vlan 12	00:13:30

This example shows the disabled ports, the cause, and the time the port was disabled. If loop-detection is configured on a physical port, the disable cause will show "itself". For VLANs configured for loop-detection, the cause will be a VLAN.

The following command shows the hardware and software resources being used by the loop-detection feature.

```
Vlans configured loop-detection use 1 HW MAC
Vlans not configured but use HW MAC: 1 10
```

	alloc	in-use	avail	get-fail	limit	get-mem	size	init
configuration pool	16	6	10	0	3712	6	15	16
linklist pool	16	10	6	0	3712	10	16	16

Displaying loop detection resource information

Use the **show loop-detection resource** command to display the hardware and software resource information on loop detection.

```
Brocade# show loop-detection resource
Vlans configured loop-detection use 1 HW MAC
Vlans not configured but use HW MAC: 1 10
```

	alloc	in-use	avail	get-fail	limit	get-mem	size	init
configuration pool	16	6	10	0	3712	6	15	16
linklist pool	16	10	6	0	3712	10	16	16

Syntax: **show loop-detection resource**

Table 15 describes the output fields for this command.

TABLE 15 Field definitions for the **show loop-detection resource** command

Field	Description
This command displays the following information for the configuration pool and the linklist pool.	
alloc	Memory allocated
in-use	Memory in use
avail	Available memory
get-fail	The number of get requests that have failed
limit	The maximum memory allocation
get-mem	The number of get-memory requests
size	The size
init	The number of requests initiated

Displaying loop detection configuration status on an interface

Use the **show interface** command to display the status of loop detection configuration on a particular interface.

```
Brocade# show interface ethernet 2/1
10GigabitEthernet2/1 is up, line protocol is up
Hardware is 10GigabitEthernet, address is 0024.3889.1100 (bia 0024.3889.1118)
Configured speed 10Gbit, actual 10Gbit, configured duplex fdx, actual fdx
Member of 9 L2 VLANs, port is tagged, port state is FORWARDING
BPDU guard is Disabled, ROOT protect is Disabled
Link Error Dampening is Disabled
STP configured to ON, priority is level0
Loop Detection is ENABLED
Flow Control is enabled
Mirror disabled, Monitor disabled
Member of active trunk ports 2/1,2/2, primary port
Member of configured trunk ports 2/1,2/2, primary port
No port name
IPG XGMII 96 bits-time
MTU 1500 bytes, encapsulation ethernet
ICL port for BH1 in cluster id 1
300 second input rate: 2064 bits/sec, 3 packets/sec, 0.00% utilization
300 second output rate: 768 bits/sec, 1 packets/sec, 0.00% utilization
171319 packets input, 12272674 bytes, 0 no buffer
```

Basic port parameter configuration

```
Received 0 broadcasts, 63650 multicasts, 107669 unicasts  
0 input errors, 0 CRC, 0 frame, 0 ignored  
0 runts, 0 giants  
51094 packets output, 3925313 bytes, 0 underruns  
Transmitted 2 broadcasts, 42830 multicasts, 8262 unicasts  
0 output errors, 0 collisions  
Relay Agent Information option: Disabled
```

Syslog message due to disabled port in loop detection

The following message is logged when a port is disabled due to loop detection. This message also appears on the console.

```
loop-detect: port ?\?\? vlan ?, into errdisable state
```

The Errdisable function logs a message whenever it re-enables a port.

Operations, Administration, and Maintenance

Table 16 lists the individual Brocade FastIron switches and the operations, administration, and maintenance (OAM) features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 16 Supported operations, administration, and maintenance features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Flash and boot code verification	Yes	Yes	Yes	Yes	Yes
Flash image verification	Yes	Yes	Yes	Yes	Yes
Software upgrade via CLI	Yes	Yes	Yes	Yes	Yes
Software upgrade via SNMP	Yes	Yes	Yes	Yes	Yes
Hitless management: <ul style="list-style-type: none"> • Hitless switchover • Hitless failover • Hitless OS upgrade 	Yes (FSX 800 and FSX 1600 only)	No	Yes (Hitless switchover and Hitless failover only; Hitless OS upgrade is not supported) Refer to “ Hitless stacking ” on page 329		
Hitless support: <ul style="list-style-type: none"> • PBR • GRE Tunnels • IPv6 to IPv4 Tunnels 	Yes (FSX 800 and FSX 1600 only)	No	Yes (PBR only)	Yes (PBR only)	No
Boot code synchronization for active and redundant management modules	Yes	No	No	No	No
Software reboot	Yes	Yes	Yes	Yes	Yes
Show boot preference	Yes	Yes	Yes	Yes	Yes
Load and save configuration files	Yes	Yes	Yes	Yes	Yes
System reload scheduling	Yes	Yes	Yes	Yes	Yes
Diagnostic error codes and remedies for TFTP transfers	Yes	Yes	Yes	Yes	Yes
IPv4 ping	Yes	Yes	Yes	Yes	Yes
IPv4 traceroute	Yes	Yes	Yes	Yes	Yes

OAM Overview

For easy software image management, all Brocade devices support the download and upload of software images between the flash modules on the devices and a Trivial File Transfer Protocol (TFTP) server on the network.

Brocade devices have two flash memory modules:

- **Primary flash** – The default local storage device for image files and configuration files.
- **Secondary flash** – A second flash storage device. You can use the secondary flash to store redundant images for additional booting reliability or to preserve one software image while testing another one.

Only one flash device is active at a time. By default, the primary image will become active upon reload.

You can update the software contained on a flash module using TFTP to copy the update image from a TFTP server onto the flash module. In addition, you can copy software images and configuration files from a flash module to a TFTP server.

NOTE

Brocade devices are TFTP clients but not TFTP servers. You must perform the TFTP transaction from the Brocade device. You cannot “put” a file onto the Brocade device using the interface of your TFTP server.

NOTE

If you are attempting to transfer a file using TFTP but have received an error message, refer to [“Diagnostic error codes and remedies for TFTP transfers”](#) on page 93.

Software versions installed and running on a device

Use the following methods to display the software versions running on the device and the versions installed in flash memory.

Determining the flash image version running on the device

To determine the flash image version running on a device, enter the **show version** command at any level of the CLI. Some examples are shown below.

Compact devices

To determine the flash image version running on a Compact device, enter the **show version** command at any level of the CLI. The following shows an example output.

```
Brocade#show version
Copyright (c) 1996-2012 Brocade Communications Systems, Inc. All rights reserved.
  UNIT 1: compiled on Mar  2 2012 at 12:38:17 labeled as ICX64S07400
          (10360844 bytes) from Primary ICX64S07400.bin
  SW: Version 07.4.00T311
  Boot-Monitor Image size = 774980, Version:07.4.00T310 (kxz07400)
  HW: Stackable ICX6450-24
=====
UNIT 1: SL 1: ICX6450-24 24-port Management Module
      Serial #: BZS0442G00G
      License: BASE_SOFT_PACKAGE (LID: dbuFJJHiFFi)
      P-ENGINE 0: type DEF0, rev 01
=====
UNIT 1: SL 2: ICX6450-SFP-Plus 4port 40G Module
=====
      800 MHz ARM processor ARMv5TE, 400 MHz bus
      65536 KB flash memory
```

```
512 MB DRAM
STACKID 1  system uptime is 3 minutes 39 seconds
The system : started=warm start  reloaded=by "reload"
```

The version information is shown in bold type in this example:

- “03.0.00T53” indicates the flash code version number. The “T53” is used by Brocade for record keeping.
- “labeled as FER03000” indicates the flash code image label. The label indicates the image type and version and is especially useful if you change the image file name.
- “Primary fer03000.bin” indicates the flash code image file name that was loaded.

Displaying flash image version on chassis devices

To determine the flash image version running on a chassis device, enter the **show version** command at any level of the CLI. The following is an example output.

```

Brocade#show version
=====
Active Management CPU [Slot-9]:
  SW: Version 07.4.00T3e3 Copyright (c) 1996-2012 Brocade Communications Systems,
Inc. All rights reserved.
    Compiled on Mar 02 2012 at 11:54:29 labeled as SXR07400
    (4585331 bytes) Primary /GA/SXR07400.bin
    BootROM: Version 07.2.00T3e5 (FEv2)
    Chassis Serial #: B32505H001
      License: SX_V6_HW_ROUTER_IPv6_SOFT_PACKAGE (LID: yGFJGOiFLd)
    HW: Chassis FastIron SX 800-PREM6 (PROM-TYPE SX-FIL3U-6-IPV6)
=====
Standby Management CPU [Slot-10]:
  SW: Version 07.4.00T3e3 Copyright (c) 1996-2012 Brocade Communications Systems,
Inc. All rights reserved.
    Compiled on Mar 02 2012 at 11:54:29 labeled as SXR07400
    BootROM: Version 07.2.00T3e5 (FEv2)
    HW: Chassis FastIron SX 800-PREM6 (PROM-TYPE SX-FIL3U-6-IPV6)
=====
SL 1: SX-FI-8XG 8-port 10G Fiber
  Serial #: BQK0421G003
  P-ASIC 0: type C341, rev 00 subrev 00
=====
SL 2: SX-FI-24GPP 24-port Gig Copper + PoE+
  Serial #: BTU0304G01J
  P-ASIC 2: type C300, rev 00 subrev 00
=====
SL 8: SX-FI-48GPP 48-port Gig Copper + PoE+
  Serial #: BFV0431F00P
  P-ASIC 14: type C300, rev 00 subrev 00
=====
SL 9: SX-FIZMR6 0-port Management
  Serial #: W10419G06B
    License: SX_V6_HW_ROUTER_IPv6_SOFT_PACKAGE (LID: yGFJGOiFLd)
=====
SL 10: SX-FIZMR6 0-port Management
  Serial #: W10419G052
    License: SX_V6_HW_ROUTER_IPv6_SOFT_PACKAGE (LID: yyyyyy)
=====
Active Management Module:
  660 MHz Power PC processor 8541 (version 0020/0020) 66 MHz bus

```

Software versions installed and running on a device

```
512 KB boot flash memory
16384 KB code flash memory
512 MB DRAM
Standby Management Module:
660 MHz Power PC processor 8541 (version 0020/0020) 66 MHz bus
512 KB boot flash memory
16384 KB code flash memory
512 MB DRAM
The system uptime is 1 minutes 2 seconds
The system : started=warm start    reloaded=by "reload"
```

The version information is shown in bold type in this example:

- “03.1.00aT3e3” indicates the flash code version number. The “T3e3” is used by Brocade for record keeping.
- “labeled as SXR03100a” indicates the flash code image label. The label indicates the image type and version and is especially useful if you change the image file name.
- “Primary SXR03100a.bin” indicates the flash code image file name that was loaded.

Displaying the boot image version running on the device

To determine the boot image running on a device, enter the **show flash** command at any level of the CLI. The following shows an example output.

```
Brocade#show flash
Active Management Module (Slot 9):
Compressed Pri Code size = 3613675, Version 03.1.00aT3e3 (sxr03100a.bin)
Compressed Sec Code size = 2250218, Version 03.1.00aT3e1 (sxs03100a.bin)
Compressed BootROM Code size = 524288, Version 03.0.01T3e5
Code Flash Free Space = 9699328
Standby Management Module (Slot 10):
Compressed Pri Code size = 3613675, Version 03.1.00aT3e3 (sxr03100a.bin)
Compressed Sec Code size = 2250218, Version 03.1.00aT3e1 (sxs03100a.bin)
Compressed BootROM Code size = 524288, Version 03.0.01T3e5
Code Flash Free Space = 524288
```

The boot code version is shown in bold type.

Displaying the image versions installed in flash memory

Enter the **show flash** command to display the boot and flash images installed on the device. An example of the command output is shown in [“Displaying the boot image version running on the device”](#) on page 74:

- The “Compressed Pri Code size” line lists the flash code version installed in the primary flash area.
- The “Compressed Sec Code size” line lists the flash code version installed in the secondary flash area.
- The “Boot Monitor Image size” line lists the boot code version installed in flash memory. The device does not have separate primary and secondary flash areas for the boot image. The flash memory module contains only one boot image.

NOTE

To minimize the boot-monitor image size on FastIron devices, the **ping** and **tftp** operations performed in the boot-monitor mode are restricted to copper ports on the FastIron Chassis management modules and to copper ports on the FastIron stackable switch combination copper and fiber ports. The fiber ports on these devices do not have the ability to **ping** or **tftp** from the boot-monitor mode.

Flash image verification

The Flash Image Verification feature allows you to verify boot images based on hash codes, and to generate hash codes where needed. This feature lets you select from three data integrity verification algorithms:

- **MD5** - Message Digest algorithm (RFC 1321)
- **SHA1** - US Secure Hash Algorithm (RFC 3174)
- **CRC** - Cyclic Redundancy Checksum algorithm

Flash image CLI commands

Use the following command syntax to verify the flash image:

Syntax: **verify md5 | sha1 | crc32** <ASCII string> | **primary** | **secondary** [<hash code>]

- **md5** – Generates a 16-byte hash code
- **sha1** – Generates a 20-byte hash code
- **crc32** – Generates a 4 byte checksum
- **ascii string** – A valid image filename
- **primary** – The primary boot image (primary.img)
- **secondary** – The secondary boot image (secondary.img)
- **hash code** – The hash code to verify

The following examples show how the **verify** command can be used in a variety of circumstances.

To generate an MD5 hash value for the secondary image, enter the following command.

```
Brocade#verify md5 secondary
Brocade#.....Done
Size = 2044830, MD5 01c410d6d153189a4a5d36c955653862
```

To generate a SHA-1 hash value for the secondary image, enter the following command.

```
Brocade#verify sha secondary
Brocade#.....Done
Size = 2044830, SHA1 49d12d26552072337f7f5fcaef4cf4b742a9f525
```

To generate a CRC32 hash value for the secondary image, enter the following command.

```
Brocade#verify crc32 secondary
Brocade#.....Done
Size = 2044830, CRC32 b31fcbcb0
```

To verify the hash value of a secondary image with a known value, enter the following commands.

Image file types

```
Brocade#verify md5 secondary 01c410d6d153189a4a5d36c955653861
Brocade#.....Done
Size = 2044830, MD5 01c410d6d153189a4a5d36c955653862
Verification FAILED.
```

In the previous example, the codes did not match, and verification failed. If verification succeeds, the output will look like this.

```
Brocade#verify md5 secondary 01c410d6d153189a4a5d36c955653861
Brocade#.....Done
Size = 2044830, MD5 01c410d6d153189a4a5d36c955653861
Verification SUCCEEDED.
```

The following examples show this process for SHA-1 and CRC32 algorithms.

```
Brocade#verify sha secondary 49d12d26552072337f7f5fcaef4cf4b742a9f525
Brocade#.....Done
Size = 2044830, sha 49d12d26552072337f7f5fcaef4cf4b742a9f525
Verification SUCCEEDED.
```

and

```
Brocade#verify crc32 secondary b31fcbbc0
Brocade#.....Done
Size = 2044830, CRC32 b31fcbbc0
Verification SUCCEEDED.
```

Image file types

This section lists the boot and flash image file types supported and how to install them on the FastIron family of switches. For information about a specific version of code, refer to the release notes.

TABLE 17 Software image files

Product	Boot image ¹	Flash image
FESX FSX 800 FSX 1600	sxzxxxxx.bin	SXSxxxxx.bin (Layer 2) or SXLxxxxx.bin (base Layer 3) or SXRxxxxx.bin (full Layer 3)
FWS	fgzxxxxx.bin	FGSxxxxx.bin (Layer 2) or FGLxxxxx.bin (base Layer 3) or FGRxxxxx.bin (edge Layer 3)
FCX ICX 6610	grzxxxxx.bin	FCXSxxxxx.bin (Layer 2) or FCXRxxxxx.bin (Layer 3)
ICX 6430 ICX 6450	kxz07400.bin	ICX64S07400.bin (Layer 2) or ICX64R07400.bin (Layer 3 - ICX 6450 only)

1. These images are applicable to these devices only and are not interchangeable. For example, you cannot load FCX boot or flash images on a FSX device, and vice versa.

Software upgrades

Refer to the release notes for instructions about upgrading the software.

Boot code synchronization feature

The Brocade device supports automatic synchronization of the boot image in the active and redundant management modules. When the new boot image is copied into the active module, it is automatically synchronized with the redundant management module.

NOTE

There is currently no option for manual synchronization of the boot image.

To activate the boot synchronization process, enter the following command.

```
Brocade#copy tftp flash 10.20.65.194 /GA/SXZ07200.bin bootrom
```

The system responds with the following message.

```
Brocade#Load to buffer (8192 bytes per dot)
.....Write to boot flash.....
TFTP to Flash Done.
Brocade#Synchronizing with standby module...
Boot image synchronization done.
```

Viewing the contents of flash files

The **copy flash console** command can be used to display the contents of a configuration file, backup file, or renamed file stored in flash memory. The file contents are displayed on the console when the command is entered at the CLI.

To display a list of files stored in flash memory, do one of the following:

- For devices other than FCX and ICX, enter the **dir** command at the monitor mode. To enter monitor mode from any level of the CLI, press the **Shift** and **Control+Y** keys simultaneously then press the **M** key. Enter the **dir** command to display a list of the files stored in flash memory. To exit monitor mode and return to the CLI, press **Control+Z**.
- For FCX devices, enter the **show dir** command at any level of the CLI, or enter the **dir** command at the monitor mode.
- For ICX devices, enter the **show files** command at the device configuration prompt.

The following shows an example command output.

Viewing the contents of flash files

```
Brocade#show dir
133 [38f4] boot-parameter
    0 [ffff] bootrom
    3802772 [0000] primary
    4867691 [0000] secondary
    163 [dd8e] stacking.boot
    1773 [0d2d] startup-config
    1808 [acfa] startup-config.backup
    8674340 bytes 7 File(s)
56492032 bytes free
```

Syntax: show dir

To display the contents of a flash configuration file, enter a command such as the following from the User EXEC or Privileged EXEC mode of the CLI:

```
Brocade#copy flash console startup-config.backup
ver 07.0.00b1T7f1 !
stack unit 1
    module 1 fcx-24-port-management-module
    module 2 fcx-cx4-2-port-16g-module
    module 3 fcx-xfp-2-port-10g-module
    priority 80
    stack-port 1/2/1 1/2/2
stack unit 2
    module 1 fcx-48-poe-port-management-module
    module 2 fcx-cx4-2-port-16g-module
    module 3 fcx-xfp-2-port-10g-module
    stack-port 2/2/1 2/2/2
stack enable
!
!
!
!
vlan 1 name DEFAULT-VLAN by port
    no spanning-tree
    metro-rings 1
    metro-ring 1
        master
        ring-interfaces ethernet 1/1/2 ethernet 1/1/3
    enable
!
vlan 10 by port
    mac-vlan-permit ethe 1/1/5 to 1/1/6 ethe 2/1/5 to 2/1/6 no spanning-tree !
vlan 20 by port
    untagged ethe 1/1/7 to 1/1/8
    no spanning-tree
    pvlan type primary
    pvlan mapping 40 ethe 1/1/8
    pvlan mapping 30 ethe 1/1/7
!
vlan 30 by port
    untagged ethe 1/1/9 to 1/1/10
    no spanning-tree
    pvlan type community
!
...
some lines omitted for brevity...
```

Syntax: `copy flash console <filename>`

For `<filename>`, enter the name of a file stored in flash memory.

Using SNMP to upgrade software

You can use a third-party SNMP management application such as HP OpenView to upgrade software on a Brocade device.

NOTE

The syntax shown in this section assumes that you have installed HP OpenView in the “/usr” directory.

NOTE

Brocade recommends that you make a backup copy of the startup-config file before you upgrade the software. If you need to run an older release, you will need to use the backup copy of the startup-config file.

1. Configure a read-write community string on the Brocade device, if one is not already configured. To configure a read-write community string, enter the following command from the global CONFIG level of the CLI.

snmp-server community <string> ro | rw

where <string> is the community string and can be up to 32 characters long.

2. On the Brocade device, enter the following command from the global CONFIG level of the CLI.

no snmp-server pw-check

This command disables password checking for SNMP set requests. If a third-party SNMP management application does not add a password to the password field when it sends SNMP set requests to a Brocade device, by default the Brocade device rejects the request.

3. From the command prompt in the UNIX shell, enter the following command.

**/usr/OV/bin/snmpset -c <rw-community-string> <brcd-ip-addr> 1.3.6.1.4.1.1991.1.1.2.1.5.0
ipaddress <tftp-ip-addr> 1.3.6.1.4.1.1991.1.1.2.1.6.0 octetstringascii <file-name>
1.3.6.1.4.1.1991.1.1.2.1.7.0 integer <command-integer>**

where

<rw-community-string> is a read-write community string configured on the Brocade device.

<brcd-ip-addr> is the IP address of the Brocade device.

<tftp-ip-addr> is the TFTP server IP address.

<file-name> is the image file name.

<command-integer> is one of the following.

20 – Download the flash code into the primary flash area.

22 – Download the flash code into the secondary flash area.

Software reboot

You can use boot commands to immediately initiate software boots from a software image stored in primary or secondary flash on a Brocade device or from a BootP or TFTP server. You can test new versions of code on a Brocade device or choose the preferred boot source from the console boot prompt without requiring a system reset.

NOTE

It is very important that you verify a successful TFTP transfer of the boot code **before** you reset the system. If the boot code is not transferred successfully but you try to reset the system, the system will not have the boot code with which to successfully boot.

By default, the Brocade device first attempts to boot from the image stored in its primary flash, then its secondary flash, and then from a TFTP server. You can modify this booting sequence at the global CONFIG level of the CLI using the **boot system...** command.

To initiate an immediate boot from the CLI, enter one of the **boot system...** commands.

NOTE

When using the **boot system tftp** command, the IP address of the device and the TFTP server should be in the same subnet.

Software boot configuration notes

- In FastIron X Series devices, the **boot system tftp** command is supported on ports e 1 through e 12 only.
- If you are booting the device from a TFTP server through a fiber connection, use the following command: **boot system tftp <ip-address> <filename> fiber-port**.
- The **boot system tftp** command is not supported in a stacking environment.

Displaying the boot preference

Use the **show boot-preference** command to display the boot sequence in the startup config and running config files. The boot sequence displayed is also identified as either user-configured or the default.

The following example shows the default boot sequence preference.

```
Brocade#show boot-preference
Boot system preference (Configured):
  Use Default
Boot system preference(Default):
  Boot system flash primary
  Boot system flash secondary
```

The following example shows a user-configured boot sequence preference.

```
Brocade#show boot-preference
Boot system preference(Configured):
  Boot system flash secondary
  Boot system tftp 10.1.1.1 FGS04000b1.bin
  Boot system flash primary
```

```
Boot system preference (Default)
  Boot system flash primary
  Boot system flash secondary
```

Syntax: show boot-preference

The results of the **show run** command for the configured example above appear as follows.

```
Brocade#show run
Current Configuration:
!
ver 04.0.00x1T7e1
!
module 1 fgs-48-port-copper-base-module
module 2 fgs-xfp-1-port-10g-module
module 3 fgs-xfp-1-port-10g-module
!
alias cp=copy tf 10.1.1.1 FGS04000bl.bin pri
!
!
boot sys fl sec
boot sys df 10.1.1.1 FGS04000bl.bin
boot sys fl pri
ip address 10.1.1.4 255.255.255.0
snmp-client 10.1.1.1
!
end
```

Loading and saving configuration files

For easy configuration management, all Brocade devices support both the download and upload of configuration files between the devices and a TFTP server on the network.

You can upload either the startup configuration file or the running configuration file to the TFTP server for backup and use in booting the system:

- **Startup configuration file** – This file contains the configuration information that is currently saved in flash. To display this file, enter the **show configuration** command at any CLI prompt.
- **Running configuration file** – This file contains the configuration active in the system RAM but not yet saved to flash. These changes could represent a short-term requirement or general configuration change. To display this file, enter the **show running-config** or **write terminal** command at any CLI prompt.

Each device can have one startup configuration file and one running configuration file. The startup configuration file is shared by both flash modules. The running configuration file resides in DRAM.

When you load the startup-config file, the CLI parses the file three times.

1. During the first pass, the parser searches for **system-max** commands. A **system-max** command changes the size of statically configured memory.
2. During the second pass, the parser implements the **system-max** commands if present and also implements trunk configuration commands (**trunk** command) if present.
3. During the third pass, the parser implements the remaining commands.

Replacing the startup configuration with the running configuration

After you make configuration changes to the active system, you can save those changes by writing them to flash memory. When you write configuration changes to flash memory, you replace the startup configuration with the running configuration.

To replace the startup configuration with the running configuration, enter the following command at any Enable or CONFIG command prompt.

```
Brocade#write memory
```

Replacing the running configuration with the startup configuration

If you want to back out of the changes you have made to the running configuration and return to the startup configuration, enter the following command at the Privileged EXEC level of the CLI.

```
Brocade#reload
```

Logging changes to the startup-config file

You can configure a Brocade device to generate a Syslog message when the startup-config file is changed. The trap is enabled by default.

The following Syslog message is generated when the startup-config file is changed.

```
startup-config was changed
```

If the startup-config file was modified by a valid user, the following Syslog message is generated.

```
startup-config was changed by <username>
```

To disable or re-enable Syslog messages when the startup-config file is changed, use the following command.

Syntax: [no] logging enable config-changed

Copying a configuration file to or from a TFTP server

To copy the startup-config or running-config file to or from a TFTP server, use one of the following methods.

NOTE

For details about the **copy** and **ncopy** commands used with IPv6, refer to [“Using the IPv6 copy command”](#) on page 87 and [“IPv6 ncopy command”](#) on page 89.

NOTE

You can name the configuration file when you copy it to a TFTP server. However, when you copy a configuration file from the server to a Brocade device, the file is always copied as “startup-config” or “running-config”, depending on which type of file you saved to the server.

To initiate transfers of configuration files to or from a TFTP server using the CLI, enter one of the following commands:

- **copy startup-config tftp** <tftp-ip-addr> <filename> – Use this command to upload a copy of the startup configuration file from the Layer 2 Switch or Layer 3 Switch to a TFTP server.
- **copy running-config tftp** <tftp-ip-addr> <filename> – Use this command to upload a copy of the running configuration file from the Layer 2 Switch or Layer 3 Switch to a TFTP server.
- **copy tftp startup-config** <tftp-ip-addr> <filename> – Use this command to download a copy of the startup configuration file from a TFTP server to a Layer 2 Switch or Layer 3 Switch.

Dynamic configuration loading

You can load dynamic configuration commands (commands that do not require a reload to take effect) from a file on a TFTP server into the running-config on the Brocade device. You can make configuration changes off-line, then load the changes directly into the device running-config, without reloading the software.

Dynamic configuration usage considerations

- Use this feature only to load configuration information that does not require a software reload to take effect. For example, you cannot use this feature to change statically configured memory (**system-max** command) or to enter trunk group configuration information into the running-config.
- Do not use this feature if you have deleted a trunk group but have not yet placed the changes into effect by saving the configuration and then reloading. When you delete a trunk group, the command to configure the trunk group is removed from the device running-config, but the trunk group remains active. To finish deleting a trunk group, save the configuration (to the startup-config file), then reload the software. After you reload the software, then you can load the configuration from the file.
- Do not load port configuration information for secondary ports in a trunk group. Since all ports in a trunk group use the port configuration settings of the primary port in the group, the software cannot implement the changes to the secondary port.

Preparing the configuration file

A configuration file that you create must follow the same syntax rules as the startup-config file the device creates.

- The configuration file is a script containing CLI configuration commands. The CLI reacts to each command entered from the file in the same way the CLI reacts to the command if you enter it. For example, if the command results in an error message or a change to the CLI configuration level, the software responds by displaying the message or changing the CLI level.
- The software retains the running-config that is currently on the device, and changes the running-config only by adding new commands from the configuration file. If the running config already contains a command that is also in the configuration file you are loading, the CLI rejects the new command as a duplicate and displays an error message. For example, if the running-config already contains a command that configures ACL 1, the software rejects ACL 1 in the configuration file, and displays a message that ACL 1 is already configured.
- The file can contain global CONFIG commands or configuration commands for interfaces, routing protocols, and so on. You cannot enter User EXEC or Privileged EXEC commands.

- The default CLI configuration level in a configuration file is the global CONFIG level. Thus, the first command in the file must be a global CONFIG command or “!”. The ! (exclamation point) character means “return to the global CONFIG level”.

NOTE

You can enter text following “!” as a comment. However, the “!” is not a comment marker. It returns the CLI to the global configuration level.

NOTE

If you copy-and-paste a configuration into a management session, the CLI ignores the “!” instead of changing the CLI to the global CONFIG level. As a result, you might get different results if you copy-and-paste a configuration instead of loading the configuration using TFTP.

- Make sure you enter each command at the correct CLI level. Since some commands have identical forms at both the global CONFIG level and individual configuration levels, if the CLI response to the configuration file results in the CLI entering a configuration level you did not intend, then you can get unexpected results.

For example, if a trunk group is active on the device, and the configuration file contains a command to disable STP on one of the secondary ports in the trunk group, the CLI rejects the commands to enter the interface configuration level for the port and moves on to the next command in the file you are loading. If the next command is a spanning-tree command whose syntax is valid at the global CONFIG level as well as the interface configuration level, then the software applies the command globally. Here is an example.

The configuration file contains these commands.

```
interface ethernet 2
no spanning-tree
```

The CLI responds like this.

```
Brocade(config)#interface ethernet 2
Error - cannot configure secondary ports of a trunk
Brocade(config)#no spanning-tree
Brocade(config)#
```

- If the file contains commands that must be entered in a specific order, the commands must appear in the file in the required order. For example, if you want to use the file to replace an IP address on an interface, you must first remove the old address using “no” in front of the **ip address** command, then add the new address. Otherwise, the CLI displays an error message and does not implement the command. Here is an example.

The configuration file contains these commands.

```
interface ethernet 11
ip address 10.10.10.69/24
```

The running-config already has a command to add an address to port 11, so the CLI responds like this.

```
Brocade(config)#interface ethernet 11
Brocade(config-if-e1000-11)#ip add 10.10.10.69/24
Error: can only assign one primary ip address per subnet
Brocade(config-if-e1000-11)#
```

To successfully replace the address, enter commands into the file as follows.

```
interface ethernet 11
no ip address 20.20.20.69/24
ip address 10.10.10.69/24
```

This time, the CLI accepts the command, and no error message is displayed.

```
Brocade(config)#interface ethernet 11
Brocade(config-if-e1000-11)#no ip add 20.20.20.69/24
Brocade(config-if-e1000-11)#ip add 10.10.10.69/24
Brocade(config-if-e1000-11)
```

- Always use the **end** command at the end of the file. The **end** command must appear on the last line of the file, by itself.

Loading the configuration information into the running-config

To load the file from a TFTP server, use either of the following commands:

- **copy tftp running-config** <ip-addr> <filename>
- **ncopy tftp** <ip-addr> <filename> **running-config**

NOTE

If you are loading a configuration file that uses a truncated form of the CLI command **access-list**, the software will not go into batch mode.

For example, the following command line *will initiate* batch mode.

```
access-list 131 permit host pc1 host pc2
```

The following command line *will not* initiate batch mode.

```
acc 131 permit host pc1 host pc2
```

Maximum file sizes for startup-config file and running-config

Each Brocade device has a maximum allowable size for the running-config and the startup-config file. If you use TFTP to load additional information into a device running-config or startup-config file, it is possible to exceed the maximum allowable size. If this occurs, you will not be able to save the configuration changes.

The maximum size for the running-config and the startup-config file is 512K each.

To determine the size of a running-config or startup-config file, copy it to a TFTP server, then use the directory services on the server to list the size of the copied file. To copy the running-config or startup-config file to a TFTP server, use one of the following commands:

- Commands to copy the running-config to a TFTP server:
 - **copy running-config tftp** <ip-addr> <filename>
 - **ncopy running-config tftp** <ip-addr> <from-name>
- Commands to copy the startup-config file to a TFTP server:
 - **copy startup-config tftp** <ip-addr> <filename>
 - **ncopy startup-config tftp** <ip-addr> <from-name>

Loading and saving configuration files with IPv6

This section describes the IPv6 **copy** and **ncopy** commands.

Using the IPv6 copy command

The **copy** command for IPv6 allows you to do the following:

- Copy a file from a specified source to an IPv6 TFTP server
- Copy a file from an IPv6 TFTP server to a specified destination

Copying a file to an IPv6 TFTP server

You can copy a file from the following sources to an IPv6 TFTP server:

- Flash memory
- Running configuration
- Startup configuration

Copying a file from flash memory

For example, to copy the primary or secondary boot image from the device flash memory to an IPv6 TFTP server, enter a command such as the following.

```
Brocade#copy flash tftp 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies the secondary boot image named test.img from flash memory to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3.

Syntax: **copy flash tftp** <ipv6-address> <source-file-name> **primary** | **secondary**

The <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy to the IPv6 TFTP server.

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

Copying a file from the running or startup configuration

For example, to copy the running configuration to an IPv6 TFTP server, enter a command such as the following.

```
Brocade#copy running-config tftp 2001:7382:e0ff:7837::3 newrun.cfg
```

This command copies the running configuration to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 and names the file on the TFTP server newrun.cfg.

Syntax: **copy running-config** | **startup-config tftp** <ipv6-address> <destination-file-name>

Specify the **running-config** keyword to copy the running configuration file to the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration file to the specified IPv6 TFTP server.

The `tftp <ipv6-address>` parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<destination-file-name>` parameter specifies the name of the file that is copied to the IPv6 TFTP server.

Copying a file from an IPv6 TFTP server

You can copy a file from an IPv6 TFTP server to the following destinations:

- Flash memory
- Running configuration
- Startup configuration

Copying a file to flash memory

For example, to copy a boot image from an IPv6 TFTP server to the primary or secondary storage location in the device flash memory, enter a command such as the following.

```
Brocade#copy tftp flash 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies a boot image named `test.img` from an IPv6 TFTP server with the IPv6 address of `2001:7382:e0ff:7837::3` to the secondary storage location in the device flash memory.

Syntax: `copy tftp flash <ipv6-address> <source-file-name> primary | secondary`

The `<ipv6-address>` parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<source-file-name>` parameter specifies the name of the file you want to copy from the IPv6 TFTP server.

The **primary** keyword specifies the primary storage location in the device flash memory, while the **secondary** keyword specifies the secondary storage location in the device flash memory.

Copying a file to the running or startup configuration

For example, to copy a configuration file from an IPv6 TFTP server to the running or startup configuration, enter a command such as the following.

```
Brocade#copy tftp running-config 2001:7382:e0ff:7837::3 newrun.cfg overwrite
```

This command copies the `newrun.cfg` file from the IPv6 TFTP server and overwrites the running configuration file with the contents of `newrun.cfg`.

NOTE

To activate this configuration, you must reload (reset) the device.

Syntax: `copy tftp running-config | startup-config <ipv6-address> <source-file-name> [overwrite]`

Specify the **running-config** keyword to copy the running configuration from the specified IPv6 TFTP server.

The `<ipv6-address>` parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<source-file-name>` parameter specifies the name of the file that is copied from the IPv6 TFTP server.

The **overwrite** keyword specifies that the device should overwrite the current configuration file with the copied file. If you do not specify this parameter, the device copies the file into the current running or startup configuration but does not overwrite the current configuration.

IPv6 ncopy command

The **ncopy** command for IPv6 allows you to do the following:

- Copy a primary or secondary boot image from flash memory to an IPv6 TFTP server.
- Copy the running configuration to an IPv6 TFTP server.
- Copy the startup configuration to an IPv6 TFTP server
- Upload various files from an IPv6 TFTP server.

Copying a primary or secondary boot Image from flash memory to an IPv6 TFTP server

For example, to copy the primary or secondary boot image from the device flash memory to an IPv6 TFTP server, enter a command such as the following.

```
Brocade#ncopy flash primary tftp 2001:7382:e0ff:7837::3 primary.img
```

This command copies the primary boot image named `primary.img` from flash memory to a TFTP server with the IPv6 address of `2001:7382:e0ff:7837::3`.

Syntax: `ncopy flash primary | secondary tftp <ipv6-address> <source-file-name>`

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

The **tftp** `<ipv6-address>` parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<source-file-name>` parameter specifies the name of the file you want to copy from flash memory.

Copying the running or startup configuration to an IPv6 TFTP server

For example, to copy a device running or startup configuration to an IPv6 TFTP server, enter a command such as the following.

```
Brocade#ncopy running-config tftp 2001:7382:e0ff:7837::3 bakrun.cfg
```

This command copies a device running configuration to a TFTP server with the IPv6 address of `2001:7382:e0ff:7837::3` and names the destination file `bakrun.cfg`.

Syntax: `ncopy running-config | startup-config tftp <ipv6-address> <destination-file-name>`

Specify the **running-config** keyword to copy the device running configuration or the **startup-config** keyword to copy the device startup configuration.

The **tftp** *<ipv6-address>* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *<destination-file-name>* parameter specifies the name of the running configuration that is copied to the IPv6 TFTP server.

IPv6 TFTP server file upload

You can upload the following files from an IPv6 TFTP server:

- Primary boot image.
- Secondary boot image.
- Running configuration.
- Startup configuration.

Uploading a primary or secondary boot image from an IPv6 TFTP server

For example, to upload a primary or secondary boot image from an IPv6 TFTP server to a device flash memory, enter a command such as the following.

```
Brocade#ncopy tftp 2001:7382:e0ff:7837::3 primary.img flash primary
```

This command uploads the primary boot image named primary.img from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device primary storage location in flash memory.

Syntax: **ncopy tftp** *<ipv6-address>* *<source-file-name>* **flash primary | secondary**

The **tftp** *<ipv6-address>* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *<source-file-name>* parameter specifies the name of the file you want to copy from the TFTP server.

The **primary** keyword specifies the primary location in flash memory, while the **secondary** keyword specifies the secondary location in flash memory.

Uploading a running or startup configuration from an IPv6 TFTP server

For example to upload a running or startup configuration from an IPv6 TFTP server to a device, enter a command such as the following.

```
Brocade#ncopy tftp 2001:7382:e0ff:7837::3 newrun.cfg running-config
```

This command uploads a file named newrun.cfg from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device.

Syntax: **ncopy tftp** *<ipv6-address>* *<source-file-name>* **running-config | startup-config**

The **tftp** *<ipv6-address>* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *<source-file-name>* parameter specifies the name of the file you want to copy from the TFTP server.

Specify the **running-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current running configuration but does not overwrite the current configuration.

Specify the **startup-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current startup configuration but does not overwrite the current configuration.

Using SNMP to save and load configuration information

You can use a third-party SNMP management application such as HP OpenView to save and load a configuration on a Brocade device. To save and load configuration information using HP OpenView, use the following procedure.

NOTE

The syntax shown in this section assumes that you have installed HP OpenView in the “/usr” directory.

1. Configure a read-write community string on the Brocade device, if one is not already configured. To configure a read-write community string, enter the following command from the global CONFIG level of the CLI.

snmp-server community <string> ro | rw

where <string> is the community string and can be up to 32 characters long.

2. On the Brocade device, enter the following command from the global CONFIG level of the CLI.

no snmp-server pw-check

This command disables password checking for SNMP set requests. If a third-party SNMP management application does not add a password to the password field when it sends SNMP set requests to a Brocade device, by default the Brocade device rejects the request.

3. From the command prompt in the UNIX shell, enter the following command.

```
/usr/OV/bin/snmpset -c <rw-community-string> <fdry-ip-addr> 1.3.6.1.4.1.1991.1.1.2.1.5.0
ipaddress <tftp-ip-addr> 1.3.6.1.4.1.1991.1.1.2.1.8.0 octetstringascii <config-file-name>
1.3.6.1.4.1.1991.1.1.2.1.9.0 integer <command-integer>
```

where

<rw-community-string> is a read-write community string configured on the Brocade device.

<fdry-ip-addr> is the IP address of the Brocade device.

<tftp-ip-addr> is the TFTP server IP address.

<config-file-name> is the configuration file name.

<command-integer> is one of the following:

20 – Upload the startup-config file from the flash memory of the Brocade device to the TFTP server.

21 – Download a startup-config file from a TFTP server to the flash memory of the Brocade device.

22 – Upload the running-config from the flash memory of the Brocade device to the TFTP server.

23 – Download a configuration file from a TFTP server into the running-config of the Brocade device.

NOTE

Option **23** adds configuration information to the running-config on the device, and does not replace commands. If you want to replace configuration information in the device, use “no” forms of the configuration commands to remove the configuration information, then use configuration commands to create the configuration information you want. Follow the guidelines in [“Dynamic configuration loading”](#) on page 84.

Erasing image and configuration files

To erase software images or configuration files, use the commands described below. These commands are valid at the Privileged EXEC level of the CLI:

- **erase flash primary** erases the image stored in primary flash of the system.
- **erase flash secondary** erases the image stored in secondary flash of the system.
- **erase startup-config** erases the configuration stored in the startup configuration file; however, the running configuration remains intact until system reboot.

System reload scheduling

In addition to reloading the system manually, you can configure the Brocade device to reload itself at a specific time or after a specific amount of time has passed.

NOTE

The scheduled reload feature requires the system clock. You can use a Simple Network Time Protocol (SNTP) server to set the clock or you can set the device clock manually. Refer to [“Specifying an SNTP server”](#) on page 24 or [“Setting the system clock”](#) on page 30.

Reloading at a specific time

To schedule a system reload for a specific time, use the **reload at** command. For example, to schedule a system reload from the primary flash module for 6:00:00 AM, April 1, 2003, enter the following command at the global CONFIG level of the CLI.

```
Brocade#reload at 06:00:00 04-01-03
```

Syntax: **reload at** <hh:mm:ss> <mm-dd-yy> [**primary** | **secondary**]

<hh:mm:ss> is the hours, minutes, and seconds.

<mm-dd-yy> is the month, day, and year.

primary | **secondary** specifies whether the reload is to occur from the primary code flash module or the secondary code flash module. The default is **primary**.

Reloading after a specific amount of time

To schedule a system reload to occur after a specific amount of time has passed on the system clock, use **reload after** command. For example, to schedule a system reload from the secondary flash one day and 12 hours later, enter the following command at the global CONFIG level of the CLI.

```
Brocade#reload after 01:12:00 secondary
```

Syntax: **reload after** <dd:hh:mm> [primary | secondary]

<dd:hh:mm> is the number of days, hours, and minutes.

primary | secondary specifies whether the reload is to occur from the primary code flash module or the secondary code flash module.

Displaying the amount of time remaining before a scheduled reload

To display how much time is remaining before a scheduled system reload, enter the following command from any level of the CLI.

```
Brocade#show reload
```

Canceling a scheduled reload

To cancel a scheduled system reload using the CLI, enter the following command at the global CONFIG level of the CLI.

```
Brocade#reload cancel
```

Diagnostic error codes and remedies for TFTP transfers

This section describes the error messages associated with TFTP transfer of configuration files, software images or flash images to or from a Brocade device.

Error code	Message	Explanation and action
1	Flash read preparation failed.	A flash error occurred during the download. Retry the download. If it fails again, contact customer support.
2	Flash read failed.	
3	Flash write preparation failed.	
4	Flash write failed.	
5	TFTP session timeout.	TFTP failed because of a time out. Check IP connectivity and make sure the TFTP server is running.
6	TFTP out of buffer space.	The file is larger than the amount of room on the device or TFTP server. If you are copying an image file to flash, first copy the other image to your TFTP server, then delete it from flash. (Use the erase flash... CLI command at the Privileged EXEC level to erase the image in the flash.) If you are copying a configuration file to flash, edit the file to remove unnecessary information, then try again.

Diagnostic error codes and remedies for TFTP transfers

Error code	Message	Explanation and action
7	TFTP busy, only one TFTP session can be active.	Another TFTP transfer is active on another CLI session, or Web management session, or network management system. Wait, then retry the transfer.
8	File type check failed.	You accidentally attempted to copy the incorrect image code into the system. For example, you might have tried to copy a Chassis image into a Compact device. Retry the transfer using the correct image.
16	TFTP remote - general error.	The TFTP configuration has an error. The specific error message describes the error. Correct the error, then retry the transfer.
17	TFTP remote - no such file.	
18	TFTP remote - access violation.	
19	TFTP remote - disk full.	
20	TFTP remote - illegal operation.	
21	TFTP remote - unknown transfer ID.	
22	TFTP remote - file already exists.	
23	TFTP remote - no such user.	

This section describes the error messages associated with the TFTP transfer of PoE firmware file to a Brocade device.

Message	Explanation and action
Firmware TFTP timeout.	TFTP failed because of a time out. Check IP connectivity and make sure the TFTP server is running.
Firmware is not valid for this platform.	TFTP failed because the specified file is not found on the TFTP server.
Firmware is not valid for the IEEE 802.3at (PoE-Plus) controller type.	Each PoE firmware file delivered by Brocade is meant to be used on the specific platform only. If the file is used on a platform for which it is not meant, then this error message will display. Download the correct file, then retry the transfer.
Firmware is not valid for the IEEE 802.3af PoE controller type.	
Firmware type cannot be detected from the firmware content.	Each PoE firmware file delivered by Brocade is meant to be used on the specific platform and the specific PoE controller on the specified module. If the file is used for a platform for which it is meant, but the PoE controller is not same then this error message will display. Download the correct file, then retry the transfer.
TFTP File not Valid for PoE Controller Type.	
Firmware tftp remote file access failed.	The TFTP server needs read access on the PoE firmware file. Check the permissions on the file, then try again.

Network connectivity testing

After you install the network cables, you can test network connectivity to other devices by pinging those devices. You also can observe the LEDs related to network connection and perform trace routes.

For more information about observing LEDs, refer to the *Brocade FastIron X Series Chassis Hardware Installation Guide* and the *Brocade FastIron Compact Switch Hardware Installation Guide*.

Pinging an IPv4 address

NOTE

This section describes the *IPv4 ping* command. For details about *IPv6 ping*, refer to [“Pinging an IPv6 address”](#) on page 370.

To verify that a Brocade device can reach another device through the network, enter a command such as the following at any level of the CLI on the Brocade device:

```
Brocade> ping 192.33.4.7
```

Syntax: `ping <ip addr> | <hostname> [source <ip addr>] [count <num>] [timeout <msec>] [ttl <num>] [size <byte>] [quiet] [numeric] [no-fragment] [verify] [data <1-to-4 byte hex>] [brief [max-print-per-sec <number>]]`

NOTE

If the device is a Brocade Layer 2 Switch or Layer 3 Switch, you can use the host name only if you have already enabled the Domain Name Server (DNS) resolver feature on the device from which you are sending the ping. Refer to [“IP Configuration”](#) on page 945.

The required parameter is the IP address or host name of the device.

The **source** `<ip addr>` specifies an IP address to be used as the origin of the ping packets.

The **count** `<num>` parameter specifies how many ping packets the device sends. You can specify from 1 – 4294967296. The default is 1.

The **timeout** `<msec>` parameter specifies how many milliseconds the Brocade device waits for a reply from the pinged device. You can specify a timeout from 1 – 4294967296 milliseconds. The default is 5000 (5 seconds).

The **ttl** `<num>` parameter specifies the maximum number of hops. You can specify a TTL from 1 – 255. The default is 64.

The **size** `<byte>` parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 – 4000. The default is 16.

The **no-fragment** parameter turns on the “don’t fragment” bit in the IP header of the ping packet. This option is disabled by default.

The **quiet** parameter hides informational messages such as a summary of the ping parameters sent to the device and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.

The **verify** parameter verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.

The **data** *<1 – 4 byte hex>* parameter lets you specify a specific data pattern for the payload instead of the default data pattern, “abcd”, in the packet data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE

For numeric parameter values, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

The **brief** parameter causes ping test characters to be displayed. The following ping test characters are supported:

- ! Indicates that a reply was received.
- . Indicates that the network server timed out while waiting for a reply.
- U Indicates that a destination unreachable error PDU was received.
- I Indicates that the user interrupted ping.

NOTE

The number of ! characters displayed may not correspond to the number of successful replies by the **ping** command. Similarly, the number of . characters displayed may not correspond to the number of server timeouts that occurred while waiting for a reply. The "success" or "timeout" results are shown in the display as “Success rate is XX percent (X/Y)”.

The optional **max-print-per-sec** *<number>* parameter specifies the maximum number of target responses the Brocade device can display per second while in brief mode. You can specify from 0 – 2047. The default is 511.

NOTE

If you address the ping to the IP broadcast address, the device lists the first four responses to the ping.

NOTE

On 48GC modules in non-jumbo mode, the maximum size of ping packets is 1486 bytes and the maximum frame size of tagged traffic is no larger than 1581 bytes.

Tracing an IPv4 route

NOTE

This section describes the **IPv4 traceroute** command. For details about **IPv6 traceroute**, refer to [“IPv6 traceroute”](#) on page 368.

Use the **traceroute** command to determine the path through which a Brocade device can reach another device. Enter the command at any level of the CLI.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a given TTL. In addition, if there are multiple equal-cost routes to the destination, the Brocade device displays up to three responses by default.

```
Brocade> traceroute 192.33.4.7
```

Syntax: **traceroute** *<host-ip-addr>* [**maxttl** *<value>*] [**minttl** *<value>*] [**numeric**] [**timeout** *<value>*] [**source-ip** *<ip-addr>*]

Possible and default values are as follows.

minttl – minimum TTL (hops) value: Possible values are 1 – 255. Default value is 1 second.

maxttl – maximum TTL (hops) value: Possible values are 1 – 255. Default value is 30 seconds.

timeout – Possible values are 1 – 120. Default value is 2 seconds.

numeric – Lets you change the display to list the devices by their IP addresses instead of their names.

source-ip <ip-addr> – Specifies an IP address to be used as the origin for the traceroute.

Hitless management on the FSX 800 and FSX 1600

Hitless management is supported on the FSX 800 and FSX 1600 chassis with dual management modules. It is a high-availability feature set that ensures no loss of data traffic during the following events:

- Management module failure or role change
- Software failure
- Addition or removal of modules
- Operating system upgrade

During such events, the standby management module takes over the active role and the system continues to forward traffic seamlessly, as if no failure or topology change has occurred. In software releases that do not support hitless management, events such as these could cause a system reboot, resulting in an impact to data traffic.

The following Hitless management features are supported:

Hitless Switchover – A manually controlled (CLI-driven) switchover of the active and standby management modules without any packet loss to the services and protocols that are supported by Hitless management. A switchover is activated by the CLI command **switch-over-active-role**.

Hitless Failover – An automatic, forced switchover of the active and standby management modules because of a failure or abnormal termination of the active management module. In the event of a failover, the active management module abruptly leaves and the standby management module immediately assumes the active role. Like a switchover, a failover occurs without any packet loss to hitless-supported services and protocols. Unlike a switchover, a failover generally happens without warning.

Hitless Operating System (OS) Upgrade – An operating system upgrade and controlled switchover without any packet loss to the services and protocols that are supported by Hitless management.

The services and protocols supported by Hitless management are listed in [Table 18](#) on page 99.

Hitless failover and hitless switchover are disabled by default. To enable these features, refer to [“Enabling hitless failover on the FSX 800 and FSX 1600”](#) on page 103 and [“Executing a hitless switchover on the FSX 800 and FSX 1600”](#) on page 104.

Benefits of hitless management

The benefits of Hitless management include the following:

- The standby management module (the module that takes over the active role) and all interface modules in the chassis are not reset
- Existing data traffic flows continue uninterrupted with no traffic loss
- Port link states remain UP for the duration of the hitless management event
- System configurations applied through Console/SNMP/HTTP interfaces remain intact
- Hitless switchover can be used by a system administrator, for example, to perform maintenance on a management module that has been functioning as the active management module. Some advantages of a hitless switchover over a hitless software reload are:
 - A manual switchover is quicker, since the standby module does not have to reboot.
 - Switched traffic through the Ethernet interfaces on the standby management module is not interrupted.

NOTE

All traffic going through Ethernet interfaces (if present) on the management modules will be interrupted during a hitless OS upgrade. This is because both management modules must be reloaded with the new image. This applies to hitless OS upgrade only. It does not apply to hitless switchover or failover, which does not interrupt traffic going through Ethernet interfaces on the standby management module (the module that takes over the active role).

Supported protocols and services for hitless management events

[Table 18](#) lists the services and protocols that are supported by Hitless management. [Table 18](#) also highlights the impact of *Hitless management events* (switchover, failover, and OS upgrade) to the system's major functions.

NOTE

Services and protocols that are not listed in [Table 18](#) may be disrupted, but will resume normal operation once the new active management module is back up and running.

TABLE 18 Hitless-supported services and protocols – FSX 800 and FSX 1600

Traffic type	Supported protocols and services	Impact
Layer 2 switched traffic, including unicast and multicast + System-level + Layer 4	<ul style="list-style-type: none"> • 802.1p and 802.1Q • 802.3ad – LACP • 802.3af – PoE • 802.3at – PoE+ • DSCP honoring and Diffserv • Dual-mode VLAN • IGMP v1, v2, and v3 snooping • IPv4 ACLs • IPv6 ACLs • Layer 2 switching (VLAN and 802.1Q-in-Q) • MLD v1 and v2 snooping • MRP • Multiple spanning tree (MSTP) • Physical port/link state • PIM SM snooping • Port mirroring and monitoring • Port trunking • Rapid spanning tree (RSTP) • Spanning tree (STP) • ToS-based QoS • Policy Based Routing • Traffic policies • UDLD • VSRP 	<p>Layer 2 switched traffic is not impacted during a Hitless management event. All existing switched traffic flows continue uninterrupted.</p> <p>New switched flows are not learned by the FastIron switch during the switchover process and are flooded to the VLAN members in hardware. After the new active management module becomes operational, new switched flows are learned and forwarded accordingly. The Layer 2 control protocol states are not interrupted during the switchover process.</p> <p>Configured ACLs, PBR or GRE & IPv6 to IPv4 Tunnels will operate in a hitless manner.</p>
Layer 3 IPv4 routed traffic	<ul style="list-style-type: none"> • BGP4 • IPv4 unicast forwarding • OSPF v2 • OSPF v2 with ECMP • Static routes • VRRP • VRRP-E • GRE • IPv6 to IPv4 Tunnels 	<p>Layer 3 routed traffic for supported protocols is not impacted during a Hitless management event.</p> <p>All existing Layer 3 IPv4 multicast flows and receivers will be interrupted. Traffic will converge to normalcy after the new active module becomes operational.</p> <p>Other Layer 3 protocols that are not supported will be interrupted during the switchover or failover.</p> <p>If BGP4 graceful restart or OSPF graceful restart is enabled, it will be gracefully restarted and traffic will converge to normalcy after the new active module becomes operational. For details about OSPF graceful restart, refer to “OSPF graceful restart” on page 1234. For details about BGP4 graceful restart, refer to “BGP4 graceful restart” on page 1347.</p> <p>Configured ACLs, PBR or GRE & IPv6 to IPv4 Tunnels will operate in a hitless manner.</p>
Management traffic	N/A	All existing management sessions (SNMP, TELNET, HTTP, HTTPS, FTP, TFTP, SSH etc.), are interrupted during the switchover or failover process. All such sessions are terminated and can be re-established after the new Active Controller takes over.

TABLE 18 Hitless-supported services and protocols – FSX 800 and FSX 1600

Traffic type	Supported protocols and services	Impact
Security	<ul style="list-style-type: none"> • 802.1X, including use with dynamic ACLs and VLANs • IPv4 ACLs • IPv6 ACLs • DHCP snooping • Dynamic ARP inspection • EAP with RADIUS • IP source guard • Multi-device port authentication, including use with dynamic ACLs and VLANs 	<p>Supported security protocols and services are not impacted during a switchover or failover.</p> <p>NOTE: If 802.1X and multi-device port authentication are enabled together on the same port, both will be impacted during a switchover or failover. Hitless support for these features applies to ports with 802.1X <i>only</i> or multi-device port authentication <i>only</i>.</p> <p>Configured ACLs will operate in a hitless manner, meaning the system will continue to permit and deny traffic during the switchover or failover process.</p>
Other services to Management	<ul style="list-style-type: none"> • AAA • DHCP • sFlow • SNMP v1, v2, and v3 • SNMP traps • SNTP • Traceroute 	<p>Supported protocols and services are not impacted during a switchover or failover.</p> <p>DNS lookups will continue after a switchover or failover. This information is not synchronized.</p> <p>Ping traffic will be minimally impacted.</p>

Hitless management configuration notes and feature limitations

The following limitations apply to hitless management support.

- All traffic going through Ethernet interfaces (if present) on the management modules will be interrupted during a hitless OS upgrade. This is because both management modules must be reloaded with the new image. This applies to hitless OS upgrade only. It does not apply to hitless switchover or failover, which does not interrupt traffic going through Ethernet interfaces on the standby management module (the module that takes over the active role).
- Static and dynamic multi-slot trunks will flap during a hitless switchover if any of the trunk port members reside on the management module.
- Layer 3 multicast traffic is not supported by Hitless management.

Hitless reload or switchover requirements and limitations

The section describes the design limitation on devices with the following configuration:

- 0-port management modules
- One or more first or second generation line cards
- One or more third generation line cards

For hitless reload or switch-over-active-role to succeed, the following requirements and limitations must be met:

- The standby management module must be up and in an "OK (Enabled)" state.
- A configuration requiring a reload must not be pending.

- A hitless-reload must not have already been issued on the previous active management module.
- POE firmware must not be in progress.
- The SXR running configuration must not be classified as too large (greater than 512KB).
- A TFTP session must not be in progress.
- An image sync session must not be in progress.
- The current active management card cannot have a memory utilization of greater than 90% of available memory.
- A line card hotswap must not be in progress.

If any of these conditions are not met, an appropriate error message is printed to the console and hitless-reload or switch-over will not succeed.

What happens during a Hitless switchover or failover

This section describes the internal events that enable a controlled or forced switchover (failover) to take place in a hitless manner, as well as the events that occur during the switchover.

Separate data and control planes

The FSX 800 and FSX 1600 management modules have separate data and control planes. The *data plane* forwards traffic between the switch fabric modules and all of the Interface modules in the chassis. The *control plane* carries traffic that is destined for the CPU of the active management module. Control plane traffic includes the following:

- Management traffic
- Control protocol traffic
- In some cases, the first packet of a data flow

During a controlled or forced switchover, the data plane is not affected. Traffic in the forwarding plane will continue to run without interruption while the standby management module takes over operation of the system. However, traffic in the control plane will be minimally impacted.

Real-time synchronization between management modules

Hitless management requires that the active and standby management modules are fully synchronized at any given point in time. This is accomplished by *baseline* and *dynamic synchronization* of the modules.

When a standby management module is inserted and becomes operational in the FSX 800 or FSX 1600 chassis, the standby module sends a baseline synchronization request to the active management module. The request prompts the active management module to copy the current state of its CPU to the standby CPU, including:

- Start-up and run-time configuration (CLI)
- Layer 2 protocols – Layer 2 protocols such as STP, RSTP, MRP, and VSRP run concurrently on both the active and standby management modules.
- Hardware Abstraction Layer (HAL) – This includes the prefix-based routing table, next hop information for outgoing interfaces, and tunnel information.

- Layer 3 IP forwarding information – This includes the routing table, IP cache table, and ARP table, as well as static and connected routes.
- Layer 3 routing protocols are not copied to the standby management module, but remain in init state on the standby module until a switchover occurs. Peer adjacency will be restored after a switchover. If BGP4 or OSPF graceful restart are enabled during a switchover, the standby management module (new active module) will initiate a graceful restart and a new set of routes will be relearned. The new set of routes will be the same as the old routes, except in the case of a network change.

As baseline synchronization is performed, the console of the active management module displays the progress of the synchronization.

```
ACTIVE: Detected Stdbby heart-beat
ACTIVE: Standby is ready for baseline synchronization.
ACTIVE: Baseline SYNC is completed. Protocol Sync is in progress.
ACTIVE: State synchronization is complete.
```

The first message indicates that the active management module has detected the standby management module. The second message indicates that the standby module has been hot-inserted and is ready for baseline synchronization. The third message is seen when baseline synchronization is completed, and the fourth message is seen when protocol synchronization is completed.

The console of the standby management module also displays the progress of the synchronization.

```
STBY: Baseline SYNC is completed. Protocol Sync is in progress.
STBY: State synchronization is complete.
```

The first message indicates that baseline synchronization is completed, and the second message indicates that protocol synchronization is completed.

When control protocols are synchronized and protocol synchronization timers expire, the standby management module will be in *hot-standby* mode, meaning the standby module is ready to take over as the active management module. In the event of a switchover, the standby module will pick up where the active module left off, without interrupting data traffic.

After baseline synchronization, any new events that occur on the active CPU will be dynamically synchronized on the standby CPU. Examples of such events include:

- CLI/HTTP/SNMP configurations
- CPU receive packets
- Link events
- Interrupts
- Layer 2 and Layer 3 forwarding table updates
- Dynamic user authentication updates such as 802.1X or multi-device port authentication

Dynamic events are synchronized in such a way that if the active CPU fails before fully executing an event, the standby CPU (newly active CPU) will execute the event after the failover. Also, if the active CPU aborts the event, the standby CPU will abort the event as well.

NOTE

Since both the standby and active management modules run the same code, a command that brings down the active management module will most likely bring down the standby management module. Because all configuration commands are synchronized from active to standby management module in real time, both management modules will reload at almost the same time. This in turn will cause the system to reset all interface modules (similar to the behavior when the **reboot** command is executed) and will cause packet loss associated with a system reboot.

NOTE

If the new active management module becomes out-of-sync with an interface module, information on the interface module can be overwritten in some cases, which can cause an interruption of traffic forwarding.

How a Hitless switchover or failover impacts system functions

For a description of the feature's impact to major system functions, refer to [Table 18](#) on page 99.

Enabling hitless failover on the FSX 800 and FSX 1600

Hitless failover is disabled by default. When disabled, the following limitations are in effect:

- If a failover occurs, the system will reload. The following message will display on the console prior to a reload.

```
STBY:- - - - Active Hitless Failover is disabled. Re-setting the system - -
```

- Manual switchover (CLI command **switch-over-active-role**) is not allowed. If this command is entered, the following message will display on the console:

```
Switch-over is not allowed. Reason: hitless-failover not configured.
```

NOTE

Hitless OS upgrade is *not* impacted by this option and is supported whether or not hitless failover is enabled.

NOTE

Synchronization between the active management module and standby management module will occur whether or not hitless failover is enabled.

To enable hitless failover, enter the following command at the Global CONFIG level of the CLI:

```
Brocade(config)#hitless-failover enable
```

The command takes effect immediately. Manual switchover is allowed, and in the event of a failover, the standby management module will take over the active role without reloading the system.

Syntax: **[no] hitless-failover enable**

Use the **no** form of the command to disable hitless failover once it has been enabled.

Executing a hitless switchover on the FSX 800 and FSX 1600

Hitless failover must be enabled before a hitless switchover can be executed. To enable hitless failover, refer to [“Enabling hitless failover on the FSX 800 and FSX 1600”](#) on page 103.

To switch over to the standby module (and thus make it the active module), enter the following command.

```
Brocade# switch-over-active-role
```

Once you enter this command, the system will prompt you as follows.

```
Are you sure? (enter 'y' or 'n'): y
Running Config data has been changed. Do you want to continue
the switch-over without saving the running config? (enter 'y' or 'n'): n
Please save the running config and try switch-over again
```

Syntax: switch-over-active role

If this command is entered when hitless failover is disabled, the following message will appear on the console:

```
Switch-over is not allowed. Reason: hitless-failover not configured.
```

Hitless OS upgrade on the FSX 800 and FSX 1600

Hitless Operating System (OS) Upgrade enables an operating system upgrade and switchover without any packet loss to the services and protocols that are supported by Hitless management.

What happens during a Hitless OS upgrade

The following steps describe the internal events that occur during a hitless OS upgrade.

1. The standby management module resets and reloads with the new software image in its flash memory.
2. The Ethernet interfaces (if present) on the standby module become operational and start carrying data traffic.
3. The active management module synchronizes the standby management module with all the information required to take over the active role.
4. The Layer 2 and Layer 3 control protocols on the standby management module converge. This process takes approximately 70 seconds.
5. The standby management module takes over the active role.
6. The old active management module resets and reloads with the same software image running on the newly active management module.
7. The FastIron switch is now operating with the new software image. The management module that was initially configured as the standby management module is now the active management module and the management module that was initially configured as the active management module is now the standby.

NOTE

The events described above occur internally and do not create or affect the external network topology.

Hitless OS upgrade considerations

Consider the following when using the hitless OS upgrade feature:

- Hitless OS upgrade allows for upgrading the software in a system between two releases of the OS that support this functionality and have compatible data structures. A hitless O/S downgrade may also be supported if the current and target code releases have compatible data structures. From time to time it may be necessary, when enhancing the software or adding new features, to change or add data structures that may cause some releases to be incompatible. In such cases, an upgrade or downgrade will not be hitless, and the software will use the regular Brocade upgrade process - relying on fast reboot.
- For a description of how this feature impacts major system functions, refer to [Table 18](#) on page 99.
- You must have both active and standby management modules installed to use this feature.
- Hitless OS upgrade is supported in software release FSX 05.0.00 or higher, with boot image FSX 05.0.00 or higher. In general, it is supported with patch upgrades, for example, when upgrading from release 07.0.01a to 07.0.01b. It is not supported during major release upgrades, for example when upgrading from release 07.0.00 to 07.1.00.
- This feature can be used to upgrade an image to a higher or lower compatible version of the software. However, if hitless upgrade to a particular software version is not supported, the software upgrade must be performed through a fast reload of the system.
- Hitless OS upgrade between different types of software images is not supported. For example, hitless OS upgrade is supported when upgrading the Layer 2 image to another Layer 2 image. It is not supported when upgrading the Layer 2 image to a Layer 3 image, or the base Layer 3 image to a full Layer 3 image, and so on.
- Hitless OS upgrade should be performed locally, since remote connectivity will be lost during the upgrade. During a reload, HTTP, SSH, Telnet, SNMP, and ping sessions will be dropped.
- The active management module switches from the initial active management module to the standby management module during the hitless upgrade process. Therefore, a connection to the console interface on both management modules is required.
- Upon being reset, any traffic going through the ports on the management module will be interrupted. Once the management module is up and running, it will be able to send and receive packets, even before the hitless upgrade process is complete.
- The running configuration is not allowed to be changed any time during the hitless upgrade process.
- System-max configuration changes require a system reload. System-max configuration changes do not take effect by the hitless upgrade. Even if a system-max parameter is changed and saved in the startup configuration, the FastIron switch will revert to the default system-max value upon a hitless software upgrade. The new system-max value will only take effect after a regular system reload.
- Other commands requiring a software reload, such as CAM mode changes, also do not take effect upon hitless upgrade and require a system reload before being placed in effect.

Hitless OS upgrade configuration steps

The following is a summary of the configuration steps for a hitless OS software upgrade.

1. Copy the software image that supports hitless software upgrade from a TFTP server to the FastIron switch. Refer to [“Loading the software onto the switch”](#).
2. Install the software image in flash memory on the active and standby management modules.
3. Enter the **hitless-reload** command on the active management module. The command triggers the events described in the section [“What happens during a Hitless OS upgrade”](#) on page 104.

Loading the software onto the switch

Hitless OS upgrade loads from the primary and secondary images on the FSX 800 and FSX 1600 Management modules. If you will be using the **hitless-reload** command to perform the hitless upgrade, you must first copy the software image that supports hitless software upgrade onto the flash memory of the active and standby management modules. For instructions, refer to the release notes.

Performing a hitless upgrade

After loading the software image onto the flash memory of the active and standby management modules, you can begin the process of performing a hitless OS upgrade using the **hitless-reload** command. For example,

```
Brocade#hitless-reload primary
```

Syntax: **hitless-reload primary | secondary**

The **primary** parameter specifies that the management module will be reloaded with the **primary** image.

The **secondary** parameter specifies that the management module will be reloaded with the **secondary** image.

NOTE

The **hitless-reload** command is accepted only when the running configuration and startup configuration files match. If the configuration file has changed, you must first save the file (**write mem**) before executing a hitless reload. Otherwise, the following message will display on the console.

```
Error: Running config and start-up config differs. Please reload the system or save the configuration before attempting hitless reload.
```

Syslog message for Hitless management events

The following Syslog message is generated as a result of a switchover or hitless OS upgrade.

```
SWITCHOVER COMPLETED - by admin - Mgmt Module in slot <slotnum> is now Active
```

The following Syslog message is generated as a result of a failover.

```
SWITCHOVER COMPLETED - by active CPU failure - Mgmt Module in slot <slotnum> is now Active
```

Displaying diagnostic information

Use the following commands to display diagnostic information for a hitless switchover or failover.

```
Brocade#show ipc
Version 6, Grp 0, Recv: stk-p0: 840918, pl: 0, sum: 840918
Message types have callbacks:
  1:Reliable IPC message 2:Reliable IPC atomic 4:fragmentation,jumbo
  20:SYNC dynamic change 22:SYNC download reply 24:SYNC download spec i
  25:SYNC restart download 26:SYNC verification 27:SYNC disable/enable
  29:SYNC mgmt hello 35:IPC Ready Msg 36:IPC Msg for Sync Fra
  38:SYNC reliable
Send message types:
  [1]=815798, [21]=1, [35]=1, [38]=24442,
Recv message types:
  [1]=816446,0, [20]=2,0 [22]=1,0
  [29]=25,0, [38]=24442,0,

Statistics:
  send pkt num : 840242, recv pkt num : 840918
  send msg num : 840242, recv msg num : 840918,
  send frag pkt num : 0, recv frag pkt num : 0,
  pkt buf alloc : 832113,

Reliable-mail    send  success  receive  time us
target ID        0         0         0         0
target MAC       0         0         0         0
There is 0 current jumbo IPC session

Possible errors:
  ***recv msg no callback 2, last msg_type=20, from stack0, e1/9
```

Syntax: show ipc

```
Brocade#show ipc_stat
Total available Hsync channel space = 1048580
Total available Appl channel space = 524292
Total number of application msgs in dyn queue = 0
Total number of hsync msgs in dyn queue = 0
Total number of rel sync msgs in dyn queue = 0
Total number of rx pkt msgs in standby dynamic queue
Total number of rx pkt msgs in active dyn queue = 0
Total number of rx pkts relayed = 0
Total number of rx pkts received = 5686578
Total number of dyn-sync messages received so far = 3
Total number of rel-sync pending complete = 0
Total number of L3 baseline-sync packets = 655
Total number of packet drops in sync = 0
Is image_sync_in_progress? = 0
Total num of rx dyn queue drops = 0
Total num of jumbo corrupts = 0
Total number of messages in IP send queue = 0
```

Syntax: show ipc_stat

Security Access

Table 19 lists the individual Brocade FastIron switches and the security access features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 19 Supported security access features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Authentication, Authorization and Accounting (AAA):	Yes	Yes	Yes	Yes	Yes
• RADIUS					
• TACACS/TACACS+					
AAA support for console commands	Yes	No	Yes	Yes	Yes
Restricting remote access to management functions	Yes	Yes	Yes	Yes	Yes
Disabling TFTP access	Yes	No	Yes	Yes	Yes
Using ACLs to restrict remote access	Yes	Yes	Yes	Yes	Yes
Local user accounts	Yes	Yes	Yes	Yes	Yes
Local user passwords	Yes	Yes	Yes	Yes	Yes
SSL security for the Web Management Interface	Yes	Yes	Yes	Yes	Yes
AAA authentication-method lists	Yes	Yes	Yes	Yes	Yes
Packet filtering on TCP flags	No	Yes	Yes	Yes	Yes

This chapter explains how to secure access to management functions on a Brocade device.

NOTE

For all Brocade devices, RADIUS Challenge is supported for 802.1x authentication but not for login authentication. Also, multiple challenges are supported for TACACS+ login authentication.

Securing access methods

The following table lists the management access methods available on a Brocade device, how they are secured by default, and the ways in which they can be secured.

TABLE 20 Ways to secure management access to Brocade devices

Access method	How the access method is secured by default	Ways to secure the access method	See page
Serial access to the CLI	Not secured	Establish passwords for management privilege levels	page 125
Access to the Privileged EXEC and CONFIG levels of the CLI	Not secured	Establish a password for Telnet access to the CLI	page 124
		Establish passwords for management privilege levels	page 125
		Set up local user accounts	page 129
		Configure TACACS/TACACS+ security	page 139
		Configure RADIUS security	page 157
Telnet access	Not secured	Regulate Telnet access using ACLs	page 113
		Allow Telnet access only from specific IP addresses	page 116
		Restrict Telnet access based on a client MAC address	page 117
		Allow Telnet access only from specific MAC addresses	page 119
		Define the Telnet idle time	page 118
		Change the Telnet login timeout period	page 118
		Specify the maximum number of login attempts for Telnet access	page 118
		Disable Telnet access	page 123
		Establish a password for Telnet access	page 124
		Establish passwords for privilege levels of the CLI	page 125
		Set up local user accounts	page 129
		Configure TACACS/TACACS+ security	page 139
		Configure RADIUS security	page 157

TABLE 20 Ways to secure management access to Brocade devices (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
Secure Shell (SSH) access	Not configured	Configure SSH	page 1419
		Regulate SSH access using ACLs	page 113
		Allow SSH access only from specific IP addresses	page 116
		Allow SSH access only from specific MAC addresses	page 117
		Establish passwords for privilege levels of the CLI	page 125
		Set up local user accounts	page 129
		Configure TACACS/TACACS+ security	page 139
		Configure RADIUS security	page 157
Web management access	SNMP read or read-write community strings	Regulate Web management access using ACLs	page 114
		Allow Web management access only from specific IP addresses	page 116
		Allow Web management access only to clients connected to a specific VLAN	page 119
		Disable Web management access	page 123
		Configure SSL security for the Web Management Interface	page 136
		Set up local user accounts	page 129
		Establish SNMP read or read-write community strings for SNMP versions 1 and 2	page 421
		Establishing user groups for SNMP version 3	page 427
		Configure TACACS/TACACS+ security	page 139
		Configure RADIUS security	page 157
SNMP access	SNMP read or read-write community strings and the password to the Super User privilege level NOTE: SNMP read or read-write community strings are always required for SNMP access to the device.	Regulate SNMP access using ACLs	page 114
		Allow SNMP access only from specific IP addresses	page 116
		Disable SNMP access	page 124
		Allow SNMP access only to clients connected to a specific VLAN	page 119
		Establish passwords to management levels of the CLI	page 125
		Set up local user accounts	page 129
		Establish SNMP read or read-write community strings	page 139

TABLE 20 Ways to secure management access to Brocade devices (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
TFTP access	Not secured	Allow TFTP access only to clients connected to a specific VLAN	page 120
		Disable TFTP access	page 124
Access for Stacked Devices	Access to multiple consoles must be secured after AAA is enabled	Extra steps must be taken to secure multiple consoles in an IronStack.	page 140

Remote access to management function restrictions

You can restrict access to management functions from remote sources, including Telnet, the Web Management Interface, and SNMP. The following methods for restricting remote access are supported:

- Using ACLs to restrict Telnet, Web Management Interface, or SNMP access
- Allowing remote access only from specific IP addresses
- Allowing Telnet and SSH access only from specific MAC addresses
- Allowing remote access only to clients connected to a specific VLAN
- Specifically disabling Telnet, Web Management Interface, or SNMP access to the device

The following sections describe how to restrict remote access to a Brocade device using these methods.

ACL usage to restrict remote access

You can use standard ACLs to control the following access methods to management functions on a Brocade device:

- Telnet
- SSH
- Web management
- SNMP

Consider the following to configure access control for these management access methods.

1. Configure an ACL with the IP addresses you want to allow to access the device.
2. Configure a Telnet access group, SSH access group, Web access group, and SNMP community strings. Each of these configuration items accepts an ACL as a parameter. The ACL contains entries that identify the IP addresses that can use the access method.

The following sections present examples of how to secure management access using ACLs. Refer to [Chapter 40, “Rule-Based IP ACLs”](#) for more information on configuring ACLs.

Using an ACL to restrict Telnet access

To configure an ACL that restricts Telnet access to the device, enter commands such as the following.

```
Brocade(config)#access-list 10 deny host 209.157.22.32 log
Brocade(config)#access-list 10 deny 209.157.23.0 0.0.0.255 log
Brocade(config)#access-list 10 deny 209.157.24.0 0.0.0.255 log
Brocade(config)#access-list 10 deny 209.157.25.0/24 log
Brocade(config)#access-list 10 permit any
Brocade(config)#telnet access-group 10
Brocade(config)#write memory
```

Syntax: `telnet access-group <num>`

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

The commands above configure ACL 10, then apply the ACL as the access list for Telnet access. The device allows Telnet access to all IP addresses except those listed in ACL 10.

To configure a more restrictive ACL, create permit entries and omit the **permit any** entry at the end of the ACL.

Example

```
Brocade(config)#access-list 10 permit host 209.157.22.32
Brocade(config)#access-list 10 permit 209.157.23.0 0.0.0.255
Brocade(config)#access-list 10 permit 209.157.24.0 0.0.0.255
Brocade(config)#access-list 10 permit 209.157.25.0/24
Brocade(config)#telnet access-group 10
Brocade(config)#write memory
```

The ACL in this example permits Telnet access only to the IP addresses in the **permit** entries and denies Telnet access from all other IP addresses.

Using an ACL to restrict SSH access

To configure an ACL that restricts SSH access to the device, enter commands such as the following.

```
Brocade(config)#access-list 12 deny host 209.157.22.98 log
Brocade(config)#access-list 12 deny 209.157.23.0 0.0.0.255 log
Brocade(config)#access-list 12 deny 209.157.24.0/24 log
Brocade(config)#access-list 12 permit any
Brocade(config)#ssh access-group 12
Brocade(config)#write memory
```

Syntax: `ssh access-group <num>`

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACL 12, then apply the ACL as the access list for SSH access. The device denies SSH access from the IP addresses listed in ACL 12 and permits SSH access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny SSH access from all IP addresses.

NOTE

In this example, the command **ssh access-group 10** could have been used to apply the ACL configured in the example for Telnet access. You can use the same ACL multiple times.

Using an ACL to restrict Web management access

To configure an ACL that restricts Web management access to the device, enter commands such as the following.

```
Brocade(config)#access-list 12 deny host 209.157.22.98 log
Brocade(config)#access-list 12 deny 209.157.23.0 0.0.0.255 log
Brocade(config)#access-list 12 deny 209.157.24.0/24 log
Brocade(config)#access-list 12 permit any
Brocade(config)#web access-group 12
Brocade(config)#write memory
```

Syntax: `web access-group <num>`

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACL 12, then apply the ACL as the access list for Web management access. The device denies Web management access from the IP addresses listed in ACL 12 and permits Web management access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny Web management access from all IP addresses.

Using ACLs to restrict SNMP access

To restrict SNMP access to the device using ACLs, enter commands such as the following.

NOTE

The syntax for using ACLs for SNMP access is different from the syntax for controlling Telnet, SSH, and Web management access using ACLs.

```
Brocade(config)#access-list 25 deny host 209.157.22.98 log
Brocade(config)#access-list 25 deny 209.157.23.0 0.0.0.255 log
Brocade(config)#access-list 25 deny 209.157.24.0 0.0.0.255 log
Brocade(config)#access-list 25 permit any
Brocade(config)#access-list 30 deny 209.157.25.0 0.0.0.255 log
Brocade(config)#access-list 30 deny 209.157.26.0/24 log
Brocade(config)#access-list 30 permit any
Brocade(config)#snmp-server community public ro 25
Brocade(config)#snmp-server community private rw 30
Brocade(config)#write memory
```

Syntax: `snmp-server community <string> ro | rw <num>`

The `<string>` parameter specifies the SNMP community string the user must enter to gain SNMP access.

The `ro` parameter indicates that the community string is for read-only (“get”) access. The `rw` parameter indicates the community string is for read-write (“set”) access.

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACLs 25 and 30, then apply the ACLs to community strings.

ACL 25 is used to control read-only access using the “public” community string. ACL 30 is used to control read-write access using the “private” community string.

NOTE

When **snmp-server community** is configured, all incoming SNMP packets are validated first by their community strings and then by their bound ACLs.

Defining the console idle time

By default, a Brocade device does not time out serial console sessions. A serial session remains open indefinitely until you close it. You can however define how many minutes a serial management session can remain idle before it is timed out.

NOTE

You must enable AAA support for console commands, AAA authentication, and Exec authorization in order to set the console idle time.

To configure the idle time for a serial console session, use the following command.

```
Brocade(config)#console timeout 120
```

Syntax: [no] console timeout <0 – 240>

Possible values: 0 – 240 minutes

Default value: 0 minutes (no timeout)

NOTE

In RADIUS, the standard attribute Idle-Timeout is used to define the console session timeout value. The attribute Idle-Timeout value is specified in seconds. Within the switch, it is truncated to the nearest minute, because the switch configuration is defined in minutes.

Remote access restrictions

By default, a Brocade device does not control remote management access based on the IP address of the managing device. You can restrict remote management access to a single IP address for the following access methods:

- Telnet access
- SSH access
- Web management access
- SNMP access

In addition, you can restrict all access methods to the same IP address using a single command.

The following examples show the CLI commands for restricting remote access. You can specify only one IP address with each command. However, you can enter each command ten times to specify up to ten IP addresses.

NOTE

You cannot restrict remote management access using the Web Management Interface.

Restricting Telnet access to a specific IP address

To allow Telnet access to the Brocade device only to the host with IP address 209.157.22.39, enter the following command.

```
Brocade(config)#telnet-client 209.157.22.39
```

Syntax: [no] telnet-client <ip-addr> | <ipv6-addr>

Restricting SSH access to a specific IP address

To allow SSH access to the Brocade device only to the host with IP address 209.157.22.39, enter the following command.

```
Brocade(config)#ip ssh client 209.157.22.39
```

Syntax: [no] ip ssh client <ip-addr> | <ipv6-addr>

Restricting Web management access to a specific IP address

To allow Web management access to the Brocade device only to the host with IP address 209.157.22.26, enter the following command.

```
Brocade(config)#web-client 209.157.22.26
```

Syntax: [no] web-client <ip-addr> | <ipv6-addr>

Restricting SNMP access to a specific IP address

To allow SNMP access only to the host with IP address 209.157.22.14, enter the following command.

```
Brocade(config)#snmp-client 209.157.22.14
```

Syntax: [no] snmp-client <ip-addr> | <ipv6-addr>

Restricting all remote management access to a specific IP address

To allow Telnet, Web, and SNMP management access to the Brocade device only to the host with IP address 209.157.22.69, enter three separate commands (one for each access type) or enter the following command.

```
Brocade(config)#all-client 209.157.22.69
```

Syntax: [no] all-client <ip-addr> | <ipv6-addr>

Restricting access to the device based on IP or MAC address

You can restrict remote management access to the Brocade device, using Telnet, SSH, HTTP, and HTTPS, based on the connecting client IP or MAC address.

Restricting Telnet connection

You can restrict Telnet connection to a device based on the client IP address or MAC address.

To allow Telnet access to the Brocade device only to the host with IP address 209.157.22.39 **and** MAC address 0007.e90f.e9a0, enter the following command.

```
Brocade(config)#telnet client 209.157.22.39 0007.e90f.e9a0
```

Syntax: [no] telnet client <ip-addr> | <ipv6-addr> <mac-addr>

NOTE

For FCX devices, this feature applies only to IPv4 clients.

The following command allows Telnet access to the Brocade device to a host with any IP address and MAC address 0007.e90f.e9a0.

```
Brocade(config)#telnet client any 0007.e90f.e9a0
```

Syntax: [no] telnet client any <mac-addr>

Restricting SSH connection

You can restrict SSH connection to a device based on the client IP address or MAC address.

To allow SSH access to the Brocade device only to the host with IP address 209.157.22.39 **and** MAC address 0007.e90f.e9a0, enter the following command.

```
Brocade(config)#ip ssh client 209.157.22.39 0007.e90f.e9a0
```

Syntax: [no] ip ssh client <ip-addr> | <ipv6-addr> <mac-addr>

To allow SSH access to the Brocade device to a host with any IP address and MAC address 0007.e90f.e9a0, enter the following command.

```
Brocade(config)#ip ssh client any 0007.e90f.e9a0
```

Syntax: [no] ip ssh client any <mac-addr>

Restricting HTTP and HTTPS connection

You can restrict an HTTP or HTTPS connection to a device based on the client IP address or MAC address.

To allow HTTP and HTTPS access to the Brocade device only to the host with IP address 209.157.22.40 **and** MAC address 0007.e90f.ab1c, enter the following command.

```
Brocade(config)#web client 209.157.22.40 0007.e90f.ab1c
```

Syntax: [no] web client <ip-addr> | <ipv6-addr> <mac-addr>

The following command allows HTTP and HTTPS access to the Brocade device to a host with any IP address and MAC address 0007.e90f.10ba.

```
Brocade(config)#web client any 0007.e90f.10ba
```

Syntax: [no] web client any <mac-addr>

Defining the Telnet idle time

You can define how many minutes a Telnet session can remain idle before it is timed out. An idle Telnet session is a session that is still sending TCP ACKs in response to keepalive messages from the device, but is not being used to send data.

To configure the idle time for a Telnet session, use the following command.

```
Brocade(config)#telnet timeout 120
```

Syntax: [no] telnet timeout <minutes>

For <minutes> enter a value from 0 – 240. The default value is 0 minutes (no timeout).

Changing the login timeout period for Telnet sessions

By default, the login timeout period for a Telnet session is 1 minute. To change the login timeout period, use the following command.

```
Brocade(config)#telnet login-timeout 5
```

Syntax: [no] telnet login-timeout <minutes>

For <minutes>, enter a value from 1 to 10. The default timeout period is 1 minute.

Specifying the maximum number of login attempts for Telnet access

If you are connecting to the Brocade device using Telnet, the device prompts you for a username and password. By default, you have up to 4 chances to enter a correct username and password. If you do not enter a correct username or password after 4 attempts, the Brocade device disconnects the Telnet session.

You can specify the number of attempts a Telnet user has to enter a correct username and password before the device disconnects the Telnet session. For example, to allow a Telnet user up to 5 chances to enter a correct username and password, enter the following command.

```
Brocade(config)#telnet login-retries 5
```

Syntax: [no] telnet login-retries <number>

You can specify from 0 – 5 attempts. The default is 4 attempts.

Changing the login timeout period for Telnet sessions

To change the login timeout period for Telnet sessions to 5 minutes, enter the following command:

```
Brocade(config)# telnet login-timeout 5
```

Syntax: [no] telnet login-timeout <minutes>

For <minutes>, specify a value from 1 – 10. The default is 2 minutes.

Restricting remote access to the device to specific VLAN IDs

You can restrict management access to a Brocade device to ports within a specific port-based VLAN. VLAN-based access control applies to the following access methods:

- Telnet access
- Web management access
- SNMP access
- TFTP access

By default, access is allowed for all the methods listed above on all ports. Once you configure security for a given access method based on VLAN ID, access to the device using that method is restricted to only the ports within the specified VLAN.

VLAN-based access control works in conjunction with other access control methods. For example, suppose you configure an ACL to permit Telnet access only to specific client IP addresses, and you also configure VLAN-based access control for Telnet access. In this case, the only Telnet clients that can access the device are clients that have one of the IP addresses permitted by the ACL **and** are connected to a port that is in a permitted VLAN. Clients who have a permitted IP address but are connected to a port in a VLAN that is not permitted still cannot access the device through Telnet.

Restricting Telnet access to a specific VLAN

To allow Telnet access only to clients in a specific VLAN, enter a command such as the following.

```
Brocade(config)#telnet server enable vlan 10
```

The command in this example configures the device to allow Telnet management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] telnet server enable vlan <vlan-id>

Restricting Web management access to a specific VLAN

To allow Web management access only to clients in a specific VLAN, enter a command such as the following.

```
Brocade(config)#web-management enable vlan 10
```

The command in this example configures the device to allow Web management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] web-management enable vlan <vlan-id>

Restricting SNMP access to a specific VLAN

To allow SNMP access only to clients in a specific VLAN, enter a command such as the following.

```
Brocade(config)#snmp-server enable vlan 40
```

The command in this example configures the device to allow SNMP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] snmp-server enable vlan <vlan-id>

Restricting TFTP access to a specific VLAN

To allow TFTP access only to clients in a specific VLAN, enter a command such as the following.

```
Brocade(config)#tftp client enable vlan 40
```

The command in this example configures the device to allow TFTP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] tftp client enable vlan <vlan-id>

Designated VLAN for Telnet management sessions to a Layer 2 Switch

All Brocade FastIron devices support the creation of management VLANs. By default, the management IP address you configure on a Layer 2 Switch applies globally to all the ports on the device. This is true even if you divide the device ports into multiple port-based VLANs.

If you want to restrict the IP management address to a specific port-based VLAN, you can make that VLAN the designated management VLAN for the device. When you configure a VLAN to be the designated management VLAN, the management IP address you configure on the device is associated only with the ports in the designated VLAN. To establish a Telnet management session with the device, a user must access the device through one of the ports in the designated VLAN.

You also can configure up to five default gateways for the designated VLAN, and associate a metric with each one. The software uses the gateway with the lowest metric. The other gateways reside in the configuration but are not used. To use one of the other gateways, modify the configuration so that the gateway you want to use has the lowest metric.

If more than one gateway has the lowest metric, the gateway that appears first in the running-config is used.

NOTE

If you have already configured a default gateway globally and you do not configure a gateway in the VLAN, the software uses the globally configured gateway and gives the gateway a metric value of 1.

To configure a designated management VLAN, enter commands such as the following.

```
Brocade(config)#vlan 10 by port
Brocade(config-vlan-10)#untag ethernet 1/1 to 1/4
Brocade(config-vlan-10)#management-vlan
Brocade(config-vlan-10)#default-gateway 10.10.10.1 1
Brocade(config-vlan-10)#default-gateway 20.20.20.1 2
```

These commands configure port-based VLAN 10 to consist of ports 1/1 – 1/4 and to be the designated management VLAN. The last two commands configure default gateways for the VLAN. Since the 10.10.10.1 gateway has a lower metric, the software uses this gateway. The other gateway remains in the configuration but is not used. You can use the other one by changing the metrics so that the 20.20.20.1 gateway has the lower metric.

Syntax: `[no] default-gateway <ip-addr> <metric>`

The `<ip-addr>` parameters specify the IP address of the gateway router.

The `<metric>` parameter specifies the metric (cost) of the gateway. You can specify a value from 1 – 5. There is no default. The software uses the gateway with the lowest metric.

Device management security

By default, all management access is disabled. Each of the following management access methods must be specifically enabled as required in your installation:

- SSHv2
- SNMP
- Web management through HTTP
- Web management through HTTPS

The commands for granting access to each of these management interfaces is described in the following.

Allowing SSHv2 access to the Brocade device

To allow SSHv2 access to the Brocade device, you must generate a Crypto Key as shown in the following command.

```
Brocade(config)#crypto key generate
```

Syntax: `crypto key [generate | zeroize]`

The **generate** parameter generates a dsa key pair.

The **zeroize** parameter deletes the currently operative dsa key pair.

In addition, you must use AAA authentication to create a password to allow SSHv2 access. For example the following command configures AAA authentication to use TACACS+ for authentication as the default or local if TACACS+ is not available.

```
Brocade(config)#aaa authentication login default tacacs+ local
```

Allowing SNMP access to the Brocade device

To allow SNMP access to the Brocade device, enter the following command.

```
Brocade(config)#snmp-server
```

Syntax: `[no] snmp-server`

Allowing Web management through HTTP for the Brocade device

To allow web management through HTTP for the Brocade device, you enable web management as shown in the following command.

```
Brocade(config)#web-management http
```

Syntax: [no] web-management http | https

When using the web-management command, specify the **http** or **https** parameters.

The **http** parameter specifies that web management is enabled for HTTP access.

The **https** parameter specifies that web management is enabled for HTTPS access.

Allowing Web management through HTTPS

To allow web management through HTTPS, you must enable web management as shown in [“Allowing Web management through HTTP for the Brocade device”](#). Additionally, you must generate a crypto SSL certificate or import digital certificates issued by a third-party Certificate Authority (CA).

To generate a crypto SSL certificate use the following command.

```
Brocade(config)#crypto-ssl certificate generate
```

Syntax: crypto-ssl certificate [generate | zeroize]

Using the web-management command without the http or https option makes web management available for both.

The **generate** parameter generates an ssl certificate.

The **zeroize** parameter deletes the currently operative ssl certificate.

To import a digital certificate issued by a third-party Certificate Authority (CA) and save it in the flash memory, use the following command.

```
Brocade(config)#ip ssl certificate-data-file tftp 10.10.10.1 cacert.pem
```

Syntax: ip ssl certificate-data-file tftp <ip-addr> <file-name>

The <ip-addr> variable is the IP address of the TFTP server from which the digital certificate file is being downloaded.

The <file-name> variable is the file name of the digital certificate that you are importing to the router.

Disabling specific access methods

You can specifically disable the following access methods:

- Telnet access
- Web management access
- SNMP access
- TFTP

NOTE

If you disable Telnet access, you will not be able to access the CLI except through a serial connection to the management module. If you disable SNMP access, you will not be able to use an SNMP-based management applications.

Disabling Telnet access

You can use a Telnet client to access the CLI on the device over the network. If you do not plan to use the CLI over the network and want to disable Telnet access to prevent others from establishing CLI sessions with the device, enter the following command.

```
Brocade(config)#no telnet server
```

To re-enable Telnet operation, enter the following command.

```
Brocade(config)#telnet server
```

Syntax: [no] telnet server

Disabling Web management access

If you want to prevent access to the device through the Web Management Interface, you can disable the Web Management Interface.

NOTE

As soon as you make this change, the device stops responding to Web management sessions. If you make this change using your Web browser, your browser can contact the device, but the device will not reply once the change takes place.

To disable the Web Management Interface, enter the following command.

```
Brocade(config)#no web-management
```

Syntax: [no] web-management [http | https]

Use the **no web-management** command with no option specified to disable both web management through http access and web management through https access.

Use the command **no web-management http** to disable only web management through http access.

Use the command **no web-management https** to disable only web management through https access.

Disabling Web management access by HP ProCurve Manager

By default, TCP ports 80 and 280 are enabled on the Brocade device. TCP port 80 (HTTP) allows access to the device Web Management Interface. TCP port 280 allows access to the device by HP ProCurve Manager.

The **no web-management** command disables both TCP ports. However, if you want to disable only port 280 and leave port 80 enabled, use the **hp-top-tools** option with the command. Here is an example.

```
Brocade(config)#no web-management hp-top-tools
```

Syntax: [no] web-management [allow-no-password | enable [vlan <vlan-id>] | front-panel | hp-top-tools | list-menu]

The **hp-top-tools** parameter disables TCP port 280.

Disabling SNMP access

To disable SNMP management of the device.

```
Brocade(config)#no snmp-server
```

To later re-enable SNMP management of the device.

```
Brocade(config)#snmp-server
```

Syntax: no snmp-server

Disabling TFTP access

You can globally disable TFTP to block TFTP client access. By default, TFTP client access is enabled.

To disable TFTP client access, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)#tftp disable
```

When TFTP is disabled, users are prohibited from using the **copy tftp** command to copy files to the system flash. If users enter this command while TFTP is disabled, the system will reject the command and display an error message.

To re-enable TFTP client access once it is disabled, enter the following command.

```
Brocade(config)#no tftp disable
```

Syntax: [no] tftp disable

Passwords used to secure access

Passwords can be used to secure the following access methods:

- Telnet access can be secured by setting a Telnet password. Refer to [“Setting a Telnet password”](#) on page 124.
- Access to the Privileged EXEC and CONFIG levels of the CLI can be secured by setting passwords for management privilege levels. Refer to [“Setting passwords for management privilege levels”](#) on page 125.

This section also provides procedures for enhancing management privilege levels, recovering from a lost password, and disabling password encryption.

NOTE

You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account a management privilege level. Refer to [“Local user accounts”](#) on page 129.

Setting a Telnet password

By default, the device does not require a user name or password when you log in to the CLI using Telnet. You can assign a password for Telnet access using one of the following methods.

Set the password “letmein” for Telnet access to the CLI using the following command at the global CONFIG level.

```
Brocade(config)#enable telnet password letmein
```

Syntax: [no] enable telnet password <string>

Suppressing Telnet connection rejection messages

By default, if a Brocade device denies Telnet management access to the device, the software sends a message to the denied Telnet client. You can optionally suppress the rejection message. When you enable the option, a denied Telnet client does not receive a message from the Brocade device. Instead, the denied client simply does not gain access.

To suppress the connection rejection message, use the following CLI method.

To suppress the connection rejection message sent by the device to a denied Telnet client, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)#telnet server suppress-reject-message
```

Syntax: [no] telnet server suppress-reject-message

Setting passwords for management privilege levels

You can set one password for each of the following management privilege levels:

- **Super User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords.
- **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters.
- **Read Only level** – Allows access to the Privileged EXEC mode and User EXEC mode of the CLI but only with read access.

You can assign a password to each management privilege level. You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account to one of the three privilege levels. Refer to [“Local user accounts”](#) on page 129.

NOTE

You must use the CLI to assign a password for management privilege levels. You cannot assign a password using the Web Management Interface.

If you configure user accounts in addition to privilege level passwords, the device will validate a user access attempt using one or both methods (local user account or privilege level password), depending on the order you specify in the authentication-method lists. Refer to [“Authentication-method lists”](#) on page 174.

Follow the steps given below to set passwords for management privilege levels.

1. At the opening CLI prompt, enter the following command to change to the Privileged level of the EXEC mode.

```
Brocade> enable
Brocade#
```

2. Access the CONFIG level of the CLI by entering the following command.

```
Brocade#configure terminal
Brocade(config)#
```

3. Enter the following command to set the Super User level password.

```
Brocade(config)#enable super-user-password <text>
```

NOTE

You must set the Super User level password before you can set other types of passwords. The Super User level password can be an alphanumeric string, but cannot begin with a number.

4. Enter the following commands to set the Port Configuration level and Read Only level passwords.

```
Brocade(config)#enable port-config-password <text>
Brocade(config)#enable read-only-password <text>
```

Syntax: enable super-user-password <text>

Syntax: enable port-config-password <text>

Syntax: enable read-only-password <text>

NOTE

If you forget your Super User level password, refer to [“Recovering from a lost password”](#) on page 127.

Augmenting management privilege levels

Each management privilege level provides access to specific areas of the CLI by default:

- Super User level provides access to all commands and displays.
- Port Configuration level gives access to:
 - The User EXEC and Privileged EXEC levels
 - The port-specific parts of the CONFIG level
 - All interface configuration levels
- Read Only level gives access to:
 - The User EXEC and Privileged EXEC levels

You can grant additional access to a privilege level on an individual command basis. To grant the additional access, you specify the privilege level you are enhancing, the CLI level that contains the command, and the individual command.

NOTE

This feature applies only to management privilege levels on the CLI. You cannot augment management access levels for the Web Management Interface.

Enhance the Port Configuration privilege level so users also can enter IP commands at the global CONFIG level.

```
Brocade(config)#privilege configure level 4 ip
```

In this command, **configure** specifies that the enhanced access is for a command at the global CONFIG level of the CLI. The **level 4** parameter indicates that the enhanced access is for management privilege level 4 (Port Configuration). All users with Port Configuration privileges will have the enhanced access. The **ip** parameter indicates that the enhanced access is for the IP commands. Users who log in with valid Port Configuration level user names and passwords can enter commands that begin with “ip” at the global CONFIG level.

Syntax: `[no] privilege <cli-level> level <privilege-level> <command-string>`

The `<cli-level>` parameter specifies the CLI level and can be one of the following values:

- **exec** – EXEC level; for example, Brocade> or Brocade#
- **configure** – CONFIG level; for example, Brocade(config)#
- **interface** – Interface level; for example, Brocade(config-if-6)#
- **loopback-interface** – loopback interface level
- **virtual-interface** – Virtual-interface level; for example, Brocade(config-vif-6)#
- **dot1x** – 802.1X configuration level
- **ipv6-access-list** – IPv6 access list configuration level
- **rip-router** – RIP router level; for example, Brocade(config-rip-router)#
- **ospf-router** – OSPF router level; for example, Brocade(config-ospf-router)#
- **dvmrp-router** – DVMRP router level; for example, Brocade(config-dvmrp-router)#
- **pim-router** – PIM router level; for example, Brocade(config-pim-router)#
- **bgp-router** – BGP4 router level; for example, Brocade(config-bgp-router)#
- **vrrp-router** – VRRP configuration level
- **gvrp** – GVRP configuration level
- **trunk** – trunk configuration level
- **port-vlan** – Port-based VLAN level; for example, Brocade(config-vlan)#
- **protocol-vlan** – Protocol-based VLAN level

The `<privilege-level>` indicates the number of the management privilege level you are augmenting. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The `<command-string>` parameter specifies the command you are allowing users with the specified privilege level to enter. To display a list of the commands at a CLI level, enter “?” at that level's command prompt.

Recovering from a lost password

Recovery from a lost password requires direct access to the serial port and a system reset.

NOTE

You can perform this procedure only from the CLI.

Follow the steps given below to recover from a lost password.

1. Start a CLI session over the serial interface to the device.
2. Reboot the device.
3. At the initial boot prompt at system startup, enter **b** to enter the boot monitor mode.
4. Enter **no password** at the prompt. (You cannot abbreviate this command.) This command will cause the device to bypass the system password check.

5. Enter **boot system flash primary** at the prompt. On ICX 6430 and ICX 6450 devices, enter **boot_primary**.
6. After the console prompt reappears, assign a new password.

Displaying the SNMP community string

If you want to display the SNMP community string, enter the following commands.

```
Brocade(config)#enable password-display  
Brocade#show snmp server
```

The **enable password-display** command enables display of the community string, but only in the output of the **show snmp server** command. Display of the string is still encrypted in the startup-config file and running-config. Enter the command at the global CONFIG level of the CLI.

Specifying a minimum password length

By default, the Brocade device imposes no minimum length on the Line (Telnet), Enable, or Local passwords. You can configure the device to require that Line, Enable, and Local passwords be at least a specified length.

For example, to specify that the Line, Enable, and Local passwords be at least 8 characters, enter the following command.

```
Brocade(config)#enable password-min-length 8
```

Syntax: **enable password-min-length** *<number-of-characters>*

The *<number-of-characters>* can be from 1 – 48.

Local user accounts

You can define up to 16 local user accounts on a Brocade device. User accounts regulate who can access the management functions in the CLI using the following methods:

- Telnet access
- Web management access
- SNMP access

Local user accounts provide greater flexibility for controlling management access to Brocade devices than do management privilege level passwords and SNMP community strings of SNMP versions 1 and 2. You can continue to use the privilege level passwords and the SNMP community strings as additional means of access authentication. Alternatively, you can choose not to use local user accounts and instead continue to use only the privilege level passwords and SNMP community strings. Local user accounts are backward-compatible with configuration files that contain privilege level passwords. Refer to [“Setting passwords for management privilege levels”](#) on page 125.

If you configure local user accounts, you also need to configure an authentication-method list for Telnet access, Web management access, and SNMP access. Refer to [“Authentication-method lists”](#) on page 174.

For each local user account, you specify a user name. You also can specify the following parameters:

- A password
- A management privilege level, which can be one of the following:
 - **Super User level (default)** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only privilege level that allows you to configure passwords.
 - **Port Configuration level** – Allows read-and-write access for specific ports but not for global parameters.
 - **Read Only level** – Allows access to the Privileged EXEC mode and User EXEC mode with read access only.
- You can set additional username and password rules. Refer to [“Enhancements to username and password”](#).

Enhancements to username and password

This section describes the enhancements to the username and password features introduced in earlier releases.

The following rules are enabled by default:

- Users are required to accept the message of the day.
- Users are locked out (disabled) if they fail to login after three attempts. This feature is automatically enabled. Use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.

The following rules are disabled by default:

- Enhanced user password combination requirements
- User password masking

- Quarterly updates of user passwords
- You can configure the system to store up to 15 previously configured passwords for each user.
- You can use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.
- A password can now be set to expire.

Enabling enhanced user password combination requirements

When strict password enforcement is enabled on the Brocade device, you must enter a minimum of eight characters containing the following combinations when you create an enable and a user password:

- At least two upper case characters
- At least two lower case characters
- At least two numeric characters
- At least two special characters

NOTE

Password minimum and combination requirements are strictly enforced.

Use the **enable strict-password-enforcement** command to enable the password security feature.

```
Brocade(config)#enable strict-password-enforcement
```

Syntax: [no] **enable strict-password-enforcement**

This feature is disabled by default.

The following security upgrades apply to the **enable strict-password-enforcement** command:

- Passwords must not share four or more concurrent characters with any other password configured on the router. If the user tries to create a password with four or more concurrent characters, the following error message will be returned.

```
Error - The substring <str> within the password has been used earlier, please  
choose a different password.
```

For example, the previous password was Mali4aYa&, the user cannot use any of the following as his or her new password:

- Malimai\$D because “Mail” were used consecutively in the previous password
- &3B9aYa& because “aYa&” were used consecutively in the previous password
- i4aYEv#8 because “i4aY” were used consecutively in the previous password
- If the user tries to configure a password that was previously used, the Local User Account configuration will not be allowed and the following message will be displayed.

```
This password was used earlier for same or different user, please choose a  
different password.
```


Enabling user password masking

By default, when you use the CLI to create a user password, the password displays on the console as you type it. For enhanced security, you can configure the Brocade device to mask the password characters entered at the CLI. When password masking is enabled, the CLI displays asterisks (*) on the console instead of the actual password characters entered.

The following shows the default CLI behavior when configuring a username and password.

```
Brocade(config)#username kelly password summertime
```

The following shows the CLI behavior when configuring a username and password when **password-masking** is enabled.

```
Brocade(config)#username kelly password
Enter Password: *****
```

NOTE

When password masking is enabled, press the [Enter] key before entering the password.

Syntax: `username <name> password [Enter]`

For [Enter], press the Enter key. Enter the password when prompted.

If **strict-password-enforcement** is enabled, enter a password which contains the required character combination. Refer to [“Enabling enhanced user password combination requirements”](#) on page 130.

To enable password masking, enter the following command.

```
Brocade(config)#enable user password-masking
```

Syntax: `[no] enable user password-masking`

Enabling user password aging

For enhanced security, password aging enforces quarterly updates of all user passwords. After 180 days, the CLI will automatically prompt users to change their passwords when they attempt to sign on.

When password aging is enabled, the software records the system time that each user password was configured or last changed. The time displays in the output of the **show running configuration** command, indicated by `set-time <time>`.

Example

```
Brocade#show run
Current configuration:
....
username waldo password .....
username raveen set-time 2086038248
....
```

The password aging feature uses the SNTP server clock to record the set-time. If the network does not have an SNTP server, then set-time will appear as **set-time 0** in the output of the **show running configuration** command.

A username set-time configuration is removed when:

- The username and password is deleted from the configuration
- The username password expires

When a username set-time configuration is removed, it no longer appears in the **show running configuration** output.

Note that if a username does not have an assigned password, the username will not have a set-time configuration.

Password aging is disabled by default. To enable it, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)#enable user password-aging
```

Syntax: [no] enable user password-aging

Configuring password history

By default, the Brocade device stores the last five user passwords for each user. When changing a user password, the user cannot use any of the five previously configured passwords.

For security purposes, you can configure the Brocade device to store up to 15 passwords for each user, so that users do not use the same password multiple times. If a user attempts to use a password that is stored, the system will prompt the user to choose a different password.

To configure enhanced password history, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#enable user password-history 15
```

Syntax: [no] enable user password-history <1 - 15>

Enhanced login lockout

The CLI provides up to three login attempts. If a user fails to login after three attempts, that user is locked out (disabled). If desired, you can increase or decrease the number of login attempts before the user is disabled. To do so, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#enable user disable-on-login-failure 7
```

Syntax: enable user disable-on-login-failure <1 - 10>

To re-enable a user that has been locked out, do one of the following:

- Reboot the Brocade device to re-enable all disabled users.
- Enable the user by entering the following command.

```
Brocade(config)#username sandy enable
```

Example

```

Brocade(config)#user sandy enable
Brocade#show user
Username Password Encrypt Priv Status Expire Time
=====
sandy $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled 0 enabled 90 days

```

Syntax: `username <name> enable`

Setting passwords to expire

You can set a user password to expire. Once a password expires, the administrator must assign a new password to the user. To configure a user password to expire, enter the following.

```
Brocade(config)#username sandy expires 20
```

Syntax: `username <name> expires <days>`

Enter 1 – 365 for number of days. The default is 90 days.

Example

```

Brocade(config)#username sandy expires 20
Brocade#show user
Username Password Encrypt Priv Status Expire Time
=====
sandy $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled 0 enabled 20 days

```

Requirement to accept the message of the day

If a message of the day (MOTD) is configured, a user will be required to press the Enter key before he or she can login. MOTD is configured using the **banner motd** command.

There are no new CLI commands for this feature.

NOTE

This requirement is disabled by default, unless configured. Users are not required to press Enter after the MOTD banner is displayed. Refer to [“Requiring users to press the Enter key after the message of the day banner”](#) on page 38.

Local user account configuration

You can create accounts for local users with or without passwords. Accounts with passwords can have encrypted or unencrypted passwords.

You can assign privilege levels to local user accounts, but on a new device, you must create a local user account that has a Super User privilege before you can create accounts with other privilege levels.

NOTE

You must grant Super User level privilege to at least one account before you add accounts with other privilege levels. You need the Super User account to make further administrative changes.

Local user accounts with no passwords

To create a user account without a password, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)#username wonka nopassword
```

Syntax: [no] **username** <user-string> **privilege** <privilege-level> **nopassword**

Local user accounts with unencrypted passwords

If you want to use unencrypted passwords for local user accounts, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#username wonka password willy
```

If password masking is enabled, press the [Enter] key before entering the password.

```
Brocade(config)#username wonka
Enter Password: willy
```

The above commands add a local user account with the user name “wonka” and the password “willy”. This account has the Super User privilege level; this user has full access to all configuration and display features.

```
Brocade(config)#username waldo privilege 5 password whereis
```

This command adds a user account for user name “waldo”, password “whereis”, with the Read Only privilege level. Waldo can look for information but cannot make configuration changes.

Syntax: [no] **username** <user-string> **privilege** <privilege-level> **password** | **nopassword** <password-string>

You can enter up to 48 characters for <user-string>.

The **privilege** <privilege-level> parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The default privilege level is **0**. If you want to assign Super User level access to the account, you can enter the command without **privilege 0**, as shown in the command example above.

The **password** | **nopassword** parameter indicates whether the user must enter a password. If you specify **password**, enter the string for the user's password. You can enter up to 48 characters for <password-string>. If **strict password enforcement** is enabled on the device, you must enter a minimum of eight characters containing the following combinations:

- At least two upper case characters
- At least two lower case characters

- At least two numeric characters
- At least two special characters

NOTE

You must be logged on with Super User access (privilege level 0) to add user accounts or configure other access parameters.

To display user account information, enter the following command.

```
Brocade#show users
```

Syntax: `show users`

Local accounts with encrypted passwords

You can create local user accounts with MD5 encrypted passwords using one of the following methods:

- Issuing the **service password-encryption** command after creating the local user account with a **username <user-string> [privilege <privilege-level>] password 0** command
- Using the **username <user-string> create-password** command

NOTE

To create an encrypted all-numeric password, use the **username <user-string> create-password** command.

If you create a local user account using the commands discussed in [“Local user accounts with unencrypted passwords”](#) on page 134, you can issue the **service password-encryption** command to encrypt all passwords that have been previously entered.

Example

```
Brocade(config)#username wonka privilege 5 password willy
Brocade(config)#service password-encryption
```

Creating a password option

As an alternative to the commands above, the **create-password** option allows you to create an encrypted password in one line of command. Also, this new option allows you to create an all-numeric, encrypted password.

You can enter.

```
Brocade(config)#username wonka privilege 5 create-password willy
```

Syntax: `[no] username <user-string> [privilege <privilege-level>] create-password <password-string>`

You can enter up to 48 characters for <user-string>. This string can be alphanumeric or all-numeric.

The **privilege** parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level

- 5 – Read Only level

Enter up to 255 alphanumeric characters for *<password-string>*.

Changing a local user password

To change a local user password for an existing local user account, enter a command such as the following at the global CONFIG level of the CLI.

NOTE

You must be logged on with Super User access (privilege level 0) to change user passwords.

```
Brocade(config)#username wonka password willy
```

If password masking is enabled, enter the username, press the [Enter] key, then enter the password.

```
Brocade(config)#username wonka password
Enter Password: willy
```

The above commands change wonka's user name password to "willy".

Syntax: [no] **username** *<user-string>* **password** *<password-string>*

Enter up to 48 characters for *<user-string>*.

The *<password-string>* parameter is the user password. The password can be up to 48 characters and must differ from the current password and two previously configured passwords.

When a password is changed, a message such as the following is sent to the Syslog.

```
SYSLOG: <14>Jan 1 00:00:00 10.44.9.11 Security: Password has been changed for user
tester from console session.
```

The message includes the name of the user whose password was changed and during which session type, such as Console, Telnet, SSH, Web, SNMP, or others, the password was changed.

SSL security for the Web Management Interface

The Brocade device supports Secure Sockets Layer / Transport Level Security (SSL 3.0 / TLS 1.0) for configuring the device using the Web Management Interface.

When enabled, the SSL protocol uses digital certificates and public-private key pairs to establish a secure connection to the Brocade device. Digital certificates serve to prove the identity of a connecting client, and public-private key pairs provide a means to encrypt data sent between the device and the client.

Configuring SSL for the Web Management Interface consists of the following tasks:

- Optionally enabling the SSL server on the Brocade device

NOTE

The SSL server is automatically enabled when an SSL certificate is generated.

- Importing an RSA certificate and private key file from a client (optional)
- Generating a certificate

Enabling the SSL server on the Brocade device

To enable the SSL server on the Brocade device, enter the following command.

```
Brocade(config)#web-management https
```

Syntax: [no] web-management http | https

You can enable either the HTTP or HTTPS servers with this command. You can disable both the HTTP and HTTPS servers by entering the following command.

```
Brocade(config)#no web-management
```

Syntax: no web-management

Specifying a port for SSL communication

By default, SSL protocol exchanges occur on TCP port 443. You can optionally change the port number used for SSL communication.

For example, the following command causes the device to use TCP port 334 for SSL communication.

```
Brocade(config)#ip ssl port 334
```

Syntax: [no] ip ssl port <port-number>

The default port for SSL communication is 443.

Changing the SSL server certificate key size

The default key size for Brocade-issued and imported digital certificates is 1024 bits. If desired, you can change the default key size to a value of 512, 2048, or 4096 bits. To do so, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#ip ssl cert-key-size 512
```

Syntax: ip ssl cert-key-size <512/ 1024/ 2048/ 4096>

NOTE

The SSL server certificate key size applies only to digital certificates issued by Brocade and does not apply to imported certificates.

Support for SSL digital certificates larger than 2048 bits

Brocade devices have the ability to store and retrieve SSL digital certificates that are up to 4000 bits in size.

Support for SSL certificates larger than 2048 bits is automatically enabled. You do not need to perform any configuration procedures to enable it.

Importing digital certificates and RSA private key files

To allow a client to communicate with other Brocade device using an SSL connection, you configure a set of digital certificates and RSA public-private key pairs on the device. A digital certificate is used for identifying the connecting client to the server. It contains information about the issuing Certificate Authority, as well as a public key. You can either import digital certificates and private keys from a server, or you can allow the Brocade device to create them.

If you want to allow the Brocade device to create the digital certificates, refer to the next section, [“Generating an SSL certificate”](#). If you choose to import an RSA certificate and private key file from a client, you can use TFTP to transfer the files.

For example, to import a digital certificate using TFTP, enter a command such as the following.

```
Brocade(config)#ip ssl certificate-data-file tftp 192.168.9.210 certfile
```

Syntax: [no] ip ssl certificate-data-file tftp <ip-addr> <certificate-filename>

NOTE

The digital certificate can be up to 4096 bits. Refer to [“Support for SSL digital certificates larger than 2048 bits”](#) on page 138.

To import an RSA private key from a client using TFTP, enter a command such as the following.

```
Brocade(config)#ip ssl private-key-file tftp 192.168.9.210 keyfile
```

Syntax: [no] ip ssl private-key-file tftp <ip-addr> <key-filename>

The <ip-addr> is the IP address of a TFTP server that contains the digital certificate or private key.

NOTE

The RSA key can be up to 4096 bits.

Generating an SSL certificate

After you have imported the digital certificate, it should automatically generate.

If the certificate does not automatically generate, enter the following command to generate it.

```
Brocade(config)#crypto-ssl certificate generate
```

Syntax: [no] crypto-ssl certificate generate

If you did not already import a digital certificate from a client, the device can create a default certificate. To do this, enter the following command.

```
Brocade(config)#crypto-ssl certificate generate default_cert
```

Syntax: [no] crypto-ssl certificate generate default_cert

Deleting the SSL certificate

To delete the SSL certificate, enter the following command.

```
Brocade(config)#crypto-ssl certificate zeroize
```

Syntax: [no] crypto-ssl certificate zeroize

TACACS and TACACS+ security

You can use the security protocol Terminal Access Controller Access Control System (TACACS) or TACACS+ to authenticate the following kinds of access to the Brocade device:

- Telnet access
- SSH access
- Console access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

The TACACS and TACACS+ protocols define how authentication, authorization, and accounting information is sent between a Brocade device and an authentication database on a TACACS/TACACS+ server. TACACS/TACACS+ services are maintained in a database, typically on a UNIX workstation or PC with a TACACS/TACACS+ server running.

How TACACS+ differs from TACACS

TACACS is a simple UDP-based access control protocol originally developed by BBN for MILNET. TACACS+ is an enhancement to TACACS and uses TCP to ensure reliable delivery.

TACACS+ is an enhancement to the TACACS security protocol. TACACS+ improves on TACACS by separating the functions of authentication, authorization, and accounting (AAA) and by encrypting all traffic between the Brocade device and the TACACS+ server. TACACS+ allows for arbitrary length and content authentication exchanges, which allow any authentication mechanism to be utilized with the Brocade device. TACACS+ is extensible to provide for site customization and future development features. The protocol allows the Brocade device to request very precise access control and allows the TACACS+ server to respond to each component of that request.

NOTE

TACACS+ provides for authentication, authorization, and accounting, but an implementation or configuration is not required to employ all three.

TACACS/TACACS+ authentication, authorization, and accounting

When you configure a Brocade device to use a TACACS/TACACS+ server for **authentication**, the device prompts users who are trying to access the CLI for a user name and password, then verifies the password with the TACACS/TACACS+ server.

If you are using TACACS+, Brocade recommends that you also configure **authorization**, in which the Brocade device consults a TACACS+ server to determine which management privilege level (and which associated set of commands) an authenticated user is allowed to use. You can also optionally configure **accounting**, which causes the Brocade device to log information on the TACACS+ server when specified events occur on the device.

NOTE

By default, a user logging into the device from Telnet or SSH would first enter the User EXEC level. The user can enter the **enable** command to get to the Privileged EXEC level.

A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. Refer to [“Entering privileged EXEC mode after a Telnet or SSH login”](#) on page 150.

Configuring TACACS/TACACS+ for devices in a Brocade IronStack

Because devices operating in a Brocade IronStack topology present multiple console ports, you must take additional steps to secure these ports when configuring TACACS/TACACS+.

The following is a sample AAA console configuration using TACACS+.

```
aaa authentication login default tacacs+ enable
aaa authentication login privilege-mode
aaa authorization commands 0 default tacacs+
aaa authorization exec default tacacs+
aaa accounting commands 0 default start-stop tacacs+
aaa accounting exec default start-stop tacacs+
aaa accounting system default start-stop tacacs+
enable aaa console
hostname Fred
ip address 144.10.6.56/255
tacacs-server host 255.253.255
tacacs-server key 1 $Gsig@U\
```

kill console

Syntax: kill console [all | unit]

- **all** - logs out all console port on stack units that are not the Active Controller
- **unit** - logs out the console port on a specified unit

Once AAA console is enabled, you should log out any open console ports on your IronStack using the **kill console** command:

```
Brocade(config)#kill console all
```

In case a user forgets to log out or a console is left unattended, you can also configure the console timeout (in minutes) on all stack units (including the Active Controller).

```
Brocade(config)#stack unit 3
Brocade(config-unit-3)#console timeout 5
Brocade(config-unit-3)#exit
Brocade(config)#stack unit 4
Brocade(config-unit-4)#console timeout 5
```

Use the **show who** and the **show telnet** commands to confirm the status of console sessions.

```

stack9#show who
Console connections (by unit number):
 1      established
      you are connecting to this session
      4 seconds in idle
 2      established
      1 hours 3 minutes 12 seconds in idle
 3      established
      1 hours 3 minutes 9 seconds in idle
 4      established
      1 hours 3 minutes 3 seconds in idle
Telnet connections (inbound):
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
Telnet connection (outbound):
 6      closed
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
stack9#

```

```

stack9#show telnet
Console connections (by unit number):
 1      established
      you are connecting to this session
      1 minutes 5 seconds in idle
 2      established
      1 hours 4 minutes 18 seconds in idle
 3      established
      1 hours 4 minutes 15 seconds in idle
 4      established
      1 hours 4 minutes 9 seconds in idle
Telnet connections (inbound):
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
Telnet connection (outbound):
 6      closed
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
stack9#

```

TACACS authentication

NOTE

Also, multiple challenges are supported for TACACS+ login authentication.

When TACACS authentication takes place, the following events occur.

1. A user attempts to gain access to the Brocade device by doing one of the following:
 - Logging into the device using Telnet, SSH, or the Web Management Interface
 - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The Brocade device sends a request containing the username and password to the TACACS server.
5. The username and password are validated in the TACACS server database.
6. If the password is valid, the user is authenticated.

TACACS+ authentication

When TACACS+ authentication takes place, the following events occur.

1. A user attempts to gain access to the Brocade device by doing one of the following:
 - Logging into the device using Telnet, SSH, or the Web Management Interface
 - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username.
3. The user enters a username.
4. The Brocade device obtains a password prompt from a TACACS+ server.
5. The user is prompted for a password.
6. The user enters a password.
7. The Brocade device sends the password to the TACACS+ server.
8. The password is validated in the TACACS+ server database.
9. If the password is valid, the user is authenticated.

TACACS+ authorization

Brocade devices support two kinds of TACACS+ authorization:

- Exec authorization determines a user privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

When TACACS+ exec authorization takes place, the following events occur.

1. A user logs into the Brocade device using Telnet, SSH, or the Web Management Interface
2. The user is authenticated.

3. The Brocade device consults the TACACS+ server to determine the privilege level of the user.
4. The TACACS+ server sends back a response containing an A-V (Attribute-Value) pair with the privilege level of the user.
5. The user is granted the specified privilege level.

When TACACS+ command authorization takes place, the following events occur.

1. A Telnet, SSH, or Web Management Interface user previously authenticated by a TACACS+ server enters a command on the Brocade device.
2. The Brocade device looks at its configuration to see if the command is at a privilege level that requires TACACS+ command authorization.
3. If the command belongs to a privilege level that requires authorization, the Brocade device consults the TACACS+ server to see if the user is authorized to use the command.
4. If the user is authorized to use the command, the command is executed.

TACACS+ accounting

TACACS+ accounting works as follows.

1. One of the following events occur on the Brocade device:
 - A user logs into the management interface using Telnet or SSH
 - A user enters a command for which accounting has been configured
 - A system event occurs, such as a reboot or reloading of the configuration file
2. The Brocade device checks the configuration to see if the event is one for which TACACS+ accounting is required.
3. If the event requires TACACS+ accounting, the Brocade device sends a TACACS+ Accounting Start packet to the TACACS+ accounting server, containing information about the event.
4. The TACACS+ accounting server acknowledges the Accounting Start packet.
5. The TACACS+ accounting server records information about the event.
6. When the event is concluded, the Brocade device sends an Accounting Stop packet to the TACACS+ accounting server.
7. The TACACS+ accounting server acknowledges the Accounting Stop packet.

AAA operations for TACACS/TACACS+

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a Brocade device that has TACACS/TACACS+ security configured.

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <method-list> <hr/> Exec authorization (TACACS+): aaa authorization exec default tacacs+ <hr/> System accounting start (TACACS+): aaa accounting system default start-stop <method-list>
User logs in using Telnet/SSH	Login authentication: aaa authentication login default <method-list> <hr/> Exec authorization (TACACS+): aaa authorization exec default tacacs+ <hr/> Exec accounting start (TACACS+): aaa accounting exec default <method-list> System accounting start (TACACS+): aaa accounting system default start-stop <method-list>
User logs into the Web Management Interface	Web authentication: aaa authentication web-server default <method-list> <hr/> Exec authorization (TACACS+): aaa authorization exec default tacacs+
User logs out of Telnet/SSH session	Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list> EXEC accounting stop (TACACS+): aaa accounting exec default start-stop <method-list>
User enters system commands (for example, reload , boot system)	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list> System accounting stop (TACACS+): aaa accounting system default start-stop <method-list>
User enters the command: [no] aaa accounting system default start-stop <method-list>	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list> System accounting start (TACACS+): aaa accounting system default start-stop <method-list>
User enters other commands	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list>

AAA security for commands pasted into the running-config

If AAA security is enabled on the device, commands pasted into the running-config are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running-config, and AAA command authorization or accounting, or both, are configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running-config. The server performing the AAA operations should be reachable when you paste the commands into the running-config file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

TACACS/TACACS+ configuration considerations

- You must deploy at least one TACACS/TACACS+ server in your network.
- Brocade devices support authentication using up to eight TACACS/TACACS+ servers. The device tries to use the servers in the order you add them to the device configuration.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select TACACS+ as the primary authentication method for Telnet CLI access, but you cannot also select RADIUS authentication as a primary method for the same type of access. However, you can configure backup authentication methods for each access type.
- You can configure the Brocade device to authenticate using a TACACS or TACACS+ server, not both.

Configuring TACACS

Follow the procedure given below for TACACS configurations.

1. Identify TACACS servers. Refer to [“Identifying the TACACS/TACACS+ servers”](#) on page 146.
2. Set optional parameters. Refer to [“Setting optional TACACS and TACACS+ parameters”](#) on page 147.
3. Configure authentication-method lists. Refer to [“Configuring authentication-method lists for TACACS and TACACS+”](#) on page 149.

Configuring TACACS+

Follow the procedure given below for TACACS+ configurations.

1. Identify TACACS+ servers. Refer to [“Identifying the TACACS/TACACS+ servers”](#) on page 146.
2. Set optional parameters. Refer to [“Setting optional TACACS and TACACS+ parameters”](#) on page 147.
3. Configure authentication-method lists. Refer to [“Configuring authentication-method lists for TACACS and TACACS+”](#) on page 149.
4. Optionally configure TACACS+ authorization. Refer to [“Configuring TACACS+ authorization”](#) on page 151.
5. Optionally configure TACACS+ accounting. Refer to [“TACACS+ accounting configuration”](#) on page 154.

Enabling TACACS

TACACS is disabled by default. To configure TACACS/TACACS+ authentication parameters, you must enable TACACS by entering the following command.

```
Brocade(config)#enable snmp config-tacacs
```

Syntax: `[no] enable snmp <config-radius | config-tacacs>`

The `<config-radius>` parameter specifies the RADIUS configuration mode. RADIUS is disabled by default.

The `<config-tacacs>` parameter specifies the TACACS configuration mode. TACACS is disabled by default.

Identifying the TACACS/TACACS+ servers

To use TACACS/TACACS+ servers to authenticate access to a Brocade device, you must identify the servers to the Brocade device.

For example, to identify three TACACS/TACACS+ servers, enter commands such as the following.

```
Brocade(config)#tacacs-server host 207.94.6.161
Brocade(config)#tacacs-server host 207.94.6.191
Brocade(config)#tacacs-server host 207.94.6.122
```

Syntax: `tacacs-server host <ip-addr> | <ipv6-addr> | <hostname> [auth-port <number>]`

The `<ip-addr> | <ipv6-addr> | <hostname>` parameter specifies the IP address or host name of the server. You can enter up to eight **tacacs-server host** commands to specify up to eight different servers.

NOTE

To specify the server's host name instead of its IP address, you must first identify a DNS server using the **ip dns server-address <ip-addr>** command at the global CONFIG level.

If you add multiple TACACS/TACACS+ authentication servers to the Brocade device, the device tries to reach them in the order you add them. For example, if you add three servers in the following order, the software tries the servers in the same order.

1. 207.94.6.161
2. 207.94.6.191
3. 207.94.6.122

You can remove a TACACS/TACACS+ server by entering **no** followed by the **tacacs-server** command. For example, to remove 207.94.6.161, enter the following command.

```
Brocade(config)#no tacacs-server host 207.94.6.161
```

NOTE

If you erase a **tacacs-server** command (by entering “no” followed by the command), make sure you also erase the **aaa** commands that specify TACACS/TACACS+ as an authentication method. (Refer to [“Configuring authentication-method lists for TACACS and TACACS+”](#) on page 149.) Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS/TACACS+ enabled and you will not be able to access the system.

The **auth-port** parameter specifies the UDP (for TACACS) or TCP (for TACACS+) port number of the authentication port on the server. The default port number is 49.

Specifying different servers for individual AAA functions

In a TACACS+ configuration, you can designate a server to handle a specific AAA task. For example, you can designate one TACACS+ server to handle authorization and another TACACS+ server to handle accounting. You can set the TACACS+ key for each server.

To specify different TACACS+ servers for authentication, authorization, and accounting, enter the command such as following.

```
Brocade(config)#tacacs-server host 1.2.3.4 auth-port 49 authentication-only key
abc
Brocade(config)#tacacs-server host 1.2.3.5 auth-port 49 authorization-only key
def
Brocade(config)#tacacs-server host 1.2.3.6 auth-port 49 accounting-only key ghi
```

Syntax: `tacacs-server host <ip-addr> | <ipv6-addr> | <server-name> [auth-port <num>] [authentication-only | authorization-only | accounting-only | default] [key 0 | 1 <string>]`

The default parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization and accounting. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until a server that can perform the requested function is found, or every server in the configured list has been tried.

Setting optional TACACS and TACACS+ parameters

You can set the following optional parameters in a TACACS and TACACS+ configuration:

- **TACACS+ key** – This parameter specifies the value that the Brocade device sends to the TACACS+ server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the Brocade device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Dead time** – This parameter specifies how long the Brocade device waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3 seconds.
- **Timeout** – This parameter specifies how many seconds the Brocade device waits for a response from a TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/TACACS+ servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the TACACS+ key

The **key** parameter in the **tacacs-server** command is used to encrypt TACACS+ packets before they are sent over the network. The value for the **key** parameter on the Brocade device should match the one configured on the TACACS+ server. The key can be from 1 – 32 characters in length and cannot include any space characters.

NOTE

The **tacacs-server key** command applies only to TACACS+ servers, not to TACACS servers. If you are configuring TACACS, do not configure a key on the TACACS server and do not enter a key on the Brocade device.

To specify a TACACS+ server key, enter a command such as following.

```
Brocade(config)#tacacs-server key rk Wong
```

Syntax: **tacacs-server key** [0 | 1] <string>

When you display the configuration of the Brocade device, the TACACS+ keys are encrypted. For example.

```
Brocade(config)#tacacs-server key 1 abc
Brocade(config)#write terminal
...
tacacs-server host 1.2.3.5 auth-port 49
tacacs key 1 $!2d
```

NOTE

Encryption of the TACACS+ keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the retransmission limit

The **retransmit** parameter specifies how many times the Brocade device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit limit can be from 1 – 5 times. The default is 3 times.

To set the TACACS and TACACS+ retransmit limit, enter a command such as the following.

```
Brocade(config)#tacacs-server retransmit 5
```

Syntax: **tacacs-server retransmit** <number>

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the Brocade device waits for a response from the TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/TACACS+ server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

```
Brocade(config)#tacacs-server timeout 5
```

Syntax: **tacacs-server timeout** <number>

Configuring authentication-method lists for TACACS and TACACS+

You can use TACACS/TACACS+ to authenticate Telnet/SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring TACACS/TACACS+ authentication, you create authentication-method lists specifically for these access methods, specifying TACACS/TACACS+ as the primary authentication method.

Within the authentication-method list, TACACS/TACACS+ is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If TACACS/TACACS+ authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for TACACS/TACACS+ authentication, you must create a separate authentication-method list for Telnet/SSH CLI access, and for access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication method list that specifies TACACS/TACACS+ as the primary authentication method for securing Telnet/SSH access to the CLI.

```
Brocade(config)#enable telnet authentication
Brocade(config)#aaa authentication login default tacacs local
```

The commands above cause TACACS/TACACS+ to be the primary authentication method for securing Telnet/SSH access to the CLI. If TACACS/TACACS+ authentication fails due to an error with the server, authentication is performed using local user accounts instead.

To create an authentication-method list that specifies TACACS/TACACS+ as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI.

```
Brocade(config)#aaa authentication enable default tacacs local none
```

The command above causes TACACS/TACACS+ to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If TACACS/TACACS+ authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

Syntax: [no] **aaa authentication enable | login default** <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]

The **web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE

If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web Management Interface, the browser sends an HTTP request for each frame. The Brocade device authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web Management Interface.

The <method1> parameter specifies the primary authentication method. The remaining optional <method> parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

TABLE 21 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the enable telnet password... command. Refer to “Setting a Telnet password” on page 124.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the enable super-user-password... command. Refer to “Setting passwords for management privilege levels” on page 125.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the username... command. Refer to “Local user account configuration” on page 133.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the tacacs-server command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the tacacs-server command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the radius-server command.
none	Do not use any authentication method. The device automatically permits access.

NOTE

For examples of how to define authentication-method lists for types of authentication other than TACACS/TACACS+, refer to [“Authentication-method lists”](#) on page 174.

Entering privileged EXEC mode after a Telnet or SSH login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command.

```
Brocade(config)#aaa authentication login privilege-mode
```

Syntax: `aaa authentication login privilege-mode`

The user privilege level is based on the privilege level granted during login.

Configuring enable authentication to prompt for password only

If Enable authentication is configured on the device, when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, by default he or she is prompted for a username and password. You can configure the Brocade device to prompt only for a password. The device uses the username entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the Brocade device to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI.

```
Brocade(config)#aaa authentication enable implicit-user
```

Syntax: `[no] aaa authentication enable implicit-user`

Telnet and SSH prompts when the TACACS+ Server is unavailable

When TACACS+ is the first method in the authentication method list, the device displays the login prompt received from the TACACS+ server. If a user attempts to login through Telnet or SSH, but none of the configured TACACS+ servers are available, the following takes place:

- If the next method in the authentication method list is "enable", the login prompt is skipped, and the user is prompted for the Enable password (that is, the password configured with the **enable super-user-password** command).
- If the next method in the authentication method list is "line", the login prompt is skipped, and the user is prompted for the Line password (that is, the password configured with the **enable telnet password** command).

Configuring TACACS+ authorization

Brocade devices support TACACS+ authorization for controlling access to management functions in the CLI. Two kinds of TACACS+ authorization are supported:

- Exec authorization determines a user privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

Configuring exec authorization

When TACACS+ exec authorization is performed, the Brocade device consults a TACACS+ server to determine the privilege level of the authenticated user. To configure TACACS+ exec authorization on the Brocade device, enter the following command.

```
Brocade(config)#aaa authorization exec default tacacs+
```

Syntax: **aaa authorization exec default tacacs+ | none**

If you specify **none**, or omit the **aaa authorization exec** command from the device configuration, no exec authorization is performed.

A user privilege level is obtained from the TACACS+ server in the "foundry-privlvl" A-V pair. If the **aaa authorization exec default tacacs** command exists in the configuration, the device assigns the user the privilege level specified by this A-V pair. If the command does not exist in the configuration, then the value in the "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.

NOTE

If the **aaa authorization exec default tacacs+** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the "foundry-privlvl" A-V pair received from the TACACS+ server. If the **aaa authorization exec default tacacs+** command does not exist in the configuration, then the value in the "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default tacacs+** command to work, either the **aaa authentication enable default tacacs+** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring an Attribute-Value pair on the TACACS+ server

During TACACS+ exec authorization, the Brocade device expects the TACACS+ server to send a response containing an A-V (Attribute-Value) pair that specifies the privilege level of the user. When the Brocade device receives the response, it extracts an A-V pair configured for the Exec service and uses it to determine the user privilege level.

To set a user privilege level, you can configure the “foundry-privlvl” A-V pair for the Exec service on the TACACS+ server.

Example

```
user=bob {
    default service = permit
    member admin
    #Global password
    global = cleartext "cat"
    service = exec {
        foundry-privlvl = 0
    }
}
```

In this example, the A-V pair `foundry-privlvl = 0` grants the user full read-write access. The value in the `foundry-privlvl` A-V pair is an integer that indicates the privilege level of the user. Possible values are 0 for super-user level, 4 for port-config level, or 5 for read-only level. If a value other than 0, 4, or 5 is specified in the `foundry-privlvl` A-V pair, the default privilege level of 5 (read-only) is used. The `foundry-privlvl` A-V pair can also be embedded in the group configuration for the user. See your TACACS+ documentation for the configuration syntax relevant to your server.

If the `foundry-privlvl` A-V pair is not present, the Brocade device extracts the last A-V pair configured for the Exec service that has a numeric value. The Brocade device uses this A-V pair to determine the user privilege level.

Example

```
user=bob {
    default service = permit
    member admin
    #Global password
    global = cleartext "cat"
    service = exec {
        privlvl = 15
    }
}
```

The attribute name in the A-V pair is not significant; the Brocade device uses the last one that has a numeric value. However, the Brocade device interprets the value for a non-“foundry-privlvl” A-V pair differently than it does for a “foundry-privlvl” A-V pair. The following table lists how the Brocade device associates a value from a non-“foundry-privlvl” A-V pair with a Brocade privilege level.

TABLE 22 Brocade equivalents for non-“foundry-privlvl” A-V pair values

Value for non-“foundry-privlvl” A-V pair	Brocade privilege level
15	0 (super-user)
From 14 – 1	4 (port-config)
Any other number or 0	5 (read-only)

In the example above, the A-V pair configured for the Exec service is `privlvl = 15`. The Brocade device uses the value in this A-V pair to set the user privilege level to 0 (super-user), granting the user full read-write access.

In a configuration that has both a “foundry-privlvl” A-V pair and a non-“foundry-privlvl” A-V pair for the Exec service, the non-“foundry-privlvl” A-V pair is ignored.

Example

```
user=bob {
    default service = permit
    member admin
    #Global password
    global = cleartext "cat"
    service = exec {
        foundry-privlvl = 4
        privlvl = 15
    }
}
```

In this example, the user would be granted a privilege level of 4 (port-config level). The `privlvl = 15` A-V pair is ignored by the Brocade device.

If the TACACS+ server has no A-V pair configured for the Exec service, the default privilege level of 5 (read-only) is used.

Configuring command authorization

When TACACS+ command authorization is enabled, the Brocade device consults a TACACS+ server to get authorization for commands entered by the user.

You enable TACACS+ command authorization by specifying a privilege level whose commands require authorization. For example, to configure the Brocade device to perform authorization for the commands available at the Super User privilege level (that is, all commands on the device), enter the following command.

```
Brocade(config)#aaa authorization commands 0 default tacacs+
```

Syntax: `aaa authorization commands <privilege-level> default tacacs+ | radius | none`

The `<privilege-level>` parameter can be one of the following:

- **0** – Authorization is performed for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE

TACACS+ command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web Management Interface.

TACACS+ command authorization is not performed for the following commands:

- At all levels: **exit**, **logout**, **end**, and **quit**.
- At the Privileged EXEC level: **enable** or **enable <text>**, where `<text>` is the password configured for the Super User privilege level.

If configured, command accounting is performed for these commands.

AAA support for console commands

NOTE

This feature is not supported on FastIron WS Series devices.

AAA support for commands entered at the console includes the following:

- Login prompt that uses AAA authentication, using authentication-method Lists
- Exec Authorization
- Exec Accounting
- Command authorization
- Command accounting
- System Accounting

To enable AAA support for commands entered at the console, enter the following command.

```
Brocade(config)#enable aaa console
```

Syntax: [no] enable aaa console

TACACS+ accounting configuration

Brocade devices support TACACS+ accounting for recording information about user activity and system events. When you configure TACACS+ accounting on a Brocade device, information is sent to a TACACS+ accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring TACACS+ accounting for Telnet/SSH (Shell) access

To send an Accounting Start packet to the TACACS+ accounting server when an authenticated user establishes a Telnet or SSH session on the Brocade device, and an Accounting Stop packet when the user logs out.

```
Brocade(config)#aaa accounting exec default start-stop tacacs+
```

Syntax: aaa accounting exec default start-stop radius | tacacs+ | none

Configuring TACACS+ accounting for CLI commands

You can configure TACACS+ accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the Brocade device to perform TACACS+ accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
Brocade(config)#aaa accounting commands 0 default start-stop tacacs+
```

An Accounting Start packet is sent to the TACACS+ accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE

If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: `aaa accounting commands <privilege-level> default start-stop radius | tacacs+ | none`

The `<privilege-level>` parameter can be one of the following:

- **0** – Records commands available at the Super User level (all commands)
- **4** – Records commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Records commands available at the Read Only level (read-only commands)

Configuring TACACS+ accounting for system events

You can configure TACACS+ accounting to record when system events occur on the Brocade device. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the TACACS+ accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed.

```
Brocade(config)#aaa accounting system default start-stop tacacs+
```

Syntax: `aaa accounting system default start-stop radius | tacacs+ | none`

Configuring an interface as the source for all TACACS and TACACS+ packets

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all TACACS/TACACS+ packets from the Layer 3 Switch. For configuration details, see [“Specifying a single source interface for specified packet types”](#) on page 977.

Displaying TACACS/TACACS+ statistics and configuration information

The **show aaa** command displays information about all TACACS+ and RADIUS servers identified on the device.

```
Brocade#show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                  opens=6 closes=3 timeouts=3 errors=0
                  packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 207.95.6.90 Auth Port=1645 Acct Port=1646:
                  opens=2 closes=1 timeouts=1 errors=0
                  packets in=1 packets out=4
no connection
```

The following table describes the TACACS/TACACS+ information displayed by the **show aaa** command.

TABLE 23 Output of the show aaa command for TACACS/TACACS+

Field	Description
Tacacs+ key	The setting configured with the tacacs-server key command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Tacacs+ retries	The setting configured with the tacacs-server retransmit command.
Tacacs+ timeout	The setting configured with the tacacs-server timeout command.
Tacacs+ dead-time	The setting configured with the tacacs-server dead-time command.
Tacacs+ Server	For each TACACS/TACACS+ server, the IP address, port, and the following statistics are displayed: <ul style="list-style-type: none"> • opens - Number of times the port was opened for communication with the server • closes - Number of times the port was closed normally • timeouts - Number of times port was closed due to a timeout • errors - Number of times an error occurred while opening the port • packets in - Number of packets received from the server • packets out - Number of packets sent to the server
connection	The current connection status. This can be "no connection" or "connection active".

The **show web connection** command displays the privilege level of Web Management Interface users.

Example

```
Brocade#show web-connection
```

We management Sessions:

User	Privilege	IP address	MAC address	Timeout(secs)	Connection
roy	READ-WRITE	10.1.1.3	0030.488.b84d9	279	HTTPS

Syntax: show web connection

Use the following command to clear web connections:

```
Brocade#clear web-connection
```

Syntax: clear web connection

After issuing the **clear web connection** command, the **show web connection** command displays the following output:

```
Brocade#show web-connection
```

```
No WEB-MANAGEMENT sessions are currently established!
```

RADIUS security

You can use a Remote Authentication Dial In User Service (RADIUS) server to secure the following types of access to the Brocade Layer 2 Switch or Layer 3 Switch:

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

RADIUS authentication, authorization, and accounting

When RADIUS **authentication** is implemented, the Brocade device consults a RADIUS server to verify user names and passwords. You can optionally configure RADIUS **authorization**, in which the Brocade device consults a list of commands supplied by the RADIUS server to determine whether a user can execute a command he or she has entered, as well as **accounting**, which causes the Brocade device to log information on a RADIUS accounting server when specified events occur on the device.

RADIUS authentication

When RADIUS authentication takes place, the following events occur.

1. A user attempts to gain access to the Brocade device by doing one of the following:
 - Logging into the device using Telnet, SSH, or the Web Management Interface
 - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.

3. The user enters a username and password.
4. The Brocade device sends a RADIUS Access-Request packet containing the username and password to the RADIUS server.
5. The RADIUS server validates the Brocade device using a shared secret (the RADIUS key).
6. The RADIUS server looks up the username in its database.
7. If the username is found in the database, the RADIUS server validates the password.
8. If the password is valid, the RADIUS server sends an Access-Accept packet to the Brocade device, authenticating the user. Within the Access-Accept packet are three Brocade vendor-specific attributes that indicate:
 - The privilege level of the user
 - A list of commands
 - Whether the user is allowed or denied usage of the commands in the list

The last two attributes are used with RADIUS authorization, if configured.

9. The user is authenticated, and the information supplied in the Access-Accept packet for the user is stored on the Brocade device. The user is granted the specified privilege level. If you configure RADIUS authorization, the user is allowed or denied usage of the commands in the list.

RADIUS authorization

When RADIUS authorization takes place, the following events occur.

1. A user previously authenticated by a RADIUS server enters a command on the Brocade device.
2. The Brocade device looks at its configuration to see if the command is at a privilege level that requires RADIUS command authorization.
3. If the command belongs to a privilege level that requires authorization, the Brocade device looks at the list of commands delivered to it in the RADIUS Access-Accept packet when the user was authenticated. (Along with the command list, an attribute was sent that specifies whether the user is permitted or denied usage of the commands in the list.)

NOTE

After RADIUS authentication takes place, the command list resides on the Brocade device. The RADIUS server is not consulted again once the user has been authenticated. This means that any changes made to the user command list on the RADIUS server are not reflected until the next time the user is authenticated by the RADIUS server, and the new command list is sent to the Brocade device.

4. If the command list indicates that the user is authorized to use the command, the command is executed.

RADIUS accounting

RADIUS accounting works as follows.

1. One of the following events occur on the Brocade device:
 - A user logs into the management interface using Telnet or SSH
 - A user enters a command for which accounting has been configured

- A system event occurs, such as a reboot or reloading of the configuration file
2. The Brocade device checks its configuration to see if the event is one for which RADIUS accounting is required.
 3. If the event requires RADIUS accounting, the Brocade device sends a RADIUS Accounting Start packet to the RADIUS accounting server, containing information about the event.
 4. The RADIUS accounting server acknowledges the Accounting Start packet.
 5. The RADIUS accounting server records information about the event.
 6. When the event is concluded, the Brocade device sends an Accounting Stop packet to the RADIUS accounting server.
 7. The RADIUS accounting server acknowledges the Accounting Stop packet.

AAA operations for RADIUS

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a Brocade device that has RADIUS security configured.

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <method-list>
	System accounting start: aaa accounting system default start-stop <method-list>
User logs in using Telnet/SSH	Login authentication: aaa authentication login default <method-list>
	EXEC accounting Start: aaa accounting exec default start-stop <method-list>
	System accounting Start: aaa accounting system default start-stop <method-list>
User logs into the Web Management Interface	Web authentication: aaa authentication web-server default <method-list>
User logs out of Telnet/SSH session	Command authorization for logout command: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list>
	EXEC accounting stop: aaa accounting exec default start-stop <method-list>
User enters system commands (for example, reload , boot system)	Command authorization: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list>
	System accounting stop: aaa accounting system default start-stop <method-list>

User action	Applicable AAA operations
User enters the command: [no] aaa accounting system default start-stop <method-list>	Command authorization: aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> System accounting start: aaa accounting system default start-stop <method-list>
User enters other commands	Command authorization: aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list>

AAA security for commands pasted Into the running-config

If AAA security is enabled on the device, commands pasted into the running-config are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running-config, and AAA command authorization or accounting, or both, are configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running-config. The server performing the AAA operations should be reachable when you paste the commands into the running-config file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

NOTE

Since RADIUS command authorization relies on a list of commands received from the RADIUS server when authentication is performed, it is important that you use RADIUS authentication when you also use RADIUS command authorization.

RADIUS configuration considerations

- You must deploy at least one RADIUS server in your network.
- Brocade devices support authentication using up to eight RADIUS servers, including those used for 802.1X authentication and for management. The device tries to use the servers in the order you add them to the device configuration. If one RADIUS server times out (does not respond), the Brocade device tries the next one in the list. Servers are tried in the same sequence each time there is a request.
- You can optionally configure a RADIUS server as a **port server**, indicating that the server will be used only to authenticate users on ports to which it is mapped, as opposed to globally authenticating users on all ports of the device. In earlier releases, all configured RADIUS servers are “global” servers and apply to users on all ports of the device. Refer to [“RADIUS server per port”](#) on page 164.
- You can map up to eight RADIUS servers to each port on the Brocade device. The port will authenticate users using only the RADIUS servers to which it is mapped. If there are no RADIUS servers mapped to a port, it will use the “global” servers for authentication. In earlier releases, all RADIUS servers are “global” servers and cannot be bound to individual ports. Refer to [“RADIUS server to individual ports mapping”](#) on page 165.

- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select RADIUS as the primary authentication method for Telnet CLI access, but you cannot also select TACACS+ authentication as the primary method for the same type of access. However, you can configure backup authentication methods for each access type.

Configuring RADIUS

Follow the procedure given below to configure a Brocade device for RADIUS.

1. Configure Brocade vendor-specific attributes on the RADIUS server. Refer to [“Brocade-specific attributes on the RADIUS server”](#) on page 161.
2. Identify the RADIUS server to the Brocade device. Refer to [“Identifying the RADIUS server to the Brocade device”](#) on page 163.
3. Optionally specify different servers for individual AAA functions. Refer to [“Specifying different servers for individual AAA functions”](#) on page 163.
4. Optionally configure the RADIUS server as a “port only” server. Refer to [“RADIUS server per port”](#) on page 164.
5. Optionally bind the RADIUS servers to ports on the Brocade device. Refer to [“RADIUS server to individual ports mapping”](#) on page 165.
6. Set RADIUS parameters. Refer to [“RADIUS parameters”](#) on page 166.
7. Configure authentication-method lists. Refer to [“Setting authentication-method lists for RADIUS”](#) on page 167.
8. Optionally configure RADIUS authorization. Refer to [“RADIUS authorization”](#) on page 169.
9. Optionally configure RADIUS accounting. [“RADIUS accounting”](#) on page 171.

Brocade-specific attributes on the RADIUS server

NOTE

For all Brocade devices, RADIUS Challenge is supported for 802.1x authentication but not for login authentication.

During the RADIUS authentication process, if a user supplies a valid username and password, the RADIUS server sends an Access-Accept packet to the Brocade device, authenticating the user. Within the Access-Accept packet are three Brocade vendor-specific attributes that indicate:

- The privilege level of the user
- A list of commands
- Whether the user is allowed or denied usage of the commands in the list

You must add these three Brocade vendor-specific attributes to your RADIUS server configuration, and configure the attributes in the individual or group profiles of the users that will access the Brocade device.

Brocade Vendor-ID is 1991, with Vendor-Type 1. The following table describes the Brocade vendor-specific attributes.

TABLE 24 Brocade vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Description
foundry-privilege-level	1	integer	<p>Specifies the privilege level for the user. This attribute can be set to one of the following:</p> <ul style="list-style-type: none"> • 0 - Super User level – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords. • 4 - Port Configuration level – Allows read-and-write access for specific ports but not for global (system-wide) parameters. • 5 - Read Only level – Allows access to the Privileged EXEC mode and User EXEC mode of the CLI but only with read access.
foundry-command-string	2	string	<p>Specifies a list of CLI commands that are permitted or denied to the user when RADIUS authorization is configured.</p> <p>The commands are delimited by semi-colons (;). You can specify an asterisk (*) as a wildcard at the end of a command string.</p> <p>For example, the following command list specifies all show and debug ip commands, as well as the write terminal command:</p> <p>show *; debug ip *; write term*</p>
foundry-command-exception-flag	3	integer	<p>Specifies whether the commands indicated by the foundry-command-string attribute are permitted or denied to the user. This attribute can be set to one of the following:</p> <ul style="list-style-type: none"> • 0 - Permit execution of the commands indicated by foundry-command-string, deny all other commands. • 1 - Deny execution of the commands indicated by foundry-command-string, permit all other commands.
foundry-access-list	5	string	<p>Specifies the access control list to be used for RADIUS authorization. Enter the access control list in the following format.</p> <p>type=string, value="ipacl.[e s].[in out] = [<acl-name> <acl-number>] <separator> macfilter.in = [<acl-name> <acl-number>]</p> <p>Where:</p> <ul style="list-style-type: none"> • separator can be a space, newline, semicolon, comma, or null character • ipacl.e is an extended ACL; ipacl.s is a standard ACL.
foundry-MAC-authent-needs-802x	6	integer	<p>Specifies whether or not 802.1x authentication is required and enabled.</p> <p>0 - Disabled</p> <p>1 - Enabled</p>

TABLE 24 Brocade vendor-specific attributes for RADIUS (Continued)

Attribute name	Attribute ID	Data type	Description
foundry-802.1x-valid-lookup	7	integer	Specifies if 802.1x lookup is enabled: 0 - Disabled 1 - Enabled
foundry-MAC-based-VLAN-QOS	8	integer	Specifies the priority for MAC-based VLAN QOS: 0 - qos_priority_0 1 - qos_priority_1 2 - qos_priority_2 3 - qos_priority_3 4 - qos_priority_4 5 - qos_priority_5 6 - qos_priority_6 7 - qos_priority_7

Enabling SNMP to configure RADIUS

To enable SNMP access to RADIUS MIB objects on the device, enter a command such as the following.

```
Brocade(config)#enable snmp config-radius
```

Syntax: [no] enable snmp <config-radius | config-tacacs>

The <config-radius> parameter specifies the RADIUS configuration mode. RADIUS is disabled by default.

The <config-tacacs> parameter specifies the TACACS configuration mode. TACACS is disabled by default.

Identifying the RADIUS server to the Brocade device

To use a RADIUS server to authenticate access to a Brocade device, you must identify the server to the Brocade device.

Example

```
Brocade(config)#radius-server host 209.157.22.99
```

Syntax: radius-server host <ip-addr> | <ipv6-addr> | <server-name> [auth-port <number>] [acct-port <number>]

The **host** <ip-addr> | <ipv6-addr> | <server-name> parameter is either an IP address or an ASCII text string.

The <auth-port> parameter is the Authentication port number. The default is 1645.

The <acct-port> parameter is the Accounting port number. The default is 1646.

Specifying different servers for individual AAA functions

In a RADIUS configuration, you can designate a server to handle a specific AAA task. For example, you can designate one RADIUS server to handle authorization and another RADIUS server to handle accounting. You can specify individual servers for authentication and accounting, but not for authorization. You can set the RADIUS key for each server.

To specify different RADIUS servers for authentication, authorization, and accounting, enter commands such as the following.

```
Brocade(config)#radius-server host 1.2.3.4 authentication-only key abc
Brocade(config)#radius-server host 1.2.3.5 authorization-only key def
Brocade(config)#radius-server host 1.2.3.6 accounting-only key ghi
```

Syntax: `radius-server host <ip-addr> | <ipv6-addr> | <server-name> [auth-port <number>] [acct-port <number>] [authentication-only | accounting-only | default] [key 0 | 1 <string>]`

The **default** parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization and accounting. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until a server that can perform the requested function is found, or every server in the configured list has been tried.

RADIUS server per port

You can optionally configure a RADIUS server per port, indicating that it will be used only to authenticate users on ports to which it is mapped. A RADIUS server that is not explicitly configured as a RADIUS server per port is a **global server**, and can be used to authenticate users on ports to which no RADIUS servers are mapped.

RADIUS server per port configuration notes

- This feature works with 802.1X and multi-device port authentication only.
- You can define up to eight RADIUS servers per Brocade device.

RADIUS configuration example and command syntax

The following shows an example configuration.

```
Brocade(config)#radius-server host 10.10.10.103 auth-port 1812 acct-port 1813
default key mykeyword dot1x port-only
Brocade(config)#radius-server host 10.10.10.104 auth-port 1812 acct-port 1813
default key mykeyword dot1x port-only
Brocade(config)#radius-server host 10.10.10.105 auth-port 1812 acct-port 1813
default key mykeyword dot1x
Brocade(config)#radius-server host 10.10.10.106 auth-port 1812 acct-port 1813
default key mykeyword dot1x
```

The above configuration has the following affect:

- RADIUS servers 10.10.10.103 and 10.10.10.104 will be used only to authenticate users on ports to which the servers are mapped. To map a RADIUS server to a port, refer to [“RADIUS server to individual ports mapping”](#) on page 165.
- RADIUS servers 10.10.10.105 and 10.10.10.106 will be used to authenticate users on ports to which no RADIUS servers are mapped. For example, port e 9, to which no RADIUS servers are mapped, will send a RADIUS request to the first configured RADIUS server, 10.10.10.105. If the request fails, it will go to the second configured RADIUS server, 10.10.10.106. It will not send requests to 10.10.10.103 or 10.10.10.104, since these servers are configured as port servers.

Syntax: `radius-server host <ip-addr> | <server-name> [auth-port <number>] [acct-port <number>] [default key <string> dot1x] [port-only]`

The **host** `<ip-addr>` is the IPv4 address.

The **auth-port** `<number>` parameter is the Authentication port number; it is an optional parameter. The default is 1645.

The **acct-port** `<number>` parameter is the Accounting port number; it is an optional parameter. The default is 1646.

The **default key** `<string> dot1x` parameter indicates that this RADIUS server supports the 802.1X standard. A RADIUS server that supports the 802.1X standard can also be used to authenticate non-802.1X authentication requests.

The **port-only** parameter is optional and specifies that the server will be used only to authenticate users on ports to which it is mapped.

RADIUS server to individual ports mapping

You can map up to eight RADIUS servers to each port on the Brocade device. The port will authenticate users using only the RADIUS servers to which the port is mapped. If there are no RADIUS servers mapped to a port, it will use the “global” servers for authentication.

As in previous releases, a port goes through the list of servers in the order in which it was mapped or configured, until a server that can perform the requested function is found, or until every server in the list has been tried.

RADIUS server-to-ports configuration notes

- This feature works with 802.1X and multic-device port authentication only.
- You can map a RADIUS server to a physical port only. You cannot map a RADIUS server to a VE.

RADIUS server-to-ports configuration example and command syntax

To map a RADIUS server to a port, enter commands such as the following.

```
Brocade(config)#int e 3
Brocade(config-if-e1000-3)#dot1x port-control auto
Brocade(config-if-e1000-3)#use-radius-server 10.10.10.103
Brocade(config-if-e1000-3)#use-radius-server 10.10.10.110
```

With the above configuration, port e 3 would send a RADIUS request to 10.10.10.103 first, since it is the first server mapped to the port. If it fails, it will go to 10.10.10.110.

Syntax: `use-radius-server <ip-addr>`

The **host** `<ip-addr>` is an IPv4 address.

RADIUS parameters

You can set the following parameters in a RADIUS configuration:

- **RADIUS key** – This parameter specifies the value that the Brocade device sends to the RADIUS server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the Brocade device will resend an authentication request when the RADIUS server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Timeout** – This parameter specifies how many seconds the Brocade device waits for a response from a RADIUS server before either retrying the authentication request, or determining that the RADIUS servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the RADIUS key

The **key** parameter in the **radius-server** command is used to encrypt RADIUS packets before they are sent over the network. The value for the **key** parameter on the Brocade device should match the one configured on the RADIUS server. The key can be from 1 – 32 characters in length and cannot include any space characters.

To specify a RADIUS server key, enter a command such as the following.

```
Brocade(config)#radius-server key mirabeau
```

Syntax: **radius-server key** [0 | 1] <string>

When you display the configuration of the Brocade device, the RADIUS key is encrypted.

Example

```
Brocade(config)#radius-server key 1 abc
Brocade(config)#write terminal
...
radius-server host 1.2.3.5
radius key 1 $!2d
```

NOTE

Encryption of the RADIUS keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the retransmission limit

The **retransmit** parameter specifies the maximum number of retransmission attempts. When an authentication request times out, the Brocade software will retransmit the request up to the maximum number of retransmissions configured. The default retransmit value is 3 retries. The range of retransmit values is from 1 – 5.

To set the RADIUS retransmit limit, enter a command such as the following.

```
Brocade(config)#radius-server retransmit 5
```

Syntax: **radius-server retransmit** <number>

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the Brocade device waits for a response from the RADIUS server before either retrying the authentication request, or determining that the RADIUS server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

```
Brocade(config)#radius-server timeout 5
```

Syntax: **radius-server timeout** *<number>*

Setting RADIUS over IPv6

Brocade devices support the ability to send RADIUS packets over an IPv6 network.

To enable the Brocade device to send RADIUS packets over IPv6, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#radius-server host ipv6 3000::300
```

Syntax: **radius-server host ipv6** *<ipv6-host address>*

The *<ipv6-host address>* is the IPv6 address of the RADIUS server. When you enter the IPv6 host address, you do not need to specify the prefix length. A prefix length of 128 is implied.

Setting authentication-method lists for RADIUS

You can use RADIUS to authenticate Telnet/SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring RADIUS authentication, you create authentication-method lists specifically for these access methods, specifying RADIUS as the primary authentication method.

Within the authentication-method list, RADIUS is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If RADIUS authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for RADIUS, you must create a separate authentication-method list for Telnet or SSH CLI access and for CLI access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing Telnet access to the CLI.

```
Brocade(config)#enable telnet authentication
Brocade(config)#aaa authentication login default radius local
```

The commands above cause RADIUS to be the primary authentication method for securing Telnet access to the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI.

```
Brocade(config)#aaa authentication enable default radius local none
```

The command above causes RADIUS to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

Syntax: [no] **aaa authentication enable | login default** <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]

The **web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE

If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web Management Interface, the browser sends an HTTP request for each frame. The Brocade device authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web Management Interface.

The <method1> parameter specifies the primary authentication method. The remaining optional <method> parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

TABLE 25 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the enable telnet password... command. Refer to “Setting a Telnet password” on page 124.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the enable super-user-password... command. Refer to “Setting passwords for management privilege levels” on page 125.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the username... command. Refer to “Local user account configuration” on page 133.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the tacacs-server command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the tacacs-server command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the radius-server command.
none	Do not use any authentication method. The device automatically permits access.

NOTE

For examples of how to define authentication-method lists for types of authentication other than RADIUS, refer to [“Authentication-method lists”](#) on page 174.

Entering privileged EXEC mode after a Telnet or SSH login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command.

```
Brocade(config)#aaa authentication login privilege-mode
```

Syntax: `aaa authentication login privilege-mode`

The user privilege level is based on the privilege level granted during login.

Configuring enable authentication to prompt for password only

If Enable authentication is configured on the device, when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, by default he or she is prompted for a username and password. You can configure the Brocade device to prompt only for a password. The device uses the username entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the Brocade device to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI.

```
Brocade(config)#aaa authentication enable implicit-user
```

Syntax: `[no] aaa authentication enable implicit-user`

RADIUS authorization

Brocade devices support RADIUS authorization for controlling access to management functions in the CLI. Two kinds of RADIUS authorization are supported:

- Exec authorization determines a user privilege level when they are authenticated
- Command authorization consults a RADIUS server to get authorization for commands entered by the user

Configuring exec authorization

When RADIUS exec authorization is performed, the Brocade device consults a RADIUS server to determine the privilege level of the authenticated user. To configure RADIUS exec authorization on the Brocade device, enter the following command.

```
Brocade(config)#aaa authorization exec default radius
```

Syntax: `aaa authorization exec default radius | none`

If you specify **none**, or omit the **aaa authorization exec** command from the device configuration, no exec authorization is performed.

NOTE

If the **aaa authorization exec default radius** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the foundry-privilege-level attribute received from the RADIUS server. If the **aaa authorization exec default radius** command does not exist in the configuration, then the value in the foundry-privilege-level attribute is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default radius** command to work, either the **aaa authentication enable default radius** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring command authorization

When RADIUS command authorization is enabled, the Brocade device consults the list of commands supplied by the RADIUS server during authentication to determine whether a user can execute a command he or she has entered.

You enable RADIUS command authorization by specifying a privilege level whose commands require authorization. For example, to configure the Brocade device to perform authorization for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
Brocade(config)#aaa authorization commands 0 default radius
```

Syntax: **aaa authorization commands** <privilege-level> **default radius** | **tacacs+** | **none**

The <privilege-level> parameter can be one of the following:

- **0** – Authorization is performed (that is, the Brocade device looks at the command list) for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE

RADIUS command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web Management Interface.

NOTE

Since RADIUS command authorization relies on the command list supplied by the RADIUS server during authentication, you cannot perform RADIUS authorization without RADIUS authentication.

Command authorization and accounting for console commands

The Brocade device supports command authorization and command accounting for CLI commands entered at the console. To configure the device to perform command authorization and command accounting for console commands, enter the following.

```
Brocade(config)#enable aaa console
```

Syntax: **enable aaa console**



CAUTION

If you have previously configured the device to perform command authorization using a RADIUS server, entering the **enable aaa console** command may prevent the execution of any subsequent commands entered on the console.

This happens because RADIUS command authorization requires a list of allowable commands from the RADIUS server. This list is obtained during RADIUS authentication. For console sessions, RADIUS authentication is performed only if you have configured Enable authentication and specified RADIUS as the authentication method (for example, with the `aaa authentication enable default radius` command). If RADIUS authentication is never performed, the list of allowable commands is never obtained from the RADIUS server. Consequently, there would be no allowable commands on the console.

RADIUS accounting

Brocade devices support RADIUS accounting for recording information about user activity and system events. When you configure RADIUS accounting on a Brocade device, information is sent to a RADIUS accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring RADIUS accounting for Telnet/SSH (Shell) access

To send an Accounting Start packet to the RADIUS accounting server when an authenticated user establishes a Telnet or SSH session on the Brocade device, and an Accounting Stop packet when the user logs out.

```
Brocade(config)#aaa accounting exec default start-stop radius
```

Syntax: `aaa accounting exec default start-stop radius | tacacs+ | none`

Configuring RADIUS accounting for CLI commands

You can configure RADIUS accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the Brocade device to perform RADIUS accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
Brocade(config)#aaa accounting commands 0 default start-stop radius
```

An Accounting Start packet is sent to the RADIUS accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE

If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: `aaa accounting commands <privilege-level> default start-stop radius | tacacs | none`

The `<privilege-level>` parameter can be one of the following:

- **0** – Records commands available at the Super User level (all commands)
- **4** – Records commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Records commands available at the Read Only level (read-only commands)

Configuring RADIUS accounting for system events

You can configure RADIUS accounting to record when system events occur on the Brocade device. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the RADIUS accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed.

```
Brocade(config)#aaa accounting system default start-stop radius
```

Syntax: `aaa accounting system default start-stop radius | tacacs+ | none`

Configuring an interface as the source for all RADIUS packets

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all RADIUS packets from the Layer 3 Switch. For configuration details, see [“Specifying a single source interface for specified packet types”](#) on page 977.

Displaying RADIUS configuration information

The **show aaa** command displays information about all TACACS/TACACS+ and RADIUS servers identified on the device.

Example

```
Brocade#show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

The following table describes the RADIUS information displayed by the **show aaa** command.

TABLE 26 Output of the show aaa command for RADIUS

Field	Description
Radius key	The setting configured with the radius-server key command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Radius retries	The setting configured with the radius-server retransmit command.
Radius timeout	The setting configured with the radius-server timeout command.
Radius dead-time	The setting configured with the radius-server dead-time command.
Radius Server	For each RADIUS server, the IP address, and the following statistics are displayed: Auth PortRADIUS authentication port number (default 1645) Acct PortRADIUS accounting port number (default 1646) <ul style="list-style-type: none"> • opens - Number of times the port was opened for communication with the server • closes - Number of times the port was closed normally • timeouts - Number of times port was closed due to a timeout • errors - Number of times an error occurred while opening the port • packets in - Number of packets received from the server • packets out - Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

The **show web connection** command displays the privilege level of Web Management Interface users.

Example

```
Brocade#show web-connection
```

```
We management Sessions:
```

User	Privilege	IP address	MAC address	Timeout (secs)	Connection
roy	READ-WRITE	10.1.1.3	0030.488.b84d9	279	HTTPS

Syntax: show web connection

Use the following command to clear web connections:

```
FastIron#clear web-connection
```

Syntax: clear web connection

After issuing the **clear web connection** command, the **show web connection** command displays the following output:

```
Brocade#show web-connection
```

```
No WEB-MANAGEMENT sessions are currently established!
```

Authentication-method lists

To implement one or more authentication methods for securing access to the device, you configure authentication-method lists that set the order in which the authentication methods are consulted.

In an authentication-method list, you specify the access method (Telnet, Web, SNMP, and so on) and the order in which the device tries one or more of the following authentication methods:

- Local Telnet login password
- Local password for the Super User privilege level
- Local user accounts configured on the device
- Database on a TACACS or TACACS+ server
- Database on a RADIUS server
- No authentication

NOTE

The TACACS/TACACS+, RADIUS, and Telnet login password authentication methods are not supported for SNMP access.

NOTE

To authenticate Telnet access to the CLI, you also must enable the authentication by entering the **enable telnet authentication** command at the global CONFIG level of the CLI. You cannot enable Telnet authentication using the Web Management Interface.

NOTE

You do not need an authentication-method list to secure access based on ACLs or a list of IP addresses. Refer to [“ACL usage to restrict remote access”](#) on page 112 or [“Remote access restrictions”](#) on page 115.

In an authentication-method list for a particular access method, you can specify up to seven authentication methods. If the first authentication method is successful, the software grants access and stops the authentication process. If the access is rejected by the first authentication method, the software denies access and stops checking.

However, if an error occurs with an authentication method, the software tries the next method on the list, and so on. For example, if the first authentication method is the RADIUS server, but the link to the server is down, the software will try the next authentication method in the list.

NOTE

If an authentication method is working properly and the password (and user name, if applicable) is not known to that method, this is not an error. The authentication attempt stops, and the user is denied access.

The software will continue this process until either the authentication method is passed or the software reaches the end of the method list. If the Super User level password is not rejected after all the access methods in the list have been tried, access is granted.

Configuration considerations for authentication-method lists

- For CLI access, you must configure authentication-method lists if you want the device to authenticate access using local user accounts or a RADIUS server. Otherwise, the device will authenticate using only the locally based password for the Super User privilege level.
- When no authentication-method list is configured specifically for Web management access, the device performs authentication using the SNMP community strings:
 - For read-only access, you can use the user name “get” and the password “public”. The default read-only community string is “public”.
 - There is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web Management Interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure as the password. Refer to [“TACACS and TACACS+ security”](#) on page 139.
- If you configure an authentication-method list for Web management access and specify “local” as the primary authentication method, users who attempt to access the device using the Web Management Interface must supply a user name and password configured in one of the local user accounts on the device. The user **cannot** access the device by entering “set” or “get” and the corresponding SNMP community string.

Examples of authentication-method lists

The following examples show how to configure authentication-method lists. In these examples, the primary authentication method for each is “local”. The device will authenticate access attempts using the locally configured usernames and passwords.

The command syntax for each of the following examples is provided in [“Command Syntax”](#) on page 176.

Example 1

To configure an authentication-method list for the Web Management Interface, enter a command such as the following.

```
Brocade(config)#aaa authentication web-server default local
```

This command configures the device to use the local user accounts to authenticate access to the device through the Web Management Interface. If the device does not have a user account that matches the user name and password entered by the user, the user is not granted access.

Example 2

To configure an authentication-method list for SNMP, enter a command such as the following.

```
Brocade(config)#aaa authentication snmp-server default local
```

This command allows certain incoming SNMP SET operations to be authenticated using the locally configured usernames and passwords. When this command is enabled, community string validation is not performed for incoming SNMP V1 and V2c packets. This command takes effect as long as the first varbind for SNMP packets is set to one of the following:

- snAgGblPassword=”<username> <password>” (for AAA method local)

- `snAgGblPassword="<password>"` (for AAA method line, enable)

NOTE

Certain SNMP objects need additional validation. These objects include but are not limited to: **snAgReload**, **snAgWriteNVRAM**, **snAgConfigFromNVRAM**, **snAgImgLoad**, **snAgCfgLoad** and **snAgGblTelnetPassword**. For more information, see **snAgGblPassword** in the *IronWare MIB Reference Guide*.

If AAA is set up to check both the username and password, the string contains the username, followed by a space then the password. If AAA is set up to authenticate with the current Enable or Line password, the string contains the password only.

Note that the above configuration can be overridden by the command **no snmp-server pw-check**, which disables password checking for SNMP SET requests.

Example 3

To configure an authentication-method list for the Privileged EXEC and CONFIG levels of the CLI, enter the following command.

```
Brocade(config)#aaa authentication enable default local
```

This command configures the device to use the local user accounts to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI.

Example 4

To configure the device to consult a RADIUS server first to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI, then consult the local user accounts if the RADIUS server is unavailable, enter the following command.

```
Brocade(config)#aaa authentication enable default radius local
```

Command Syntax

The following is the command syntax for the preceding examples.

Syntax: `[no] aaa authentication snmp-server | web-server | enable | login default <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]`

The **snmp-server | web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE

TACACS/TACACS+ and RADIUS are supported only with the **enable** and **login** parameters.

The `<method1>` parameter specifies the primary authentication method. The remaining optional `<method>` parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

TABLE 27 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the enable telnet password... command. Refer to “Setting a Telnet password” on page 124.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the enable super-user-password... command. Refer to “Setting passwords for management privilege levels” on page 125.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the username... command. Refer to “Local user account configuration” on page 133.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the tacacs-server command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the tacacs-server command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the radius-server command. Refer to “RADIUS security” on page 157.
none	Do not use any authentication method. The device automatically permits access.

TCP Flags - edge port security

NOTE

This feature is not supported on FastIron X Series devices.

The edge port security feature works in combination with IP ACL rules, and supports all 6 TCP flags present in the offset 13 of the TCP header:

- +|- urg = Urgent
- +|- ack = Acknowledge
- +|- psh = Push
- +|- rst = Reset
- +|- syn = Synchronize
- +|- fin = Finish

TCP flags can be combined with other ACL functions (such as dscp-marking and traffic policies), giving you greater flexibility when designing ACLs.

The TCP flags feature offers two options, match-all and match-any:

- **Match-any** - Indicates that incoming TCP traffic must be matched against any of the TCP flags configured as part of the match-any ACL rule. In CAM hardware, the number of ACL rules will match the number of configured flags.
- **Match-all** - Indicates that incoming TCP traffic must be matched against all of the TCP flags configured as part of the match-all ACL rule. In CAM hardware, there will be only one ACL rule for all configured flags.

Example

```
Brocade(config-ext-nACL)#permit tcp 1.1.1.1 0.0.0.255 eq 100 2.2.2.2 0.0.0.255  
eq 300 match-all +urg +ack +syn -rst
```

This command configures a single rule in CAM hardware. This rule will contain all of the configured TCP flags (urg, ack, syn, and rst).

Using TCP Flags in combination with other ACL features

The TCP Flags feature has the added capability of being combined with other ACL features.

Example

```
Brocade(config-ext-nACL)#permit tcp any any match-all +urg +ack +syn -rst  
traffic-policy test
```

This command configures the ACL to match incoming traffic with the TCP Flags urg, ack, and syn and also to apply the traffic policy (rate, limit, etc.) to the matched traffic.

```
Brocade(config-ext-nACL)#permit tcp any any match-all +urg +ack +syn -rst tos  
normal
```

This command configures the ACL to match incoming traffic with the flags urg, ack, and syn, and also sets the tos bit to normal when the traffic exits the device.

NOTE

TCP Flags combines the functionality of older features such as TCP Syn Attack and TCP Establish. Avoid configuring these older features on a port where you have configured TCP Flags. TCP Flags can perform all of the functions of TCP Syn Attack and TCP Establish, and more. However, if TCP Syn Attack is configured on a port along with TCP Flags, TCP Syn Attack will take precedence.

NOTE

If an ACL clause with match-any exists, and the system runs out of CAM, if the total number of TCP rules to TCP Flags will not fit within 1021 entries (the maximum rules allowed per device), then none of the TCP Flag rules will be programmed into the CAM hardware.

NOTE

If a range option and match-any TCP-flags are combined in the same ACL, the total number of rules will be calculated as: Total number of rules in CAM hardware = (number of rules for range)* (number of rules for match-any TCP-flags).

SSH2 and SCP

Table 28 lists individual Brocade switches and the SSH2 and Secure Copy features they support.

TABLE 28 Supported SSH2 and Secure Copy features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Secure Shell (SSH) version 2	Yes	Yes	Yes	Yes	Yes
AES encryption for SSH2	Yes	Yes	Yes	Yes	Yes
Optional parameters for SSH2	Yes	Yes	Yes	Yes	Yes
Using secure copy (SCP) with SSH2	Yes	Yes	Yes	Yes	Yes
Filtering SSH access using ACLs	Yes	Yes	Yes	Yes	Yes
Terminating an active SSH connection	Yes	Yes	Yes	Yes	Yes
SSH client	Yes	Yes	Yes	Yes	Yes

SSH version 2 overview

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a Brocade device. SSH provides a function similar to Telnet. Users can log into and configure the device using a publicly or commercially available SSH client program, just as they can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the device.

The Brocade SSH2 implementation is compatible with all versions of the SSH2 protocol (2.1, 2.2, and so on). At the beginning of an SSH session, the Brocade device negotiates the version of SSH2 to be used. The highest version of SSH2 supported by both the Brocade device and the client is the version that is used for the session. Once the SSH2 version is negotiated, the encryption algorithm with the highest security ranking is selected to be used for the session.

Brocade devices also support Secure Copy (SCP) for securely transferring files between a Brocade device and SCP-enabled remote hosts.

NOTE

The SSH feature includes software that is copyright Allegro Software Development Corporation.

SSH2 is supported in the Layer 2 and Layer 3 codes.

SSH2 is a substantial revision of Secure Shell, comprising the following hybrid protocols and definitions:

- SSH Transport Layer Protocol
- SSH Authentication Protocol
- SSH Connection Protocol

- SECSH Public Key File Format
- SSH Fingerprint Format
- SSH Protocol Assigned Numbers
- SSH Transport Layer Encryption Modes
- SCP/SSH URI Format

Tested SSH2 clients

The following SSH clients have been tested with SSH2:

- SSH Secure Shell 3.2.3
- Van Dyke SecureCRT 5.2.2
- F-Secure SSH Client 5.3 and 6.0
- PuTTY 0.60
- OpenSSH 4.3p2
- Brocade FastIron SSH Client

NOTE

Supported SSH client public key sizes are 1024 bits for DSA keys, and 1024 or 2048 bits for RSA keys.

SSH2 supported features

SSH2 (Secure Shell version 2 protocol) provides an SSH server and an SSH client. The SSH server allows secure remote access management functions on a Brocade device. SSH provides a function that is similar to Telnet, but unlike Telnet, SSH provides a secure, encrypted connection.

Brocade SSH2 support includes the following:

- Key exchange methods are **diffie-hellman-group1-sha1**
- The supported public key algorithms are **ssh-dss** and **ssh-rsa**.
- Encryption is provided with **3des-cbc**, **aes128-cbc**, **aes192-cbc** or **aes256-cbc**. AES encryption has been adopted by the U.S. Government as an encryption standard.
- Data integrity is ensured with **hmac-sha1**.
- Supported authentication methods are **Password** and **publickey**.
- Five inbound SSH connection at one time are supported.
- One outbound SSH is supported.

SSH2 unsupported features

The following are not supported with SSH2:

- Compression
- TCP/IP port forwarding, X11 forwarding, and secure file transfer
- SSH version 1

SSH2 authentication types

The Brocade implementation of SSH2 supports the following types of user authentication:

- **DSA challenge-response authentication**, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.
- **RSA challenge-response authentication**, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.
- **Password authentication**, where users attempting to gain access to the device using an SSH client are authenticated with passwords stored on the device or on a TACACS or TACACS+ server or a RADIUS server.

Configuring SSH2

You can configure the device to use any combination of these authentication types. The SSH server and client negotiate which type to use.

To configure SSH2, follow these steps:

1. Generate a host Digital Signature Algorithm (DSA) or Really Secure Algorithm (RSA) public and private key pair for the device.

See the section [“Enabling and disabling SSH by generating and deleting host keys”](#) on page 181.

2. Configure DSA or RSA challenge-response authentication.

See the section [“Configuring DSA or RSA challenge-response authentication”](#) on page 183.

3. Set optional parameters.

See the section [“Optional SSH parameters”](#) on page 185.

Enabling and disabling SSH by generating and deleting host keys

To enable SSH, you generate a public and private DSA or RSA host key pair on the device. The SSH server on the Brocade device uses this host DSA or RSA key pair, along with a dynamically generated server DSA or RSA key pair, to negotiate a session key and encryption method with the client trying to connect to it.

While the SSH listener exists at all times, sessions can not be started from clients until a host key is generated. After a host key is generated, clients can start sessions.

To disable SSH, you delete all of the host keys from the device.

When a host key pair is generated, it is saved to the flash memory of all management modules. When a host key pair is deleted, it is deleted from the flash memory of all management modules.

The time to initially generate SSH keys varies depending on the configuration, and can be from a under a minute to several minutes.

NOTE

If you have generated SSH keys on the switch, you should delete and regenerate it when you upgrade or downgrade the software version before ssh session.

Setting the CPU priority for key generation (ICX 6430 and ICX 6450 only)

Generating the key is a resource-intensive operation. You can set the priority for this operation to high so that the device allocates more CPU time for this operation. So you must use this option only when the device is in the maintenance window. This option reduces the time taken for key generation.

To set high priority for the key generation operation, enter the following command:

```
Brocade(config)#crypto-gen priority high
```

Syntax: `crypto key crypto-gen priority default | high`

The **default** keyword sets the priority as default. The key generation task is handled with the regular priority.

The **high** keyword sets the high priority for the key generation task. Use this option only when the device is in the maintenance window.

Generating and deleting a DSA key pair

To generate a DSA key pair, enter the following command.

```
Brocade(config)#crypto key generate dsa
```

To delete the DSA host key pair, enter the following command.

```
Brocade(config)#crypto key zeroize dsa
```

Syntax: `crypto key generate | zeroize dsa`

The **generate** keyword places a host key pair in the flash memory and enables SSH on the device, if it is not already enabled.

The **zeroize** keyword deletes the host key pair from the flash memory. This disables SSH if no other server host keys exist on the device.

The **dsa** keyword specifies a DSA host key pair. This keyword is optional. If you do not enter it, the command **crypto key generate** generates a DSA key pair by default, and the command **crypto key zeroize** works as described in [“Deleting DSA and RSA key pairs”](#) on page 183.

Generating and deleting an RSA key pair

To generate an RSA key pair, enter a command such as the following:

```
Brocade(config)#crypto key generate rsa modulus 2048
```

To delete the RSA host key pair, enter the following command.

```
Brocade(config)#crypto key zeroize rsa
```

Syntax: `crypto key generate | zeroize rsa [modulus modulus-size]`

The **generate** keyword places an RSA host key pair in the flash memory and enables SSH on the device, if it is not already enabled.

The optional [**modulus** *modulus-size*] parameter specifies the modulus size of the RSA key pair, in bits. The valid values for *modulus-size* are 1024 or 2048. The default value is 1024.

The **zeroize** keyword deletes the RSA host key pair from the flash memory. This disables SSH if no other authentication keys exist on the device.

The **rsa** keyword specifies an RSA host key pair.

NOTE

On ICX 6430 and ICX 6450 devices, the **crypto key generate** command can take up to 30 minutes to complete.

Deleting DSA and RSA key pairs

To delete DSA and RSA key pairs from the flash memory, enter the following command:

```
Brocade(config)#crypto key zeroize
```

Syntax: **crypto key zeroize**

The **zeroize** keyword deletes the host key pair from the flash memory. This disables SSH.

Providing the public key to clients

The host DSA or RSA key pair is stored in the system-config file of the Brocade device. Only the public key is readable. Some SSH client programs add the public key to the known hosts file automatically. In other cases, you must manually create a known hosts file and place the public key of the Brocade device in it.

If you are using SSH to connect to a Brocade device from a UNIX system, you may need to add the public key on the Brocade device to a “known hosts” file on the client UNIX system; for example, \$HOME/.ssh/known_hosts. The following is an example of an entry in a known hosts file.

```
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yF5SJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDiABDhtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
leg9e4NnCRleaQoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAIEALN92+Bb7D4KLYk3IwRbXblwXdkPgga4pfdtW9v
GfJ0/RHd+NjB4eolD+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uLlJn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKvlgHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXGlvo+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb1ljuqnF0GD1B3VvmxHLMxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
```

Configuring DSA or RSA challenge-response authentication

With DSA or RSA challenge-response authentication, a collection of clients' public keys are stored on the Brocade device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

When DSA or RSA challenge-response authentication is enabled, the following events occur when a client attempts to gain access to the device using SSH:

1. The client sends its public key to the Brocade device.
2. The Brocade device compares the client public key to those stored in memory.
3. If there is a match, the Brocade device uses the public key to encrypt a random sequence of bytes.
4. The Brocade device sends these encrypted bytes to the client.
5. The client uses its private key to decrypt the bytes.
6. The client sends the decrypted bytes back to the Brocade device.
7. The Brocade device compares the decrypted bytes to the original bytes it sent to the client. If the two sets of bytes match, it means that the client private key corresponds to an authorized public key, and the client is authenticated.

Setting up DSA or RSA challenge-response authentication consists of the following steps.

1. Import authorized public keys into the Brocade device.
2. Enable DSA or RSA challenge response authentication.

Importing authorized public keys into the Brocade device

SSH clients that support DSA or RSA authentication normally provide a utility to generate a DSA or RSA key pair. The private key is usually stored in a password-protected file on the local host; the public key is stored in another file and is not protected. You must import the client public key for each client into the Brocade device.

Collect one public key of each key type (DSA and/or RSA) from each client to be granted access to the Brocade device and place all of these keys into one file. This public key file may contain up to 32 keys. The following is an example of a public key file containing one public key:

```
----- BEGIN SSH2 PUBLIC KEY -----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxahvx5w0J0rzZdzoSOXxbET
W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDiABDhtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
leg9e4NnCRleaQoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eolD+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKW0ocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GD1B3VvmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
----- END SSH2 PUBLIC KEY -----
```

NOTE

Each key in the public key file must begin and end with the first and last lines in this example. If your client does not include these lines in the public key, you must manually add them.

Import the authorized public keys into the Brocade device active configuration by loading this public key file from a TFTP server.

To load a public key file called pkeys.txt from a TFTP server, enter a command such as the following:

```
Brocade(config)#ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
```

Syntax: `ip ssh pub-key-file tftp <tftp-server-ip-addr> <filename> | remove`

The `<tftp-server-ip-addr>` variable is the IP address of the tftp server that contains the public key file that you want to import into the Brocade device.

The `<filename>` variable is the name of the public key file that you want to import into the Brocade device.

The **remove** parameter deletes the public keys from the device.

To display the currently loaded public keys, enter the following command.

```
Brocade#show ip client-pub-key
---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIABDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
leg9e4NnCRleaQoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eolD+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uLlJn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetZrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKvlgHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXGlvo+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GD1B3VVMxHLMxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

Syntax: `show ip client-pub-key [begin <expression> | exclude <expression> | include <expression>]`

To clear the public keys from the buffers, enter the following command.

```
Brocade#clear public-key
```

Syntax: `clear public-key`

Enabling DSA or RSA challenge-response authentication

DSA and RSA challenge-response authentication is enabled by default. You can disable or re-enable it manually.

To enable DSA and RSA challenge-response authentication.

```
Brocade(config)#ip ssh key-authentication yes
```

To disable DSA and RSA challenge-response authentication.

```
Brocade(config)#ip ssh key-authentication no
```

Syntax: `ip ssh key-authentication yes | no`

Optional SSH parameters

You can adjust the following SSH settings on the Brocade device:

- The number of SSH authentication retries
- The user authentication method the Brocade device uses for SSH connections
- Whether the Brocade device allows users to log in without supplying a password

- The port number for SSH connections
- The SSH login timeout value
- A specific interface to be used as the source for all SSH traffic from the device
- The maximum idle time for SSH sessions

Setting the number of SSH authentication retries

By default, the Brocade device attempts to negotiate a connection with the connecting host three times. The number of authentication retries can be changed to between 1 – 5.

For example, the following command changes the number of authentication retries to 5.

```
Brocade(config)#ip ssh authentication-retries 5
```

Syntax: `ip ssh authentication-retries <number>`

Deactivating user authentication

After the SSH server on the Brocade device negotiates a session key and encryption method with the connecting client, user authentication takes place. The Brocade implementation of SSH supports DSA or RSA challenge-response authentication and password authentication.

With DSA or RSA challenge-response authentication, a collection of clients' public keys are stored on the Brocade device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

With password authentication, users are prompted for a password when they attempt to log into the device (provided empty password logins are not allowed). If there is no user account that matches the user name and password supplied by the user, the user is not granted access.

You can deactivate one or both user authentication methods for SSH. Note that deactivating both authentication methods essentially disables the SSH server entirely.

To disable DSA or RSA challenge-response authentication, enter the following command.

```
Brocade(config)#ip ssh key-authentication no
```

Syntax: `ip ssh key-authentication yes | no`

The default is **yes**.

To deactivate password authentication, enter the following command.

```
Brocade(config)#ip ssh password-authentication no
```

Syntax: `ip ssh password-authentication no | yes`

The default is **yes**.

Enabling empty password logins

By default, empty password logins are not allowed. This means that users with an SSH client are always prompted for a password when they log into the device. To gain access to the device, each user must have a user name and password. Without a user name and password, a user is not granted access.

If you enable empty password logins, users are **not** prompted for a password when they log in. Any user with an SSH client can log in without being prompted for a password.

To enable empty password logins, enter the following command.

```
Brocade(config)#ip ssh permit-empty-passwd yes
```

Syntax: `ip ssh permit-empty-passwd no | yes`

Setting the SSH port number

By default, SSH traffic occurs on TCP port 22. You can change this port number. For example, the following command changes the SSH port number to 2200.

```
Brocade(config)#ip ssh port 2200
```

Note that if you change the default SSH port number, you must configure SSH clients to connect to the new port. Also, you should be careful not to assign SSH to a port that is used by another service. If you change the SSH port number, Brocade recommends that you change it to a port number greater than 1024.

Syntax: `ip ssh port <number>`

Setting the SSH login timeout value

When the SSH server attempts to negotiate a session key and encryption method with a connecting client, it waits a maximum of 120 seconds for a response from the client. If there is no response from the client after 120 seconds, the SSH server disconnects. You can change this timeout value to between 1 – 120 seconds. For example, to change the timeout value to 60 seconds, enter the following command.

```
Brocade(config)#ip ssh timeout 60
```

Syntax: `ip ssh timeout <seconds>`

Designating an interface as the source for all SSH packets

You can designate a loopback interface, virtual interface, or Ethernet port as the source for all SSH packets from the device. For details, see [“Specifying a single source interface for specified packet types”](#) on page 977.

Configuring the maximum idle time for SSH sessions

By default, SSH sessions do not time out. Optionally, you can set the amount of time an SSH session can be inactive before the Brocade device closes it. For example, to set the maximum idle time for SSH sessions to 30 minutes, enter the following command.

```
Brocade(config)#ip ssh idle-time 30
```

Syntax: `ip ssh idle-time <minutes>`

If an established SSH session has no activity for the specified number of minutes, the Brocade device closes it. An idle time of 0 minutes (the default value) means that SSH sessions never time out. The maximum idle time for SSH sessions is 240 minutes.

Filtering SSH access using ACLs

You can permit or deny SSH access to the Brocade device using ACLs. To use ACLs, first create the ACLs you want to use. You can specify a numbered standard IPv4 ACL, a named standard IPv4 ACL

Enter commands such as the following.

```
Brocade(config)#access-list 10 permit host 192.168.144.241
Brocade(config)#access-list 10 deny host 192.168.144.242 log
Brocade(config)#access-list 10 permit host 192.168.144.243
Brocade(config)#access-list 10 deny any
Brocade(config)#ssh access-group 10
```

Syntax: `ssh access-group <standard-named-acl> | <standard-numbered-acl>`

Terminating an active SSH connection

To terminate one of the active SSH connections, enter the following command

```
Brocade#kill ssh 1
```

Syntax: `kill ssh <connection-id>`

Displaying SSH information

Up to five SSH connections can be active on the Brocade device.

Displaying SSH connection information

To display information about SSH connections, enter the **show ip ssh** command.

```
Brocade#show ip ssh
Connection  Version      Encryption      Username      HMAC          Server Hostkey      IP Address
Inbound:
  1          SSH-2        3des-cbc       Raymond      hmac-sha1     ssh-dss             10.120.54.2
Outbound:
  6          SSH-2        aes256-cbc     Steve        hmac-sha1     ssh-dss             10.37.77.15
```

```
SSH-v2.0 enabled; hostkey: DSA(1024), RSA(2048)
```

Syntax: `show ip ssh [begin <expression> | exclude <expression> | include <expression>]`

This display shows the following information about the active SSH connections.

TABLE 29 SSH connection information

Field	Description
Inbound	Connections listed under this heading are inbound.
Outbound	Connections listed under this heading are outbound.
Connection	The SSH connection ID.
Version	The SSH version number.

TABLE 29 SSH connection information (Continued)

Field	Description
Encryption	The encryption method used for the connection.
Username	The user name for the connection.
HMAC	The HMAC version
Server Hostkey	The type of server hostkey. This can be DSA or RSA.
IP Address	The IP address of the SSH client
SSH-v2.0 enabled	Indicates that SSHv2 is enabled.
hostkey	Indicates that at least one host key is on the device. It is followed by a list of the the host key types and modulus sizes.

Displaying SSH configuration information

To display SSH configuration information, use the **show ip ssh config** command:

```
Brocade# show ip ssh config
SSH server           :Enabled
SSH port             :22
Encryption           :AES-256 AES-192 AES-128 3-DES
Permit empty password :Yes
Authentication methods :Password Public-key Interactive
Authentication retries :10
Login timeout (seconds) :20
Idle timeout (minutes) :10
Strict management VRF :Enabled
SCP                  :Disabled
SSH IPv4 clients      :200.200.200.201. 200.200.200.202. 200.200.200.203
SSH IPv6 clients       :3ffe:1934:4545:3112:2040:f8ff:fe21:6001
SSH IPv4 access-list   :4
SSH IPv6 access-list   :ssh_ipv6_acl
Brocade#
```

Syntax: show ip ssh config

This display shows the following information.

Field	Description
SSH server	SSH server is enabled or disabled
SSH port	SSH port number
Encryption	The encryption used for the SSH connection. The following values are displayed when AES only is enabled: <ul style="list-style-type: none"> AES-256, AES-192, and AES-128 indicate the different AES methods used for encryption. 3-DES indicates 3-DES algorithm is used for encryption.
Permit empty password	Empty password login is allowed or not allowed.

Field	Description
Authentication methods	The authentication methods used for SSH. The authentication can have one or more of the following values: <ul style="list-style-type: none"> • Password - indicates that you are prompted for a password when attempting to log into the device. • Public-key - indicates that DSA or RSA challenge-response authentication is enabled. • Interactive - indicates the interactive authentication si enabled.
Authentication retries	The number of authentication retries. This number can be from 1 to 5.
Login timeout (seconds)	SSH login timeout value in seconds. This can be from 0 to 120.
Idle timeout (minutes)	SSH idle timeout value in minutes. This can be from 0 to 240.
Strict management VRF	Strict management VRF is enabled or disabled.
SCP	SCP is enabled or disabled.
SSH IPv4 clients	The list of IPv4 addresses to which SSH access is allowed. The default is "All".
SSH IPv6 clients	The list of IPv6 addresses to which SSh access is allowed. Default "All".
SSH IPv4 access-list	The IPv4 ACL used to permit or deny access using SSH.
SSH IPv6 access-list	The IPv6 ACL used to permit or deny access to device using SSH.

Displaying additional SSH connection information

The **show who** command also displays information about SSH connections:

```
Brocade#show who
  Console connections:
    Established
    you are connecting to this session
    2 minutes 56 seconds in idle

SSH server status: Enabled
SSH connections (inbound):
1. established, client ip address 2.2.2.1, server hostkey DSA
   1 minutes 15 seconds in idle
2. established, client ip address 2.2.2.2, server hostkey RSA
   2 minutes 25 seconds in idle
SSH connection (outbound):
3. established, server ip address 10.37.77.15, server hostkey RSA
   7 seconds in idle
```

show who [**begin** <expression> | **exclude** <expression> | **include** <expression>]

Secure copy with SSH2

Secure Copy (SCP) uses security built into SSH to transfer image and configuration files to and from the device. SCP automatically uses the authentication methods, encryption algorithm, and data compression level configured for SSH. For example, if password authentication is enabled for SSH, the user is prompted for a user name and password before SCP allows a file to be transferred. No additional configuration is required for SCP on top of SSH.

You can use SCP to copy files on the Brocade device, including the startup configuration and running configuration files, to or from an SCP-enabled remote host.

Enabling and disabling SCP

SCP is enabled by default and can be disabled. To disable SCP, enter the following command.

```
Brocade(config)#ip ssh scp disable
```

Syntax: `ip ssh scp disable | enable`

NOTE

If you disable SSH, SCP is also disabled.

Secure copy configuration notes

- When using SCP, enter the **scp** commands on the SCP-enabled client, rather than the console on the Brocade device.
- Certain SCP client options, including -p and -r, are ignored by the SCP server on the Brocade device. If an option is ignored, the client is notified.
- An SCP AES copy of the running or start configuration file from the Brocade device to Linux WS 4 or 5 may fail if the configuration size is less than 700 bytes. To work around this issue, use PuTTY to copy the file.

Example file transfers using SCP

The following are examples of using SCP to transfer files to and from a Brocade device.

Copying a file to the running config

To copy a configuration file (c:\cfg\brocade.cfg) to the running configuration file on a Brocade device at 192.168.1.50 and log in as user terry, enter the following command on the SCP-enabled client.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:runConfig
```

If password authentication is enabled for SSH, the user is prompted for user terry password before the file transfer takes place.

Copying a file to the startup config

To copy the configuration file to the startup configuration file, enter the following command.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:startConfig
```

Copying the running config file to an SCP-enabled client

To copy the running configuration file on the Brocade device to a file called c:\cfg\fdryrun.cfg on the SCP-enabled client, enter the following command.

```
C:\> scp terry@192.168.1.50:runConfig c:\cfg\brcdrun.cfg
```

Copying the startup config file to an SCP-enabled client

To copy the startup configuration file on the Brocade device to a file called c:\cfg\brcdstart.cfg on the SCP-enabled client, enter the following command.

```
C:\> scp terry@192.168.1.50:startConfig c:\cfg\brcdstart.cfg
```

To overwrite the running configuration file

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:runConfig-overwrite
```

Copying a software image file to flash memory

The **scp** command syntax differs between device series. Use the command syntax in the appropriate section.

Brocade FCX Series, ICX 6610, and FastIron X Series Devices

To copy a software image file from an SCP-enabled client to the **primary** flash on these devices, enter one of the following commands.

```
C:\> scp FCXR07000.bin terry@192.168.1.50:flash:primary
```

or

```
C:\> scp terry@192.168.1.50:flash:primary FCXR07000.bin
```

To copy a software image file from an SCP-enabled client to the **secondary** flash on these devices, enter one of the following commands.

```
C:\> scp FCXR07000.bin terry@192.168.1.50:flash:secondary
```

or

```
C:\> scp terry@192.168.1.50:flash:secondary FCXR07000.bin
```

NOTE

On ICX 6430 and ICX 6450 devices, you can use the same syntax as FCX devices. However, after the copy operation is completed at the host, you do not get the command prompt back because the switch is synchronizing the image to flash. To ensure that you have successfully copied the file, issue the **show flash** command. If the copy operation is not complete, the **show flash** command output will show the partition (primary or secondary) as **EMPTY**.

FastIron WS devices

To copy a software image file from an SCP-enabled client to the **primary** flash on these devices, enter the following command.

```
C:\> scp SXL03200.bin terry@192.168.1.50:flash:primary.bin
```

or

```
C:\> scp terry@192.168.1.50:flash:primary.bin SXL03200.bin
```

To copy a software image file from an SCP-enabled client to the **secondary** flash on these devices, enter the following command.

```
C:\> scp SXL03200.bin terry@192.168.1.50:flash:secondary.bin
```

or

```
C:\> scp terry@192.168.1.50:flash:secondary.bin SXL03200.bin
```

NOTE

The Brocade device supports only one SCP copy session at a time.

Copying a Software Image file from flash memory

The **scp** command syntax differs between device series. Use the command syntax in the appropriate section.

Brocade FCX Series, ICX Series, and FastIron X Series Devices

To copy a software image file from the **primary** flash on these devices to an SCP-enabled client, enter a command such as the following.

```
C:\> scp terry@192.168.1.50:flash:primary FCXR07000.bin
```

To copy a software image file from the **secondary** flash on these devices to an SCP-enabled client, enter a command such as the following.

```
C:\> scp terry@192.168.1.50:flash:secondary FCXR07000.bin
```

FastIron WS Series devices

To copy a software image from the **primary** flash on these devices to an SCP-enabled client, enter a command such as the following.

```
C:\> scp terry@192.168.1.50:flash:primary.bin SXL03200.bin
```

To copy a software image from the secondary flash on these devices to an SCP-enabled client, enter a command such as the following.

```
C:\> scp terry@192.168.1.50:flash:secondary.bin SXL03200.bin
```

NOTE

The Brocade device supports only one SCP copy session at a time.

Importing a digital certificate using SCP

To import a digital certificate using SCP, enter a command such as the following one:

```
C:\> scp certfile user@192.168.89.210:sslCert
```

Syntax: `scp <certificate-filename> <user>@<ip-address>:sslCert.`

The `<ip-address>` variable is the IP address of the server from which the digital certificate file is downloaded.

The `<certificate-filename>` variable is the file name of the digital certificate that you are importing to the device.

The **scp** command can be used when TFTP access is unavailable or not permitted and the command has an equivalent functionality to the **ip ssl certificate-data-file tftp**. For more information on the **ip ssl certificate-data-file tftp** command, refer to [“Importing digital certificates and RSA private key files”](#) on page 138.

Importing an RSA private key

To import an RSA private key from a client using SCP, enter a command such as the following one:

```
Brocade(config)# scp keyfile user@192.168.9.210:sslPrivKey
```

Syntax: `scp <key-filename> <user>@<ip-address>: sslPrivKey`

The `<ip-address>` variable is the IP address of the server that contains the private key file.

The `<key-filename>` variable is the file name of the private key that you want to import into the device.

The **scp** command can be used when TFTP access is unavailable or not permitted and the command has an equivalent functionality to the **ip ssl private-key-file tftp** command. For more information on the **ip ssl private-key-file tftp** command, refer to [“Importing digital certificates and RSA private key files”](#) on page 138.

Importing a DSA or RSA public key

To import a DSA or RSA public key from a client using SCP, enter a command such as the following one:

```
Brocade(config)# scp pkeys.txt user@192.168.1.234:sshPubKey
```

Syntax: `scp <key-filename> <user>@<ip-address>:sshPubKey`

The `<ip-address>` variable is the IP address of the server that contains the public key file.

The `<key-filename>` variable is the name of the DSA or RSA public key file that you want to import into the device.

The **scp** command can be used when TFTP access is unavailable or not permitted and the command has an equivalent function to the **ip ssh pub-key-file tftp** command. For more information on the **ip ssh pub-key-file tftp** command, refer to [“Importing authorized public keys into the Brocade device”](#) on page 184.

SSH2 client

SSH2 client allows you to connect from a Brocade device to an SSH2 server, including another Brocade device that is configured as an SSH2 server. You can start an outbound SSH2 client session while you are connected to the device by any connection method (SSH2, Telnet, console). Brocade devices support one outbound SSH2 client session at a time.

The supported SSH2 client features are as follows:

- Encryption algorithms, in the order of preference:
 - aes256-cbc
 - aes192-cbc
 - aes128-cbc
 - 3des-cbc
- SSH2 client session authentication algorithms:
 - Password authentication
 - Public Key authentication
- Message Authentication Code (MAC) algorithm: hmac-sha1
- Key exchange algorithm: diffie-hellman-group1-sha1
- No compression algorithms are supported.
- The client session can be established through either in-band or out-of-band management ports.
- The client session can be established through IPv4 or IPv6 protocol access.
- The client session can be established to a server listening on a non-default SSH port.

Enabling SSH2 client

To use SSH2 client, you must first enable SSH2 server on the device. See [“SSH2 authentication types”](#) on page 181.

When SSH2 server is enabled, you can use SSH client to connect to an SSH server using password authentication.

Configuring SSH2 client public key authentication

To use SSH client for public key authentication, you must generate SSH client authentication keys and export the public key to the SSH servers to which you want to connect.

The following sections describe how to configure SSH client public key authentication:

- [“Generating and deleting a client DSA key pair”](#) on page 196
- [“Generating and deleting a client RSA key pair”](#) on page 196
- [“Exporting client public keys”](#) on page 196

Generating and deleting a client DSA key pair

To generate a client DSA key pair, enter the following command.

```
Brocade(config)#crypto key client generate dsa
```

To delete the DSA host key pair, enter the following command.

```
Brocade(config)#crypto key client zeroize dsa
```

Syntax: `crypto key client generate | zeroize dsa`

The **generate** keyword places a host key pair in the flash memory.

The **zeroize** keyword deletes the host key pair from the flash memory.

The **dsa** keyword specifies a DSA host key pair.

Generating and deleting a client RSA key pair

To generate a client RSA key pair, enter a command such as the following:

```
Brocade(config)#crypto key client generate rsa modulus 2048
```

To delete the RSA host key pair, enter the following command.

```
Brocade(config)#crypto key client zeroize rsa
```

Syntax: `crypto key client generate | zeroize rsa [modulus modulus-size]`

The **generate** keyword places an RSA host key pair in the flash memory.

The **zeroize** keyword deletes the RSA host key pair from the flash memory.

The optional `[modulus modulus-size]` parameter specifies the modulus size of the RSA key pair, in bits. The valid values for *modulus-size* are 1024 or 2048. It is used only with the **generate** parameter. The default value is 1024.

The **rsa** keyword specifies an RSA host key pair.

Exporting client public keys

Client public keys are stored in the following files in flash memory:

- A DSA key is stored in the file **\$\$sshdapub.key**.
- An RSA key is stored in the file **\$\$sshrapub.key**.

To copy key files to a TFTP server, you can use the **copy flash tftp** command.

You must copy the public key to the SSH server. If the SSH server is a Brocade device, see the section [“Importing authorized public keys into the Brocade device”](#) on page 184.

Using SSH2 client

To start an SSH2 client connection to an SSH2 server using password authentication, enter a command such as the following:

```
Brocade# ssh 10.10.10.2
```

To start an SSH2 client connection to an SSH2 server using public key authentication, enter a command such as the following:

```
Brocade# ssh 10.10.10.2 public-key dsa
```

Syntax: `ssh ipv4Addr | ipv6Addr | host-name [public-key [dsa | rsa]] [port portnum]`

The *ipv4Addr* | *ipv6Addr* | *host-name* variable identifies an SSH2 server. You identify the server to connect to by entering its IPv4 or IPv6 address or its hostname.

The optional [**public-key** [**dsa** | **rsa**]] parameter specifies the type of public key authentication to use for the connection, either DSA or RSA. If you do not enter this parameter, the default authentication type is password.

The optional **port** *portnum* parameter specifies that the SSH2 connection will use a non-default SSH2 port, where *portnum* is the port number. The default port number is 22.

Displaying SSH2 client information

For information about displaying SSH2 client information, see the following sections:

- [“Displaying SSH connection information”](#) on page 188
- [“Displaying additional SSH connection information”](#) on page 190

SSH2 client

Software-based Licensing

Table 30 lists the individual Brocade FastIron switches and the software licensing features they support.

TABLE 30 Supported software licensing features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
Software-based licensing	Yes	Yes	Yes	Yes	Yes
License generation	(FESX6, FSX 800 and FSX 1600 only)				
License query					
Deleting a license					

Software license terminology

This section defines the key terms used in this chapter.

- **Entitlement certificate** – The proof-of-purchase certificate (paper-pack) issued by Brocade when a license is purchased. The certificate contains a unique transaction key that is used in conjunction with the License ID of the Brocade device to generate and download a software license from the Brocade software portal.
- **License file** – The file produced by the Brocade software portal when the license is generated. The file is uploaded to the Brocade device and controls access to a licensed feature or feature set.
- **License ID (LID)** – This is a number that uniquely identifies the Brocade device. The LID is used in conjunction with a transaction key to generate and download a software license from the Brocade software portal. The software license is tied to the LID of the Brocade device for which the license was ordered and generated.
- **Licensed feature** – Any hardware or software feature or set of features that require a valid software license in order to operate on the device.
- **Transaction key** – A unique key, along with the LID, used to generate a software license from the Brocade software portal. The transaction key is issued by Brocade when a license is purchased. The transaction key is delivered according to the method specified when the order is placed:
 - **Paper-pack** – The transaction key is recorded on an entitlement certificate, which is mailed to the customer.
 - **Electronic** – The transaction key is contained in an e-mail message, which is sent instantly to the customer after the order is placed. The customer receives the e-mail message within a few minutes after the order is placed, though the timing will vary depending on the network, Internet connection, and so on.

If a delivery method is not specified at the time of the order, the key will be delivered by the way of paper-pack.

Software-based licensing overview

Prior to the introduction of software-based licensing, Brocade supported hardware-based licensing, where an EEPROM was used to upgrade a Layer 2 or base Layer 3 switch to a premium or advanced Layer 3 switch. With the introduction of software-based licensing, one or more valid software licenses are required to run such licensed features on the device.

Software-based licensing is designed to work together with hardware-based licensing. The first release of software-based licensing employs a combination of hardware-based and software-based licensing. A Brocade device can use hardware-based licensing, software-based licensing, or both. Future releases that support software-based licensing will use software-based licensing only, eliminating the need for a customer- or factory-installed EEPROM on the management module or switch backplane.

Software-based licensing provides increased scalability and rapid deployment of hardware and software features on the supported Brocade family of switches. For example, for premium upgrades, it is no longer necessary to physically open the chassis and install an EEPROM to upgrade the system. Instead, the Web is used to generate, download, and install a software license that enables premium features on the device.

NOTE

An ICX device only supports software-based licensing. Hardware-based licensing is not supported on ICX devices.

How software-based licensing works

A permanent license can be ordered pre-installed in a Brocade device when first shipped from the factory, or later ordered and installed by the customer. In either case, additional licenses can be ordered as needed.

When a license is ordered separately (not pre-installed), an entitlement certificate or e-mail message, along with a transaction key, are issued to the customer by Brocade as proof of purchase. The transaction key and LID of the Brocade device are used to generate a license key from the Brocade software licensing portal. The license key is contained within a license file, which is downloaded to the customer's PC, where the file can then be transferred to a TFTP or SCP server, and then uploaded to the Brocade device.

Once a license is installed on the Brocade device, it has the following effects:

- For FCX and ICX devices, the license unlocks the licensed feature and it becomes available immediately. There is no need to reload the software.
- For FWS, FESX6, and FSX devices, the license unlocks the ability to upload the software image (for example, edge Layer 3 or full Layer 3) onto the device. Once the software image is installed and the software is reloaded, the license unlocks the licensed feature.

NOTE

You cannot load the edge Layer 3 software image on a FWS device without first purchasing and installing a license on the device. Likewise, you cannot load the full Layer 3 software image on a FESX6 or FSX device without first purchasing and installing a license on the device.

- When a trial license expires, the commands and CLI related to the feature are disabled, but the feature itself cannot be disabled until the system reloads.

Seamless transition for legacy devices

In this chapter, the term *legacy device* refers to a Brocade device that was shipped prior to the introduction of software-based licensing, has an EEPROM installed, and is running pre-release 07.1.00 software.

The transition to software-based licensing is seamless for legacy devices. When upgraded to a release that supports software-based licensing, these devices will continue to operate as previously configured.

NOTE

There are special considerations and instructions for legacy FastIron devices in need of replacement (by way of a Return Merchandise Agreement [RMA]). For details, refer to [“Special replacement instructions for legacy devices”](#) on page 226.

License types

The following license types are supported on FastIron devices:

- **Port-related** – Applies to FWS devices only. A port-related licensed feature enables a maximum number of ports on the device, for example 24 ports or 48 ports.
- **Application-related** – Enables premium or advanced features on the device, for example, edge Layer 3 for the FWS, advanced Layer 3 for the FCX, premium and advanced Layer 3 for ICX 6610, premium Layer 3 for ICX 6450, and full Layer 3 for the FESX6, FSX 800, and FSX 1600.
- **Trial license** – Also called a *temporary license*, this enables a license-controlled feature to run on the device on a temporary basis. A trial license enables demonstration and evaluation of a licensed feature and can be valid for a period of 45 days. For more information about a trial license, refer to [“Using a trial license”](#) on page 223.
- **Normal license** – Also called a *permanent license*, this enables a license-controlled feature to run on the device indefinitely.

Non-licensed features

[Table 31](#) lists the FastIron software images that do not require a license to run on the device.

TABLE 31 Software image files that do not require a license

Product	Image filename
FESX6	SXSxxxxx.bin (Layer 2)
FSX 800	SXLxxxxx.bin (base Layer 3)
FSX 1600	SXRxxxxx.bin
FWS	FWSxxxxx.bin (Layer 2) FWSLxxxxx.bin (base Layer 3)
FCX	FCXSxxxxx.bin (Layer 2)
ICX 6610	FCXRxxxxx.bin (Layer 3)
ICX 6430	ICX64Sxxxxx.bin (Layer 2)
ICX 6450	ICX64Rxxxxx.bin (Layer 3)

For a list of features supported with these images, refer to the release notes.

Licensed features and part numbers

[Table 32](#) lists the supported licensed features, associated image filenames, and related part numbers.

NOTE

There are no changes to the part numbers for products with pre-installed (factory-installed) licenses. These part numbers are listed for reference in the last column of [Table 32](#).

TABLE 32 Licensed features and part numbers

Product	Licensed feature or feature set	Image filename	Part numbers for software license only	Part numbers for hardware with pre-installed software license
FWS	EPREM Layer 3:	FWSRxxxxx.bin (edge Layer 3)	FWS624-L3U-SW	FWS624-EPREM
	• OSPF V2 (IPv4)		FWS624G-L3U-SW	FWS624-POE-EPREM
	• Full RIP V1 and V2			FWS624G-EPREM
	• Route-only support (Global CONFIG level only)			FWS624G-POE-EPREM
	• Route redistribution		FWS648-L3U-SW	FWS648-EPREM
	• 1020 routes in hardware maximum		FWS648G-L3U-SW	FWS648-POE-EPREM
	• VRRP-E			FWS648G-EPREM FWS648G-POE-EPREM
	Number of ports enabled:	N/A ¹	FWS624-EL3U-SW	Same part numbers (for 24-port devices) as listed above
	• 24 ports			
	• 48 ports	N/A ¹	FWS648-EL3U-SW	Same part numbers (for 48-port devices) as listed above

TABLE 32 Licensed features and part numbers (Continued)

Product	Licensed feature or feature set	Image filename	Part numbers for software license only	Part numbers for hardware with pre-installed software license
FCX	Advance Layer 3: <ul style="list-style-type: none"> BGP4 GRE 	N/A ¹	FCX-ADV-LIC-SW	FCX624-E-ADV FCX624-I-ADV FCX624S-ADV FCX624S-HPOE-ADV <hr/> FCX624S-F-ADV <hr/> FCX648-E-ADV FCX648-I-ADV FCX648S-ADV FCX648S-HPOE-ADV
ICX 6610 Software-based licensing is only supported on ICX 6610 devices.	Premium Layer 3: <ul style="list-style-type: none"> OSPF v2 OSPF v3 PIM-DM PIM-SM PIM-SSM PIM passive PBR RIP v1, v2 RIPng VRRP VRRP v3 for IPv6 VRRP-E VRRP-E for IPv6 IPv6 unicast routing 	N/A ¹	ICX 6610-PREM-LIC-SW	ICX 6610-24-PE ICX 6610-24-PI ICX 6610-24P-PE ICX 6610-24P-PI ICX 6610-24F-PE ICX 6610-24F-PI ICX 6610-48-PE ICX 6610-48-PI ICX 6610-48P-PE ICX 6610-48P-PI
	Advance Layer 3: <ul style="list-style-type: none"> All features in the Premium license (see the cell above - Premium Layer 3). BGP4 	N/A ¹	ICX 6610-ADV-LIC-SW	Sold separately. To purchase the ICX 6610 Advance license, contact your Brocade representative.
	Upgrade from Premium to Advance license This license is used to upgrade from Premium to Advance license. The license can only be installed on all SKUs that have a Premium license installed.	N/A ¹	ICX 6610-ADV-UPG-LIC-SW	Sold separately. To purchase the Upgrade license, contact your Brocade representative.
	ICX 6610 - Ports on Demand license To upgrade the ICX 6610 1 Gbps ports to 10 Gbps port speed, use the ICX6610-10G-LIC-POD license. By default, the ICX 6610 device has eight active 1 Gbps uplink ports. To increase the uplink capacity of four ports from 1 Gbps to 10 Gbps port speed, purchase a single ICX6610-10G-LIC-POD license. To increase the uplink capacity of all eight ports from 1 Gbps to 10 Gbps port speed, purchase a second ICX6610-10G-LIC-POD license.	N/A ¹	ICX 6610-10G-LIC-POD	Sold separately. To purchase the Ports on Demand license, contact your Brocade representative.
ICX 6450 Software-based licensing is only supported on ICX 6450 devices.	Premium Layer 3: <ul style="list-style-type: none"> OSPF v2 RIP v1, v2 VRRP VRRP-E 	N/A ¹	ICX6450-PREM-LIC	N/A

TABLE 32 Licensed features and part numbers (Continued)

Product	Licensed feature or feature set	Image filename	Part numbers for software license only	Part numbers for hardware with pre-installed software license
FESX6	ICX 6450- Ports on Demand license The ICX 6450 device has four active uplink or stacking ports on slot 2. By default, ports 1 and 3 are 10 Gbps ports. By default, without a license at bootup, ports 2 and 4 come up in 10 Gbps port speed in an error disabled state. To enable ports 2 and 4 to 10 Gbps port speed, purchase the ICX6450-2X10G-LIC-POD license. The PoD feature is not applicable to ICX 6430 devices because there are no 10 Gbps ports on the device.	N/A ¹	ICX6450-2X10G-LIC-POD	Sold separately. To purchase the Ports on Demand license, contact your Brocade representative.
	IPv4 PREM Layer 3: <ul style="list-style-type: none"> 6,000 active host routes Anycast RP BGP4 DVMRP V2 IGMP V1, V2, and V3 ICMP redirect messages IGMP V3 fast leave (for routing) MSDP OSPF V2 PIM-DM PIM-SM PIM passive Policy-based routing RIP V1 and V2 Route-only support Route redistribution VRRP-E 	SXRxxxxx.bin (full Layer 3)	FESX624-L3U-IPV4-SW	FESX624-PREM FESX624-PREM-DC FESX624+2XG-PREM FESX624+2XG-PREM-DC FESX624HF-PREM FESX624HF-PREM-DC FESX624HF+2XG-PREM FESX624HF+2XG-PREM-DC
			FESX648-L3U-IPV4-SW	FESX648-PREM FESX648-PREM-DC FESX648+2XG-PREM FESX648+2XG-PREM-DC
	IPv6 PREM Layer 3: <ul style="list-style-type: none"> Same features as "IPv4 PREM Layer 3:", plus the following: IPv6 Layer 3 forwarding IPv6 over IPv4 tunnels in hardware IPv6 redistribution IPv6 static routes OSPF V3 RIPng 	SXRxxxxx.bin (full Layer 3)	FESX624-L3U-IPV6-SW	FESX624-PREM6 FESX624-PREM6-DC FESX624+2XG-PREM6 FESX624+2XG-PREM6-DC FESX624HF-PREM6 FESX624HF-PREM6-DC FESX624HF+2XG-PREM6 FESX624HF+2XG-PREM6-DC
			FESX648-L3U-IPV6-SW	FESX648-PREM6 FESX648-PREM6-DC FESX648+2XG-PREM6 FESX648+2XG-PREM6-DC

TABLE 32 Licensed features and part numbers (Continued)

Product	Licensed feature or feature set	Image filename	Part numbers for software license only	Part numbers for hardware with pre-installed software license
FSX 800 and FSX 1600	IPv4 PREM Layer 3 for IPv4 management modules: <ul style="list-style-type: none"> • 6,000 active host routes • Anycast RP • BGP4 • DVMRP V2 • IGMP V1, V2, and V3 • ICMP redirect messages • IGMP V3 fast leave (for routing) • MSDP • OSPF V2 • PIM-DM • PIM-SM • PIM passive • Policy-based routing • RIP V1 and V2 • Route-only support • Route redistribution • VRRP-E • IPv4 over GRE • Multicast over GRE 	SXRxxxxx.bin (full Layer 3)	SX-FIL3U-SW	SX-FIZMR-PREM SX-FI2XGMR4-PREM
	IPv4 PREM Layer 3 for IPv6-ready management modules: Same features as “IPv4 PREM Layer 3 for IPv4 management modules:”	SXRxxxxx.bin (full Layer 3)	SX-FIL3U-6-IPV4-SW	SX-FIZMR-6-PREM SX-FI2XGMR6-PREM
	IPv4 and IPv6 PREM Layer 3 for IPv6-ready management modules: Same features as “IPv4 PREM Layer 3 for IPv4 management modules:” , plus the following: <ul style="list-style-type: none"> • IPv6 Layer 3 forwarding • IPv6 over IPv4 tunnels in hardware • IPv6 redistribution • IPv6 static routes • OSPF V3 • RIPng • IPv6 over GRE 	SXRxxxxx.bin (full Layer 3)	SX-FIL3U-6-IPV4-SW and SX-FIL3U-6-IPV6-SW	SX-FIZMR-6-PREM6 SX-FI2XGMR6-PREM6

1. This licensed feature does not require a separate software image file. Feature capability is disabled on the switch until a license is loaded.

Table 33 lists the supported software packages.

TABLE 33 Software packages

Product	Software package name	License needed?
FWS	BASE_SOFT_PACKAGE	No
	FWS_BASE_L3_SOFT_PACKAGE	No
	FWS_EDGE_SOFT_PACKAGE	Yes
FCX	BASE_SOFT_PACKAGE	No
	FCX_FULL_ROUTER_SOFT_PACKAGE	No
	FCX_ADV_ROUTER_SOFT_PACKAGE	Yes
ICX 6610	ICX6610_BASE_ROUTER_SOFT_PACKAGE	No
	ICX6610_PREM_ROUTER_SOFT_PACKAGE	Yes
	ICX6610_ADV_ROUTER_SOFT_PACKAGE	Yes
ICX 6450	ICX6450_BASE_ROUTER_SOFT_PACKAGE	No
	ICX6450_PREM_ROUTER_SOFT_PACKAGE	Yes
FESX6, FSX 800, FSX 1600	BASE_SOFT_PACKAGE	No
	SX_V4_HW_BASE_L3_SOFT_PACKAGE	No
	SX_V4_HW_ROUTER_SOFT_PACKAGE	Yes
	SX_V6_HW_BASE_L3_SOFT_PACKAGE	No
	SX_V6_HW_ROUTER_SOFT_PACKAGE	Yes
	SX_V6_HW_ROUTER_IPV6_SOFT_PACKAGE	Yes

Licensing rules

This section lists the software licensing rules and caveats related to the Brocade devices that support software-based licensing.

General notes about licensing

The following licensing rules apply to all FastIron devices that support software licensing:

- A license is tied to the unique LID of the management module or fixed configuration switch for which the license was ordered. Therefore, a license can be used on one device only. It cannot be used on any other device.
- More than one license can be installed per device. For example, an FSX 800 with IPv6 hardware can have the license SX-FIL3U-6-IPV4 and the license SX-FIL3U-6-IPV6, and both can be in effect.
- Only one normal or trial license at a time can be in effect for a licensed feature.
- More than one trial license can be in effect at the same time, as long as each trial license applies to a unique licensed feature.
- A trial license cannot replace or supersede a normal license.

Licensing rules for FCX and ICX 6610 devices

The following licensing rules apply to FCX and ICX 6610 devices for software-based licensing. To describe the behavior for running software-based licensing in an FCX IronStack, the FCX-ADV-LIC-SW license is used as an example.

NOTE

For FCX and ICX 6610 devices, the behavior for running software-based licensing with different licenses (Premium, Advance, or Upgrade licenses) is the same. One license allows multiple protocols to run in a stack. All units must have a separate license to run the same licensed feature in a stack. If all units do not have the same license, the Active controller cannot enable the licensed feature on the stack. If a member unit without a license joins a stack, the Active controller must make sure that no protocols are enabled in a stack before putting a member unit into full operational state.

- Each stack unit in an FCX IronStack must have a separate software license for the same licensed feature. For example, if there are eight units in an IronStack, eight separate licenses must be purchased to run the licensed features in the stack. If there is any unit in a stack without the FCX-ADV-LIC-SW license, the Active controller cannot run the licensed features on the stack.
- For example, to run BGP on the stack, the **router bgp** command must be enabled through the CLI on the Active controller. If the Active controller does not have the FCX-ADV-LIC-SW license, the user cannot configure the **router bgp** command at the CLI level.
- If the Active controller has the FCX-ADV-LIC-SW license, and the **router bgp** command is enabled at the CLI level, the system checks all operational units to verify that each unit has the FCX-ADV-LIC-SW license. Only if all the operational units have the FCX-ADV-LIC-SW license will the licensed feature run in the stack.
- If any unit does not have the FCX-ADV-LIC-SW license, the **router bgp** command is rejected and the licensed feature cannot run in the stack. For example, the following error message is displayed on the console.

```
Brocade(config)#router bgp
Error! cannot run BGP because unit 2 has no FCX-ADV license
```

- If the Active controller is running BGP (and all other licensed features if enabled), and a unit without the FCX-ADV-LIC-SW license joins the stack, the unit is put into a non-operational state. If a user copies the FCX-ADV-LIC-SW license to a non-operational unit, it takes effect immediately and the unit becomes operational. If the operational unit has a higher priority than the current Standby controller, the unit replaces the existing Standby controller and becomes the new Standby controller. This behavior applies to all cases in which a non-operational unit becomes operational.
- If a user disables BGP from the stack, the Active controller puts all non-operational units in the operational state. The Active controller and the Standby controller must have the same non-operational units. When the Standby controller receives the runtime configuration from the Active controller, the Standby controller must update the state of every unit (operational or non-operational state).
- If a user deletes the FCX-ADV-LIC-SW license in any stack unit, the Active controller does not change the unit to the non-operational state regardless of running BGP or not. Even if a new election algorithm change occurs, an operational unit is not changed to a non-operational unit. The stack continues its BGP running state. The stack cannot run BGP again after the user disables BGP, or after a reload.

- If BGP is not enabled on the Active controller, a stack unit is operational whether or not the Active controller or the stack units have the FCX-ADV-LIC-SW license. This implies that in a stack where all units (Active controller, Standby controller, and member units) have the FCX-ADV-LIC-SW license, a stack can be formed whether or not BGP is enabled. However, if there is a license mismatch between any of the units in a stack, a stack can still be formed provided that the **router bgp** command is not enabled on the Active controller.
- The FCX-ADV-LIC-SW license is not considered when selecting a unit to be the Standby controller.

Licensing rules for FESX6, FSX 800, and FSX 1600 devices

SX 800 and SX 1600 devices with redundant management modules must have two separate licenses to run the same licensed feature on both management modules. The license file in the active management module is never copied to or updated on the standby management module.

Upon bootup, the active management module compares its license with the standby management module. If the license differs, the active management module immediately shuts down the standby management module. To enable the standby management module, you must install a separate license. For example, if the active management module has the license SX-FIL3U-SW, the standby management module must also have this license.

Licensing for Ports on Demand

NOTE

Licensing for Ports on Demand is applicable to ICX 6610 devices and ICX 6450 devices.

You can use the Ports on Demand (PoD) feature to enable 1 Gbps ports to 10 Gbps port speed.

By default, regardless of what SFP+ media optic is used, the ICX 6610 device has eight active 1 Gbps uplink ports. To increase the uplink capacity of four ports from 1 Gbps to 10 Gbps port speed, purchase a single ICX6610-10G-LIC-POD license. To increase the uplink capacity of all eight ports from 1 Gbps to 10 Gbps port speed, purchase a second ICX6610-10G-LIC-POD license.

The ICX 6450 device has four active uplink and stacking ports on slot 2. By default, regardless of what SFP+ media optic is used, ports 1 and 3 are 10 Gbps ports. By default, without a license at bootup, ports 2 and 4 come up in 10 Gbps port speed in an error disabled state. To enable ports 2 and 4 to 10 Gbps port speed, purchase the ICX6450-2X10G-LIC-POD license. For more information about enabling ports 2 and 4 to 10 Gbps port speed, refer to [“Configuration considerations when configuring PoD for ICX 6450 devices only”](#) on page 214.

The PoD feature is not applicable to ICX 6430 devices because there are no 10 Gbps ports on the device.

Configuring PoD on an interface

To upgrade the ICX 6610 and ICX 6450 ports from 1 Gbps to 10 Gbps port speed, perform the following steps.

1. Download the PoD license to the device. For more information about copying the license file on ICX devices, refer to [“Using TFTP to copy a license file on FCX and ICX devices”](#) on page 221.
2. Insert the 10 Gbps optic transceiver.
3. Enter the **speed-duplex 10g-full** command on a single, multiple, or interface range as shown in the example below.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# speed-duplex 10g-full
```

4. Enter the **write memory** command to save the configuration.

Syntax: [no] **speed-duplex [10g-full | 1000-full-master]**

The **10g-full** parameter allows you to enable the port speed to 10 Gbps speed.

The **1000-full-master** parameter allows you to enable the port speed to 1 Gbps speed.

If you enable the port to 10 Gbps port speed, and then issue the **no speed-duplex** command, the port continues to run at 10 Gbps speed. Upon bootup, the port reverts to 1 Gbps speed. In a stacking environment, if you issue the **no speed-duplex 10g-full** command on a 10 Gbps port interface, or a range of interfaces, the 10 Gbps port interface defaults back to 1 Gbps port speed in five to ten seconds.

Configuring the upper PoD ports in a stack for ICX 6610 devices only

By default, when a single ICX6610-10G-LIC-POD license is downloaded onto the device, all four lower PoD ports in the stack (1/3/1 to 1/3/4) are eligible for an upgrade to 10 Gbps port speed. If you have a single ICX6610-10G-LIC-POD license, and you want to enable the upper four PoD ports (1/3/5 to 1/3/8) to 10 Gbps port speed, instead of the lower four PoD ports (1/3/1 to 1/3/4), complete the following steps. If the lower four PoD ports are already configured for 10 Gbps speed, you must first issue the **no speed-duplex 10g-full** command on the lower four PoD ports before configuring the upper four PoD ports to 10 Gbps port speed.

The procedure below assumes that you have already downloaded the license.

NOTE

If you purchased a second ICX6610-10G-LIC-POD license, you do not need to perform the steps outlined below. The **flexible-10g-ports upper** command is used to configure the upper four PoD ports to 10 Gbps port speed using a single ICX6610-10G-LIC-POD license. The command can only be used on the upper four PoD ports.

1. Enter the **flexible-10g-ports upper** command at the stack unit configuration level.

```
Brocade(config)# stack unit 2
Brocade(config-unit-2)#flexible-10g-ports upper
Brocade(config-unit-2)#exit
```

2. Specify the upper four PoD ports in a group with a single ICX6610-10G-LIC-POD license at the interface configuration level.

```
Brocade(config)#interface ethernet 2/3/5 to 2/3/8
```

3. Enable the 10 Gbps port speed for the upper four PoD ports by entering the following command.

```

Brocade(config-mif-2/3/5-2/3/8)#speed-duplex 10g-full
Port 2/3/5 mode changed from 1G to 10G
Port 2/3/6 mode changed from 1G to 10G
Port 2/3/7 mode changed from 1G to 10G
Port 2/3/8 mode changed from 1G to 10G
Brocade(config-mif-2/3/5-2/3/8)#end

```

Syntax: [no] flexible-10g-ports upper

Use the **no** form of the **flexible-10g-ports upper** command when you want to enable the lower four PoD ports, instead of the upper four PoD ports, to 10 Gbps port speed. Before you issue the **no flexible-10g-ports upper** command, you must first issue the **no speed-duplex 10g-full** command on the upper four PoD ports.

To display the configuration for the **flexible-10g-ports upper** command on a stack unit, use the **show stack <stack-unit>** command.

```

Brocade#show stack 1
stack unit 1
  module 1 icx6610-24f-sf-port-management-module
  module 2 icx6610-qsfp-10-port-160g-module
  module 3 icx6610-8-port-10g-dual-mode-module
  stack-trunk 1/2/1 to 1/2/2
  stack-trunk 1/2/6 to 1/2/7
  stack-port 1/2/1 1/2/6
  flexible-10g-ports upper

```

Syntax: show stack <stack-unit>

Displaying license configuration for PoD ports after a license upgrade

NOTE

The **show pod** command cannot be used to display the configuration for 1 Gbps or 40 Gbps ports on ICX 6610 and ICX 6450 devices.

The **show pod** command is used to display a license configuration for PoD ports in a stack after a license upgrade on ICX 6610 and ICX 6450 devices. To display general license information about the PoD license in a stack unit, use the **show license** command. For more information about the **show license** command, refer to [“Viewing the license database”](#) on page 229.

Displaying license configuration for PoD ports for ICX 6610 devices

To display a license configuration for all PoD ports for all units in a stack, enter the following command at the CLI level.

```

Brocade#show pod
Unit-Id: 1
PoD license capacity: 4      (UPPER)
PoD license capacity used: 4

PoD-ports  Lic-state
1/3/1      default
1/3/2      default
1/3/3      default
1/3/4      default
1/3/5      acquired

```



```

1/3/6      acquired
1/3/7      acquired
1/3/8      acquired

Unit-Id: 2
PoD license capacity:  8
PoD license capacity used:  8

PoD-ports  Lic-state
2/3/1      acquired
2/3/2      acquired
2/3/3      acquired
2/3/4      acquired
2/3/5      acquired
2/3/6      acquired
2/3/7      acquired
2/3/8      acquired

Unit-Id: 3
PoD license capacity:  4      (LOWER)
PoD license capacity used:  4

PoD-ports  Lic-state
3/3/1      acquired
3/3/2      acquired
3/3/3      acquired
3/3/4      acquired
3/3/5      default
3/3/6      default
3/3/7      default
3/3/8      default

```

Syntax: `show pod [unit <id>]`

The **unit <id>** parameter specifies the unit ID number of the PoD you want to display.

[Table 34](#) describes the information displayed in the output of the **show pod unit** command.

TABLE 34 Output from the **show pod unit** command

Field	Description
Unit-Id	The unit ID number of the PoD.
PoD license capacity	The port capacity of the PoD license that is purchased. For ICX 6610 devices, the port capacity can be four or eight 10 Gbps ports. UPPER is displayed in parentheses when the upper four PoD ports are selected using the flexible-10g-ports upper command. LOWER is displayed in parentheses when the four lower PoD ports are selected for an upgrade to 10 Gbps port speed.
PoD license capacity used	The number of PoD ports that are upgraded to 10 Gbps port speed.
PoD-ports	The list of PoD ports in the PoD unit.
Lic-state	The license state of the PoD ports in the PoD unit. The Lic-state can be one of the following: <ul style="list-style-type: none"> default - The port is configured to 1 Gbps speed. acquired - The port is configured to 10 Gbps speed.

Displaying license configuration for PoD ports for ICX 6450 devices

By default at bootup, the license state for ports 2 and 4 are in the acquired state. The following output from the **show pod** command displays port 2 and 4 as acquired. Upon installing the ICX6450-2X10G-LIC-POD license, ports 2 and 4 can be enabled to run in 10 Gbps port speed. The license state for ports 2 and 4 remains in the acquired state.

```
Brocade#show pod
Unit-Id: 1
PoD-ports Lic-state
1/2/2 acquired
1/2/4 acquired
```

If ports 2 and 4 are configured to 1 Gbps port speed, the license state changes to default. The following output from the **show pod** command displays port 2 and 4 in the default state.

```
Brocade#show pod
Unit-Id: 1
PoD-ports Lic-state
1/2/2 default
1/2/4 default
```

Syntax: **show pod** [unit <id>]

The **unit** <id> parameter specifies the unit ID number of the PoD you want to display.

For a description of the fields in the **show pod** and **show pod** [unit <id>] command outputs, refer to [Table 34](#) on page 211.

Upgrading or downgrading configuration considerations for PoD

Consider the following when upgrading or downgrading PoD ports for ICX 6610 or ICX 6450 devices:

- When a single ICX6610-10G-LIC-POD license is downloaded onto the device, you can upgrade the first four or the last four PoD ports to 10 Gbps port speed. However, if you upgrade the fifth port to 10 Gbps port speed, the following syslog message and error message displays.

```
SYSLOG: <14>Jul 31 00:33:46 10.20.147.22 PoD: No license present for port
1/3/3".
POD:No license present for port 3.
```

The error message is displayed because the port you are attempting to upgrade to 10 Gbps port speed has exceeded the license capacity that is downloaded onto the device. To upgrade all eight ICX 6610 ports to 10 Gbps port speed, purchase a second ICX6610-10G-LIC-POD license.

- There is no trial license available for the PoD feature for ICX 6610 and ICX 6450 devices.

Configuration considerations for stacking or trunking PoD ports

Consider the following when stacking or trunking PoD ports for ICX 6610 or ICX 6450 devices:

- In an ICX Ironstack, a stack member unit without a PoD license can join a stack even when the active or master stack unit has a PoD license.
- All trunk ports must operate at 1 Gbps or 10 Gbps speed in a stack. You cannot mix and match trunk ports with different port speeds.
- In a trunk formation, if there is no license installed in a stack upon bootup or hot swap of a stack unit, a port is disabled. This does not affect the trunk formation.

Configuration considerations when configuring PoD on an interface

Consider the following when configuring PoD on an interface or a range of interfaces on ICX 6610 or ICX 6450 devices:

- The **speed-duplex 10g-full** command is rejected when there is no license or an insufficient license in the unit. For example, an insufficient license implies that you are attempting to upgrade all eight ports to 10 Gbps port speed, and you have downloaded a single ICX6610-10G-LIC-POD license. The following syslog message and error message is generated.

```
SYSLOG: <14>Jul 31 00:33:46 10.20.147.22 PoD: No license present for port 1/3/3".
POD:No license present for port 2
```

- When the **speed-duplex 10g-full** command is configured for a port or multiple ports on an interface, and there is no license or an insufficient license in the stack upon bootup or hot swap of a stack unit, the port is configured to 10 Gbps port speed. However, the port is in ERROR_DISABLED state until you install the correct license in the stack. A syslog message is generated every 30 seconds stating that the license is not present on the port. The error disable recovery timer checks every 30 seconds to see if the correct license is installed in the stack. Once the correct license is installed, the port is automatically enabled to operate at 10 Gbps port speed on the next cycle of the timer. You can also enable the port manually to operate at 10 Gbps port speed once the correct license is installed.
- In a stack, the **speed-duplex 10g-full** command is rejected on the primary port if all ports in a trunk do not have the correct license installed to upgrade to 10 Gbps port speed. The following error message is displayed.

```
Error:Not enough 10Gig License present for all the ports in trunk
```

- If you delete the license from the stack, the port runs in 10 Gbps mode until the switch is reloaded. If the **speed-duplex 10g-full** command is entered, the following syslog message is displayed.

```
SYSLOG: <14>Jul 31 00:33:46 10.20.147.22 PoD: No license present for port 1/3/3".
```

- The **show interface ethernet <stack-unit>/<slotnum>/<portnum>** command can be used to display the configuration for a disabled 10 Gbps interface port, or a range of port interfaces after bootup for both ICX 6610 and ICX 6450 devices. The **show interface ethernet** command displays the port in the ERROR_DISABLED state. The following example output is from an ICX 6450 device.

```
Brocade#show interface ethernet 3/2/2
10GigabitEthernet3/2/2 is ERR-DISABLED (invalid license), line protocol is
down
  Hardware is 10GigabitEthernet, address is 748e.f883.01fa (bia
748e.f883.01fa)
  Configured speed 10Gbit, actual unknown, configured duplex fdx, actual
unknown
  Stacking Port, port state is DISABLED
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0, mac-learning is enabled
  Flow Control is enabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
```

Once the correct license is installed, the port displays the state as up as shown in the example output below.

```
Brocade#show interface ethernet 3/2/2
10GigabitEthernet6/2/2 is up, line protocol is up
  Hardware is 10GigabitEthernet, address is 748e.f882.f872 (bia 748e.f882.f872)
  Configured speed 10Gbit, actual 10Gbit, configured duplex fdx, actual fdx
  Stacking Port, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0, mac-learning is enabled
  Flow Control is enabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
```

Configuration considerations when configuring PoD for ICX 6450 devices only

Consider the following when configuring PoD for ICX 6450 devices only:

- By default, without a license at bootup, ports 2 and 4 come up in 10 Gbps port speed in an error disabled state. The **show interface ethernet** command displays the port in the `ERROR_DISABLED` state at bootup. Refer to section, [“Configuration considerations when configuring PoD on an interface”](#) on page 213 for output example from the **show interface ethernet** command.
- From the default state, ports 2 and 4 can be configured to 1 Gbps port speed using the **speed-duplex 1000-full-master** command without a license. You do not need to reboot the switch for the links to come up in 1 Gbps port speed.
- If you download the ICX6450-2X10G-LIC-POD license to the device, insert the correct 10 Gbps optic transceiver, and enter the **speed-duplex 10g-full** command on the interface, you can immediately begin using ports 2 and 4 in 10 Gbps port speed. You do not need to reboot the switch for the links to come up.

- For any of the four uplink ports on slot 2, if you re-configure any port from 1 Gbps to 10 Gbps port speed, you must reload the switch to begin using the ports in 10 Gbps port speed. Until you reload the switch, the ports will remain in an error-disabled state. The following example output displays ethernet port 4 in an error-disabled state.

```
Brocade#show interface ethernet 1/2/4
10GigabitEthernet1/2/4 is ERR-DISABLED (Reload the switch or stack to enable
this port in 10G speed), line protocol is down
  Hardware is 10GigabitEthernet, address is 748e.f882.e39c (bia 748e.f882.e39c)
  Configured speed 10Gbit, actual unknown, configured duplex fdx, actual
unknown
  Member of L2 VLAN ID 1, port is untagged, port state is DISABLED
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0, mac-learning is enabled
  Flow Control is enabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1500 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
```

Software licensing configuration tasks

This section describes the configuration tasks for generating and obtaining a software license, and then installing it on the Brocade device. Perform the tasks in the order listed in [Table 35](#).

TABLE 35 Configuration tasks for software licensing

Configuration task	Reference...
1 Order the desired license.	For a list of available licenses and associated part numbers, refer to “Licensed features and part numbers” on page 202.
2 When you receive the transaction key, retrieve the LID of the Brocade device. If you received the transaction key by way of paper-pack, record the LID on the entitlement certificate in the space provided.	“Viewing the License ID” on page 227
3 Log in to the Brocade software portal to generate and obtain the license file.	“Obtaining a license” on page 216
4 Upload the license file to the Brocade device.	“Installing a license file” on page 221
5 Verify that the license is installed.	“Using a trial license” on page 223
6 For FESX6, FSX 800, FSX 1600, FWS, FCX, ICX 6610, and ICX 6450 devices, upload the software image to the device.	Refer to the release notes.

Obtaining a license

The procedures in this section show how to generate and obtain a software license.

1. Order a license for the desired licensed feature. Refer to [“Licensed features and part numbers”](#) on page 202 for a list of valid part numbers and licensed features.

NOTE

To order and obtain a trial license, contact your Brocade representative.

2. When you receive the paper-pack or electronic transaction key, retrieve the LID of your Brocade device by entering the **show version** command on the device. Example command output is shown in [“Viewing the License ID”](#) on page 227.”

If you received a paper-pack transaction key, write the LID in the space provided on the entitlement certificate.

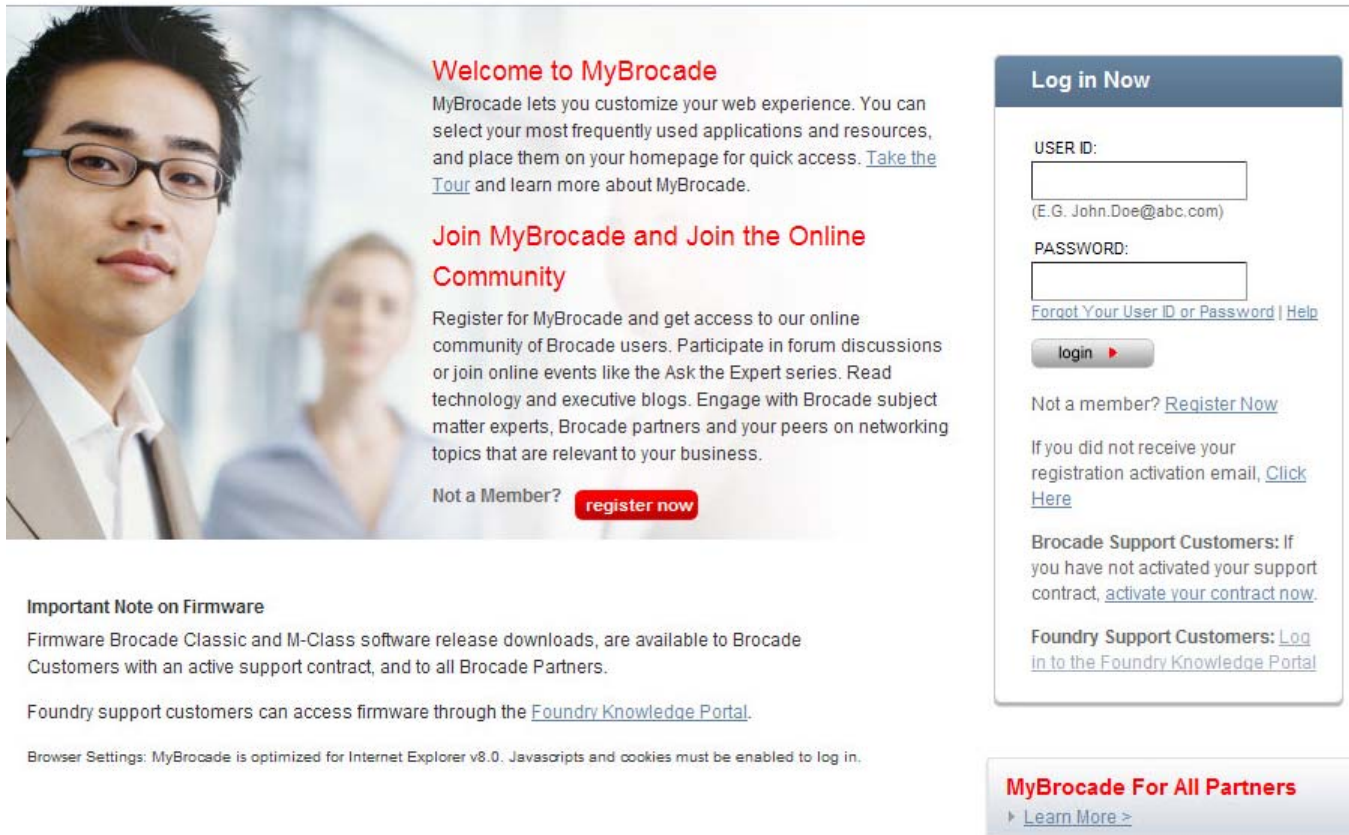
NOTE

Do not discard the entitlement certificate or e-mail with electronic key. Keep it in a safe place in case it is needed for technical support or product replacement (RMAs).

3. Log in to the Brocade software portal at <http://swportal.brocade.com> and complete the software license request. If you do not have a login ID and password, request access by following the instructions on the screen.

Figure 5 shows the **Software Portal Login** window.

FIGURE 5 Brocade Software Portal Login window



Welcome to MyBrocade

MyBrocade lets you customize your web experience. You can select your most frequently used applications and resources, and place them on your homepage for quick access. [Take the Tour](#) and learn more about MyBrocade.

Join MyBrocade and Join the Online Community

Register for MyBrocade and get access to our online community of Brocade users. Participate in forum discussions or join online events like the Ask the Expert series. Read technology and executive blogs. Engage with Brocade subject matter experts, Brocade partners and your peers on networking topics that are relevant to your business.

Not a Member? [register now](#)

Log in Now

USER ID:

 (E.G. John.Doe@abc.com)

PASSWORD:

[Forgot Your User ID or Password](#) | [Help](#)

[login](#)

Not a member? [Register Now](#)

If you did not receive your registration activation email, [Click Here](#)

Brocade Support Customers: If you have not activated your support contract, [activate your contract now](#).

Foundry Support Customers: [Log In to the Foundry Knowledge Portal](#)

MyBrocade For All Partners

[Learn More >](#)

Important Note on Firmware

Firmware Brocade Classic and M-Class software release downloads, are available to Brocade Customers with an active support contract, and to all Brocade Partners.

Foundry support customers can access firmware through the [Foundry Knowledge Portal](#).

Browser Settings: MyBrocade is optimized for Internet Explorer v8.0. Javascripts and cookies must be enabled to log in.

From the **License Management** menu, select **Brocade IP/ADP > License Generation with Transaction key**. The **IP/ADP License Generation** dialog box displays.

FIGURE 6 License Management Welcome window



Figure 7 shows the **IP/ADP License Generation** dialog box for generating a license using a transaction key and LID.

FIGURE 7 IP/ADP License Generation window

BROCADE support1 Log Out

License Management

IP/ADP Licence Generation
Please check the Unique ID to make sure it is correct! License install failure may result!

Customer Information

* Indicates required field

Customer email ID*

Site Name

Technical Contact

Company Name

City

State/Province

Zip/Postal Code

Country

Phone

E-mail Notification Information

Self

☐ Other e-mail address(es)

Unit Information

Unique ID Type*

Unit's Unique ID*

Transaction Key*

☐ I have read and accept the [Brocade End User License Agreement](#)

Enter the required information.

- For a description of the field, move the pointer over the field.
- An asterisk next to a field indicates that the information is required.

NOTE

You can generate more than one license at a time. For each license request, enter the **Unit's Unique ID** and **Transaction Key**, and click **Add**.

When you have finished entering the required information, read the Brocade End User License Agreement, and select the **I have read and accept** check box.

Click the **Generate** button to generate the license. Figure 8 shows the results window, which displays an order summary and the results of the license request.

- If the license request is successful, the **Status** field shows “Success” and the **License File** field contains a hyperlink to the generated license file. The license file is automatically sent by e-mail to the specified customer e-mail address.
- If the license request fails, the **Status** field indicates the reason it failed and the action to be taken.

FIGURE 8 IP/ADP License Generation Results window

IP/ADP License Generation - Result

Customer Information

Customer email ID: partner501@company.com
 Site Name
 Technical Contact
 Company Name
 City
 State/Province
 Zip/Postal Code
 Country
 Phone

Following Generated Licenses have been sent to Email ID(s): partner501@company.com

ID Type	Unique Id	Transaction Key	Description	Status	License File
LID	pkeguceGFHM	A0D57D0038E39D9427131B	BR-NI-CES-2048-L3U	Success	License Key

Generate Another License

4. Download the license file to your PC by either clicking the hyperlink in the **License File** field or saving the license file from the e-mail attachment.
5. Upload the license file to the Brocade device as instructed in [“Installing a license file”](#) on page 221.

Installing a license file

Once you obtain a license file, place it on a TFTP or SCP server to which the Brocade device has access, and then use TFTP or SCP to copy the file to the license database of the Brocade device.

Using TFTP to copy a license file on FESX, SX 800 and SX 1600, and FWS devices

To copy a license file from a TFTP server to the license database of the Brocade device, enter a command such as the following at the Privileged EXEC level of the CLI:

```
Brocade# copy tftp license 10.1.1.1 lic.xml
```

Syntax: `copy tftp license [<IP_address> | <ipv6_address>] <license_filename_on_host>`

The `<IP_address>` variable is the address of the IPv4 TFTP server.

The `<ipv6_address>` variable is the address of the IPv6 TFTP server.

The `<license_filename_on_host>` variable is the filename of the license file.

If you attempt to download the same license twice on the device, the following error message is displayed on the console.

```
Can't add the license string - 93 (DUPLICATE_LICENSE)
```

Using TFTP to copy a license file on FCX and ICX devices

You can copy a license file from the active unit to all other member units without having to physically disable the stack to install a license for each unit. To copy a license file from the active unit to all other member units in the system, enter a command such as the following at the Privileged EXEC level of the CLI:

```
Brocade# copy tftp license 10.120.54.185 FCX_ADV_LIC_PERP.xml unit 2
```

Syntax: `copy tftp license [<IP_address> | <ipv6_address>] <license_filename_on_host> unit <unit_id>`

The `<IP_address>` variable is the address of the IPv4 TFTP server.

The `<ipv6_address>` variable is the address of the IPv6 TFTP server.

The `<license_filename_on_host>` variable is the filename of the license file.

The `unit <unit_id>` parameter specifies a unit for which you want to add a software license file. The `<unit_id>` variable can be from 1 through 8.

If you attempt to download the same license twice on the device, the following error message is displayed on the console.

```
Can't add the license string - 93 (DUPLICATE_LICENSE)
```

Using Secure Copy to install a license

SSH and Secure Copy (SCP) must be enabled on the Brocade device before the procedures in this section can be performed. For details, refer to the chapter [Chapter 5, "SSH2 and SCP"](#).

Deleting a license file

The **scp** *<license_file_on_host>* *<user>@<IP_address>:license* command is supported on FESX, SX 800 and SX 1600, and FWS devices.

To copy a license file from an SCP-enabled client to the license database of the Brocade device, enter a command such as the following on the SCP-enabled client:

```
c:\scp c:\license\license101 terry@10.1.1.1:license
```

Syntax: **scp** *<license_file_on_host>* *<user>@<IP_address>:license*

On FCX and ICX devices, to copy a license file from an SCP-enabled client to the license database of a specific unit, enter a command such as the following on the SCP-enabled client:

```
scp license.xml terry@10.20.91.39:license:3
```

In the example above, the license is copied to unit 3.

Syntax: **scp** *<license_file_on_host>* *<user>@<IP_address>:license:<unit id>*

The **unit** *<unit_id>* parameter specifies a unit for which you want to add a software license file. The *<unit_id>* variable can be from 1 through 8.

Verifying the license file installation

Use the **show license** command to verify that the license is installed on the device. Details about this command are in [“Viewing information about software licenses”](#) on page 227.

Deleting a license file

A license remains in the license database until it is deleted. If you want to delete a license, Brocade recommends that you first disable the licensed feature before deleting the associated license.

NOTE

If a trial license and a normal license for the same licensed feature are added to the system (in either order), the normal license cannot be deleted first. The trial license should be deleted first, and then the normal license can be deleted. If necessary, the same trial license can be reinstalled and used as long as it has not expired.

Deleting a license on FESX, SX 800 and SX 1600, and FWS devices

To delete a license, enter a command such as the following at the Privileged EXEC level of the CLI:

```
Brocade# license delete 7
```

This command immediately removes the license from the license database. The CLI commands related to the licensed feature will no longer be available from the CLI. The licensed feature will continue to run as configured until the software is reloaded, at which time the feature will be disabled and removed from the system. Syslog and trap messages are generated when the license is deleted.

Syntax: **license delete** *<index_number>*

The `<index_number>` variable is a valid license index number. The license index number can be retrieved from the **show license** command output. For more information, refer to [“Viewing information about software licenses”](#) on page 227.

Deleting a license on FCX and ICX devices

To delete all license files for a specific unit, enter the following command at the Privileged EXEC level of the CLI:

```
Brocade# license delete unit 3 all
```

To delete a specific license file from a unit, enter the following command at the Privileged EXEC level of the CLI:

```
Brocade# license delete unit 3 index 3
```

Syntax: `license delete unit <unit_id> [all | index <license_index>]`

The `<unit_id>` variable specifies the unit ID number.

The **all** option allows you to delete all license files for a specific unit.

The **index <license_index>** parameter specifies the software license file, and is generated by the member unit. The license index number is the license file you want to delete from a unit. The license index number is not unique across stack units, and the user must specify both the unit number and the index number to delete a license from a specific unit. For example, the FCX-ADV-LIC-SW license is installed on both stack unit 3, index 1, and stack unit 5, index 1. Because the index numbers are the same, the user must specify both the unit number and the index number to delete a license from a specific unit.

Using a trial license

NOTE

A trial license must be ordered and installed by a Brocade representative only.

A trial license enables demonstration and evaluation of a licensed feature. The trial license is valid for a period of up to 45 days, and is renewable for an additional 45 days on the second time. A licensed feature operating under a trial license has the same functionality (CLI and **show** commands) as does a licensed feature operating under a normal license.

What happens when a trial license expires

A trial license expires when it exceeds the specified expiration time or date. The countdown starts when the trial license is generated. When the license expires, the CLI commands related to the licensed feature will no longer be available from the CLI. The licensed feature will continue to run as configured until the system is reloaded, at which time the feature will be disabled and removed from the system.

NOTE

Trial licenses are not cumulative. The new license replaces the current license. To extend the license, you must contact your Brocade representative.

Console, syslog, and trap messages for trial license expiration

Three days prior to the date that a trial license is set to expire, the following warning message will appear daily on the console. On the day that the license will expire, the warning message will appear every two hours. Syslog and trap messages will also be generated.

```
SYSLOG: <12>Jan  1 00:00:00 624-top License: Package FESX624-L3U-IPV6 with LID  
NFLJMI expires in 3 days
```

When the license has expired, the following message will appear on the console. Syslog and trap messages will also be generated.

```
SYSLOG: <13>Jan  1 00:00:00 624-top License: Package FESX624-L3U-IPV6 with LID  
NFLJMI has expired
```

Renewing or extending a trial license

A trial license can be extended once by another trial license of the same type, or by a normal license of the same type. To avoid any interruptions to the network, obtain and install the second trial license before the first license expires. When extended by another trial license, the duration is not cumulative. The countdown starts when the trial license is generated.

To extend the license, you must contact your Brocade representative.

Viewing software license information from the Brocade software portal

This section describes other software licensing tasks supported from the Brocade software portal. You can use the **License Query** option to view software license information for a particular unit, transaction key, or both. You can export the report to Excel for sharing or archiving purposes.

Depending on the status of the license (for example, whether or not the license was generated), the report will include the following Information:

- Hardware part number, serial number, and description
- Software part number, serial number, and description
- Date the license was installed
- Transaction key
- LID
- Feature name
- Product line

From the **License Management** menu, select **Brocade IP/ADP > License Query**.

The **License Query** window displays. (Refer to Figure 9).

FIGURE 9 License Query window

- To view software license information for a particular unit, enter the LID in the **Unit ID** field and click **Search**.
- To view software license information for a particular transaction key, enter the unique number in the **Transaction key** field and click **Search**.

Figure 10 shows an example of the license query results.

FIGURE 10 License Query Results window

Level	Part Number	Serial Number	Installed Date	License File	Transaction Key	LID	Feature Name	Description	Product Line	C
1	NI-CER-2048C-AC	NICERAC1099				pkagtceGFOO		NI CER 2048C 1 500W AC BASE SW	NICER2048	NI
2	BR-NI-CER-2048-ADV	UA200001757	02-Feb-2010	License File	FD8D0E5B08842E13032C5D	pkagtceGFOO	BR-NI-CER-2048-ADV	BR-NI-CER-2048-ADV	CER 2048C	NI

In this example, the line items for Level 1 display hardware-related information and the line items for Level 2 display software-related information. If the query was performed before the transaction key was generated, the first row (Level 1) would not appear as part of the search results. Similarly, if the query was performed before the license was generated, some of the information in the second row would not be displayed.

Transferring a license

A license can be transferred between Brocade devices if both the following conditions are true:

- The device is under an active support contract.
- The license is being transferred between two similar models (for example, from a 24-port model to another 24-port model or from a 48-port model to another 48-port model).

NOTE

Transferring a license is only available internally for TAC, and externally for designated partners with specific accounts in the Software Portal. Contact your Brocade representative for more information.

Special replacement instructions for legacy devices

A legacy device refers to a Brocade device that was shipped prior to the introduction of software-based licensing, has an EEPROM installed, and is running pre-release 07.1.00 software.

For Brocade legacy devices in need of replacement (by way of a Return Merchandise Agreement [RMA]), the following actions must be taken:

- If the replacement device will be upgraded to a software release that supports software-based licensing, registration of the replacement device is required after the software is upgraded.
- If the replacement device will be using a software release that does *not* support software-based licensing, follow these instructions:

NOTE

This procedure is not supported on FCX and ICX platforms.

1. Prior to shipping the device in need of replacement back to the factory, remove the EEPROM from the device. To remove the EEPROM, follow the instructions in the appropriate hardware installation guide or in the instructions that shipped with the EEPROM.
2. After removing the EEPROM, store it in a safe place.
3. When the replacement device is received from the factory, install the previously removed EEPROM in the device. To do so, follow the instructions that shipped with the EEPROM.

Syslog messages and trap information

Table 36 lists the syslog messages and traps that are supported for software-based licensing.

TABLE 36 Syslog messages

Message level	Message	Explanation
Informational	License: Package <package_name> with LID <LID_number> is added	The license package has been added.
Informational	License: Package <package_name> with LID <LID_number> is removed	The license package has been deleted.
Warning	License: Package <package_name> with LID <LID_number> expires in <number> days	The trial license is about to expire. This message will begin to display three days before the expiration date, and every two hours on the last day that the license will expire.
Notification	License: Package <package_name> with LID <LID_number> has expired	The trial license has expired.

Viewing information about software licenses

This section describes the **show** commands associated with software-based licensing. These commands are issued on the Brocade device, at any level of the CLI.

NOTE

You can also view information about software licenses from the Brocade software portal. Refer to [“Viewing software license information from the Brocade software portal”](#) on page 224.

Viewing the License ID

Brocade devices that ship during and after the release of software-based licensing have the License ID (LID) imprinted on the label affixed to the device. You also can use the **show version** CLI command to view the LID on these devices, and on devices that shipped before the release of software-based licensing.

Use the **show version** command to display the serial number, software and hardware license package name, and LID of all units in the device. The following example is sample output from an FCX unit with the package FCX_ADV_ROUTER_SOFT_PACKAGE installed on units 1, 3, and 5.

```
Brocade#show version
Copyright (c) 1996-2011 Brocade Communications Systems, Inc.
  UNIT 5: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
          (6674957 bytes) from Primary FCXR07203a.bin
  UNIT 1: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
          (6674957 bytes) from Primary FCXR07203a.bin
  UNIT 2: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
          (6674957 bytes) from Primary FCXR07203a.bin
  UNIT 3: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
          (6674957 bytes) from Primary FCXR07203a.bin
  UNIT 4: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
          (6674957 bytes) from Primary FCXR07203a.bin
Boot-Monitor Image size = 369286, Version:07.0.01T7f5 (grz07001)
```

Viewing information about software licenses

```
HW: Stackable FCX648S
=====
UNIT 1: SL 1: FCX-48GS POE 48-port Management Module
      Serial #: BCY2253E0PM
      License: FCX_ADV_ROUTER_SOFT_PACKAGE (LID: deaHHKIgFro)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
      PROM-TYPE: FCX-ADV-U
=====
UNIT 1: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 1: SL 3: FCX-2XG 2-port 10G Module (2-XFP)
=====
UNIT 2: SL 1: FCX-48GS POE 48-port Management Module
      Serial #: upgrade7072
      License: FCX_FULL_ROUTER_SOFT_PACKAGE (LID: ZU0W478MFMH)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
=====
UNIT 2: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 3: SL 1: FCX-48GS POE 48-port Management Module
      Serial #: BCY2253E0P8
      License: FCX_ADV_ROUTER_SOFT_PACKAGE (LID: deaHHKIgFrN)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
=====
UNIT 3: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 4: SL 1: FCX-24GS 24-port Management Module
      Serial #: BCV2233E00B
      License: FCX_FULL_ROUTER_SOFT_PACKAGE (LID: dexHHIIgFFd)
      P-ENGINE 0: type DB90, rev 01
=====
UNIT 4: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 5: SL 1: FCX-48GS 48-port Management Module
      Serial #: UPGRADE7072
      License: FCX_ADV_ROUTER_SOFT_PACKAGE (LID: writcfgMFMH)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
=====
```

Syntax: show version

In the **show license** command output, only unit 3 and unit 5 are shown with the software license, FCX-ADV-LIC-SW. Unit 1 is not displayed in the **show license** command output because it has a hardware license installed on the device as indicated by the PROM-TYPE: FCX-ADV-U. For more information about the **show license** command, refer to [“Viewing the license database”](#) on page 229.

```

Brocade#show license
Index      License Name      Lid      License Type      Status      License Period      License
Capacity
Stack unit 3:
1          FCX-ADV-LIC-SW      deaHHKigFrN Normal          Active      Unlimited          1
Stack unit 4:
1          FCX-ADV-LIC-SW      dexHHIlgFFd Normal          Active      Unlimited          1
Stack unit 5:
1          FCX-ADV-LIC-SW      writcfgMFMH Normal          Active      Unlimited          1

```

Syntax: show license

Viewing the license database

NOTE

The **show license** command can be used to display software license information for the FESX, SX 800 and SX 1600, FCX, ICX 6610, FWS, and ICX 6450 devices.

To display general information about all software licenses for all units in a device, use the **show license** command. The **show license** command only displays software license information for a unit, not hardware license information, as shown in the following example.

```

Brocade#show license
Index      License Name Lid      License Type      Status License Period      License Capacity
Stack unit 3:
1 FCX-ADV-LIC-SW deaHHKigFrN Normal          Active      Unlimited          1
Stack unit 5:
1 FCX-ADV-LIC-SW writcfgMFMH Normal          Active      Unlimited          1

```

To display software license information on an ICX 6610 device (for example, the ICX 6610 premium and advance licenses) enter the following command.

```

Brocade#show license
Index      License Name      Lid      License Type      Status      License Period      License Capacity
Stack unit 1:
1          ICX6610-PREM-LIC-SW      FJdnjFJFGiF Normal          Active      Unlimited          1
2          ICX6610-10G-LIC-POD      FJdnjFJFGiF Normal          Active      Unlimited          8
Stack unit 2:
1          ICX6610-ADV-LIC-SW      FJdnmFJFGiF Normal          Active      Unlimited          1
2          ICX6610-10G-LIC-POD      FJdnjFJFGiF Normal          Invalid     Unlimited          8
Stack unit 3:
1          ICX6610-ADV-LIC-SW      FJdnlFJFGiF Normal          Active      Unlimited          1
4          ICX6610-10G-LIC-POD      FJdnlFJFGiF Normal          Active      Unlimited          8

```

Syntax: show license

Viewing information about software licenses

To display software license information on an ICX 6430 device, enter the following command. In the example below, the premium and PoD licenses are installed on stack unit 1, and on stack unit 2 only the premium license is installed.

```
Brocade#show license
Index      License Name      Lid      License Type  Status  License Period  License Capacity
Stack unit 1:
1          ICX6450-PREM-LIC-SW  dbtFJIKiFFI Normal      Active  Unlimited      2
2          ICX6450-10G-LIC-POD  dbtFJIKiFFI Normal      Active  Unlimited      2
Stack unit 2:
1          ICX6450-PREM-LIC-SW  dbtFJIKiFFI Normal      Active  Unlimited      2
```

Syntax: show license

To display software license information on an FWS device, enter the following command.

NOTE

The output from the **show license** command is the same for an SX 800, SX 1600, FWS, and a FESX device.

```
Brocade#show license
Index      Package Name      Lid      License Type  Status  License
Period
1          FWS624-EL3U        cpFNJHFFGO normal      active  unlimited
```

Syntax: show license

To display specific software license information installed on a SX 800, SX 1600, FWS, or a FESX device, enter the following command.

```
Brocade#show license 1
License information for license <1>:
+package name:      FWS624-EL3U
+lid:               cpFNJHFFGO
+license type:      normal
+status:            active
+license period:    unlimited
```

Syntax: show license <index_number>

The <index_number> variable specifies the specific license file installed on the device. The **unit** <unit_id> parameter is not applicable on a SX 800, SX 1600, FWS, or a FESX device.

To display software license information for a specific stack unit on an ICX 6610, ICX 6450, or a FCX device, enter the following command. In the output below, the ICX 6610 premium license, and the POD license are installed on unit 3.

```
Brocade#show license unit 3
Index      License Name      Lid      License Type  Status  License Period  License Capacity
Stack unit 3:
1          ICX6610-PREM-LIC-SW  FJdnjFJFGiF Normal      Active  Unlimited      1
2          ICX6610-10G-LIC-POD  FJdnjFJFGiF Normal      Active  Unlimited      8
```

Syntax: `show license [unit <unit_id>]`

The **unit** <unit_id> parameter specifies the unit ID number. The unit ID number is available only on FCX, ICX 6610, and ICX 6450 devices.

[Table 37](#) describes the information displayed by the **show license unit** <unit_id> command

TABLE 37 Output from the **show license unit** command

Field	Description
Index	The index number specifies the software license file for a specific stack. The index number is generated by the member unit.
License Name	The name of license installed for the license index number on the stack unit.
Lid	The license ID. This number is embedded in the Brocade device.
License Type	Indicates whether the license is normal (permanent) or trial (temporary).
Status	Indicates the status of the license: <ul style="list-style-type: none"> • Valid – A license is valid if the LID matches the serial number of the device for which the license was purchased, and the package name is recognized by the system. • Invalid – The LID does not match the serial number of the device for which the license was purchased. • Active – The license is valid and in effect on the device. • Not used – The license is not in effect on the device. • Expired – For trial licenses only, this indicates that the trial license has expired.
License Period	If the license type is trial (temporary), this field displays the number of days the license is valid. If the license type is normal (permanent), this field displays Unlimited.
License capacity	The port capacity of the PoD license. For ICX 6610 devices, the PoD license can be a 4 port capacity license, or a 8 port capacity license depending on the number of licenses purchased. For ICX 6450 devices, the field displays license capacity 2 when the PoD license is purchased for two ports (ports two and four). The license capacity field is displayed in the show license command outputs for ICX 6610, ICX 6450, and FCX devices only.

To display detailed information about a license for a specific unit, use the **show license unit** <unit_id> [**index** <index_number>] command. The following example shows a sample output.

```
Brocade#show license unit 3 index 1
License information for unit 3 license <1>:
+license name:      FCX-ADV-LIC-SW
+lid:               deaHHKIGFrN
+license type:      Normal
+status:            Active
+license period:    Unlimited
Trial license information:
+days used:        10
+hours used:        21
+days left:        30
+hours left:        18
```

Syntax: `show license unit <unit_id> [index <index_number>]`

The **index** *<license_index>* parameter specifies the software license file that you want to display information for. The **index** *<index_number>* option is available only on FCX, ICX 6610, and ICX 6450 devices.

[Table 38](#) describes the information displayed by the **show license unit** *<unit_id>* [**index** *<index_number>*] command.

TABLE 38 Output from the **show license** command

Field	Description
+license name	The name of the license installed on the unit.
+lid	The license ID. This number is embedded in the Brocade device.
+license type	Indicates whether the license is normal (permanent) or trial (temporary).
+status	Indicates the status of the license: <ul style="list-style-type: none"> • Valid – A license is valid if the LID matches the serial number of the device for which the license was purchased, and the package name is recognized by the system. • Invalid – The LID does not match the serial number of the device for which the license was purchased. • Active – The license is valid and in effect on the device. • Not used – The license is not in effect on the device. • Expired – For trial licenses only, this indicates that the trial license has expired.
+license period	If the license type is trial (temporary), this field displays the number of days the license is valid. If the license type is normal (permanent), this field displays Unlimited.
+ days used	The number of days the trial license has been in effect.
+ hours used	The number of hours the trial license has been in effect.
+ days left	The number of days left before the trial license expires.
+ hours left	The number of hours left before the trial license expires.

Viewing software packages installed in the device

Use the **show version** command to view the software packages that are currently running in the device. The following example is sample output from an FCX unit with the package FCX_ADV_ROUTER_SOFT_PACKAGE installed on units 1, 3, and 5.

NOTE

The software package name is not the same as the license name.

```
Brocade#show version
Copyright (c) 1996-2011 Brocade Communications Systems, Inc.
    UNIT 5: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
            (6674957 bytes) from Primary FCXR07203a.bin
UNIT 1: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
            (6674957 bytes) from Primary FCXR07203a.bin
UNIT 2: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
            (6674957 bytes) from Primary FCXR07203a.bin
UNIT 3: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
            (6674957 bytes) from Primary FCXR07203a.bin
UNIT 4: compiled on Jun 24 2011 at 14:20:38 labeled as FCXR07203a
            (6674957 bytes) from Primary FCXR07203a.bin
```

```

Boot-Monitor Image size = 369286, Version:07.0.01T7f5 (grz07001)
HW: Stackable FCX648S
=====
UNIT 1: SL 1: FCX-48GS POE 48-port Management Module
      Serial #: BCY2253E0PM
      License: FCX_ADV_ROUTER_SOFT_PACKAGE (LID: deaHHKIgFro)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
      PROM-TYPE: FCX-ADV-U
=====
UNIT 1: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 1: SL 3: FCX-2XG 2-port 10G Module (2-XFP)
=====
UNIT 2: SL 1: FCX-48GS POE 48-port Management Module
      Serial #: upgrade7072
      License: FCX_FULL_ROUTER_SOFT_PACKAGE (LID: ZU0W478MFMH)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
=====
UNIT 2: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 3: SL 1: FCX-48GS POE 48-port Management Module
      Serial #: BCY2253E0P8
      License: FCX_ADV_ROUTER_SOFT_PACKAGE (LID: deaHHKIgFrN)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
=====
UNIT 3: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 4: SL 1: FCX-24GS 24-port Management Module
      Serial #: BCV2233E00B
      License: FCX_FULL_ROUTER_SOFT_PACKAGE (LID: dexHHIIGFFd)
      P-ENGINE 0: type DB90, rev 01
=====
UNIT 4: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 5: SL 1: FCX-48GS 48-port Management Module
      Serial #: UPGRADE7072
      License: FCX_ADV_ROUTER_SOFT_PACKAGE (LID: writcfgMFMH)
      P-ENGINE 0: type DB90, rev 01
      P-ENGINE 1: type DB90, rev 01
=====

```

Syntax: show version

For a list of supported software packages installed on the device, refer to [Table 33](#) on page 206.

Viewing information about software licenses

Brocade Stackable Devices

Table 39 lists the individual Brocade FastIron switches and the IronStack features they support. These features are supported only on FastIron stackable devices, and are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 39 Supported IronStack features

Feature	FESX FSX 800 FSX 1600	FWS	FCX ¹	ICX 6610 ²	ICX 6430 ICX 6450
Building an IronStack	No	No	Yes	Yes	Yes
<ul style="list-style-type: none"> Secure-setup Automatic configuration Manual configuration 					
IronStack management	No	No	Yes	Yes	Yes
IronStack management MAC address	No	No	Yes	Yes	Yes
IronStack partitioning	No	No	Yes	Yes	Yes
Persistent MAC address	No	No	Yes	Yes	Yes
IronStack software upgrade	No	No	Yes	Yes	Yes
IronStack and stack mismatch troubleshooting	No	No	Yes	Yes	Yes
Hitless stacking:	No	No	Yes	Yes	Yes
<ul style="list-style-type: none"> Hitless failover Hitless switchover 					
Trunking of stacked ports	No	No	No	Yes	Yes
Auto Image Copy for stack units	No	No	Yes	Yes	Yes

1. All FCX models can be ordered from the factory as -ADV models. ADV models include support for Layer 3 BGP. FCX-E and FCX-I models require an optional 10 Gbps SFP+ module to support stacking.

2. All ICX 6610 models can be ordered from the factory as -ADV models. ADV models include support for Layer 3 BGP.

Brocade IronStack overview

This section gives a brief overview of IronStack technology, including IronStack terminology. This section also lists the FastIron models that support stacking.

Brocade IronStack features

A stack is a group of devices that are connected so that they operate as a single chassis. Brocade IronStack technology features include:

- Management by a single IP address
- Support for up to eight units per stack. ICX 6430 supports only up to four units in the stack
- Flexible stacking ports
- Linear and ring stack topology support
- Secure-setup utility to make stack setup easy and secure
- Active Controller, Standby Controller, and member units in a stack
- Active Controller management of entire stack
- Active Controller download of software images to all stack units
- Standby Controller for stack redundancy
- Active Controller maintenance of information database for all stack units
- Packet switching in hardware between ports on stack units
- All protocols operate on an IronStack in the same way as on a chassis system

Brocade stackable models

FCX devices

All FCX devices can be active members of a Brocade IronStack. FCX-E and FCX-I models require an optional 10 Gbps SFP+ module to support stacking. For information about how to install FCX devices, refer to the *Brocade FastIron CX Hardware Installation Guide*.

All FCX devices can be ordered from the factory as -ADV models with support for Layer 3 BGP.

ICX devices

All ICX 6610, ICX 6430, and ICX 6450 devices can be active members of a Brocade IronStack. For information about how to install ICX 6610 devices, refer to the *Brocade ICX 6610 Series Hardware Installation Guide*. For information about how to install ICX 6430 and ICX 6450 devices, refer to the *Brocade ICX 6450 and ICX 6430 Hardware Installation Guide*.

ICX devices also support trunked ports. For ICX 6610 devices, refer to [“ICX 6610 stack topologies”](#) on page 242 for details. For ICX 6450 and ICX 6430 devices, refer to [“Connecting ICX 6450 and ICX 6430 devices in a stack”](#) on page 244.

All ICX 6610 devices can be ordered from the factory as -ADV models with support for Layer 3 BGP.

Brocade IronStack terminology

Stack unit roles

- **Active Controller** - Handles stack management and configures all system- and interface-level features.
 - **Future Active Controller** - The unit that will take over as Active Controller after the next reload, if its priority has been changed to the highest priority. When a priority for a stack unit is changed to be higher than the existing Active Controller, the takeover does not happen immediately to prevent disruptions in the stack operation.
- **Standby Controller** - The stack member with the highest priority after the Active Controller. The Standby Controller takes over if the current Active Controller fails.
- **Stack Member** - A unit functioning in the stack in a capacity other than Active or Standby Controller.
- **Stack Unit** - Any device functioning within the stack, including the Active Controller and Standby Controller.
 - **Upstream Stack Unit** - An upstream unit is connected to the first stacking port on the Active Controller. (The left-hand port as you face the stacking ports.) Refer to [Figure 14](#) and [Figure 15](#).
 - **Downstream Stack Unit** - A downstream unit is connected to the second stacking port on the Active Controller. (The right-hand port as you face the stacking ports.) Refer to [Figure 14](#) and [Figure 15](#).

General IronStack terminology

- **Bootup Role** - the role a unit takes during the boot sequence. This role can be standalone, Active Controller, Standby Controller, or stack member. The Active Controller or a standalone unit can access the full range of the CLI. Until a stack is formed, the local consoles on the Standby Controller and stack members provide access to a limited form of the CLI, such as the show, stack, and a few debug commands. When the stack is formed, all local consoles are directed to the Active Controller, which can access the entire CLI. The last line of output from the **show version** command indicates the role of a unit, unless it is a standalone unit, in which case it is not shown. For example:

```
My stack unit ID = 1, bootup role = active
```

- **Clean Unit** - A unit that contains no startup flash configuration or run time configuration. To erase old configuration information, enter the **erase startup-config** command and reset the unit. For FCX devices, the run-time configuration on a clean unit may also contain default-port information,
- **Control Path** - A path across stacking links dedicated to carrying control traffic such as commands to program hardware or software image data for upgrades. A stack unit must join the control path to operate fully in the stack.
- **Default Port** - FCX devices use the **default-port** command to define stacking port candidates.
- **Interprocessor Communications (IPC)** - The process by which proprietary packets are exchanged between stack unit CPUs.

- **IronStack** - A set of Brocade stackable units (maximum of eight) and their connected stacking links so that: all units can be accessed through their common connections, a single unit can manage the entire stack, and configurable entities, such as VLANs and trunk groups, can have members on multiple stack units.
- **Non-Functioning Stack Unit** - A stack unit that is recognized as a stack member, and is communicating with the Active Controller over the Control Path, but is in a non-functioning state. Because of this state, traffic from the non-stack ports will not be forwarded into the stack - they will be dropped or discarded. This may be caused by an image or configuration mismatch.
- **Sequential Connection** - Stack unit IDs, beginning with the Active Controller, are sequential. For example, 1, 3, 4, 6, 7 is sequential if Active Controller is 1. 1, 7, 6, 4, 3 are non-sequential in a linear topology, but become sequential in a ring topology when counted from the other direction as: 1, 3, 4, 6, 7. Gaps in numbering are allowed.
- **Standalone Unit** - A unit that is not enabled for stacking, or an Active Controller without any Standby Controller or stack members.
- **Stacking Link** - A cable that connects a stacking port on one unit to a stacking port on another unit.
- **Secure Setup** - A software utility that establishes a secure stack.
- **Unit Replacement** - The process of swapping out a unit with a Clean Unit. No configuration change is required. Refer to [“Adding, removing, or replacing units in an IronStack”](#) on page 311.
- **Reserved / Provisional Unit** - A unit configuration number that has no physical unit associated with it.
- **Trunked Stacking Port (Trunk)** - A trunk consists of multiple stacking ports and is treated as one logical link. It provides more bandwidth and better resilience.
- **Stack Path** - A data path formed across the stacking links to determine the set of stack members that are present in the stack topology, and their locations in the stack.
- **Stacking Port** - A physical interface on a stack unit that connects a stacking link. Stacking ports are point-to-point links that exchange proprietary packets. Stacking ports must be 10 Gbps Ethernet ports, and cannot be configured for any other purpose while operating as stacking ports. Brocade stacking units contain two ports that can be stacking ports. However, the flexible stacking port feature also allows you to use one port as a stacking port and the other port as a regular data port. Refer to [“Controlling stack topology”](#) on page 283.
- **Stack Slot** - A slot in a stack is synonymous with *line model* in a chassis. [Table 40](#) shows the port and slot designations for FastIron stackable devices.
- **Stack Topology** - A contiguously-connected set of stack units in an IronStack that are currently communicating with each other. All units that are present in the stack topology appear in output from the **show stack** command.
- **Static Configuration** - A configuration that remains in the database of the Active Controller even if the unit it refers to is removed from the stack. Static configurations are derived from the startup configuration file during the boot sequence, are manually entered, or are converted from dynamic configurations after a **write memory** command is issued.
- **Dynamic Configuration** - A unit configuration that is dynamically learned by a new stack unit from the Active Controller. A dynamic configuration disappears when the unit leaves the stack.

Supported IronStack topologies

This section describes how to build an IronStack. Before you begin, you should be familiar with the supported stack topologies and the software requirements. When you are ready to build your stack, you can go directly to the instructions.

Brocade IronStack topologies

Brocade IronStack technology supports linear and ring stack topologies. Although Brocade stackable units may be connected in a simple linear topology, Brocade recommends a ring topology because it offers the best redundancy and the most resilient operation.

Mixed unit topologies

FCX, ICX 6610, ICX 6450 and ICX 6430 devices cannot be combined in a stack with other devices. Stacks must contain devices of the same type. For more information about FCX stack topologies, refer to [“FCX stack topologies”](#) on page 239. For more information about stack topologies for ICX 6430 and ICX 6450 devices, see [“Configuring an ICX 6430 and ICX 6450 IronStack”](#) on page 266.

FCX stack topologies

A Brocade IronStack can contain all one model, or any combination of the FCX models. You can mix 24-port and 48-port FCX devices in a single stack, to a maximum of eight units per stack.

NOTE

FCX devices cannot be combined in a stack with non-FCX devices.

The procedure for cabling a stack of FCX devices differs depending on whether your stack contains FCX-E and FCX-I devices. [Figure 11](#) shows FCX-S and FCXS-F devices cabled in linear and ring stack topologies. Note that these devices are cabled from the rear panel. [Figure 12](#) shows FCX-E devices in a ring topology stack. [Figure 13](#) shows FCX-E devices in a linear topology stack.

Figure 14 shows a mixed linear topology stack of FCX-S, FCXS-F, and FCX-E or FCX-I devices. Because the FCX-E and FCX-I devices are cabled from the front panel, and FCX-S and FCXS-F devices are cabled from the rear panel by default, you need to reconfigure the default stacking ports on FCX-S or FCXS-F devices to the ports on the front panel. For more information about reconfiguring default stacking ports, refer to [“Configuring default ports on FCX devices”](#) on page 261.

FIGURE 11 FCX linear and ring stack topologies

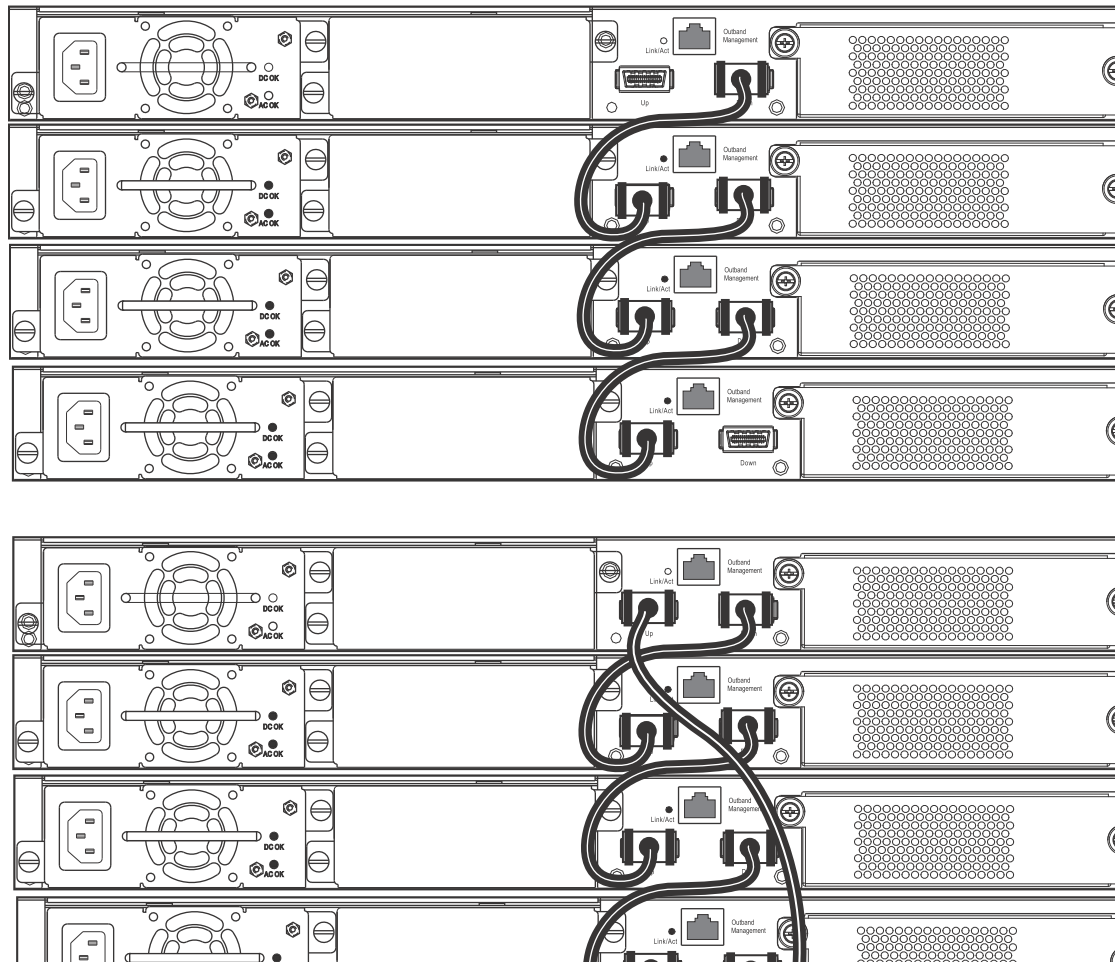
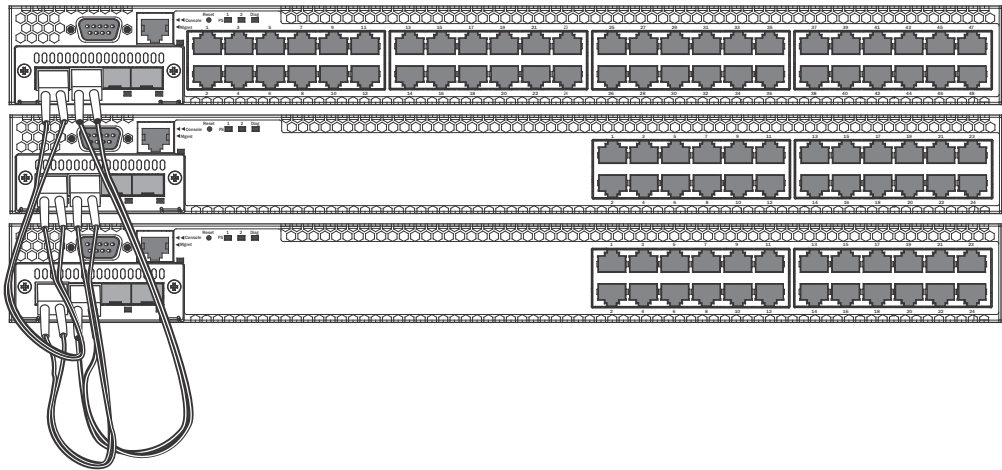
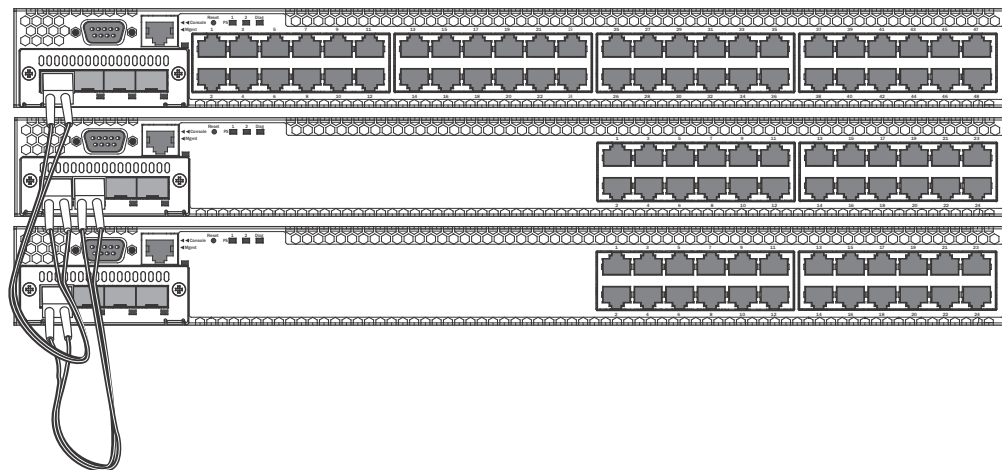
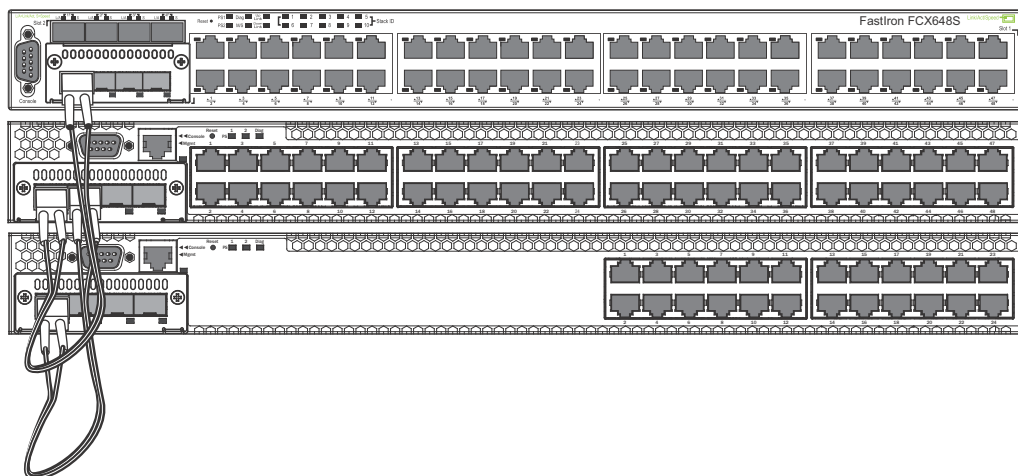


FIGURE 12 FCX-E ring topology stack using SFP+ module ports**FIGURE 13** FCX-E linear topology stack using SFP+ module ports**FIGURE 14** Mixed linear stack of FCX-E devices and FCX-S devices

ICX 6610 stack topologies

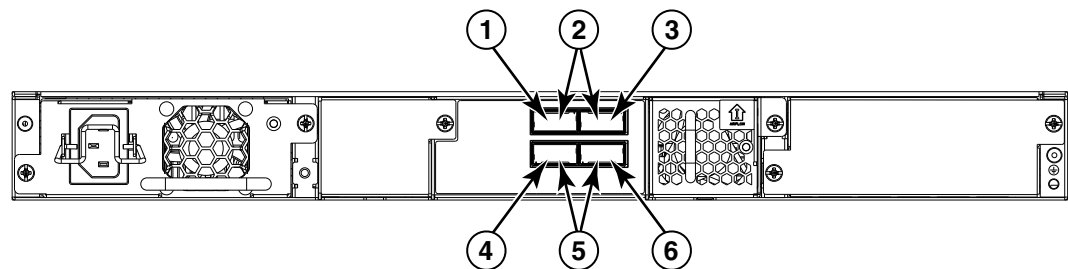
In contrast to earlier generations of switches, ICX 6610 devices have four ports on their back panels that are exclusively dedicated to stacking. These cannot be used as data ports, even when stacking is not enabled.

These are the two 40-Gbps ports and two 4 x 10-Gbps ports arranged in two rows. The top row consists of port 1/2/1 and 1/2/2, and the bottom row has 1/2/6 and 1/2/7. Here ports 1/2/1 and 1/2/6 are 40G ports. Ports 1/2/2 and 1/2/7 are 4*10G ports. Refer to [Figure 15](#).

You can create a stacking trunk by connecting one port of each type to ports of the same type on another switch in the stack.

When creating a trunk, the two ports in the same row are always trunked, one 40G and one 4*10G and they must connect to ports in the same row of another device. For example, one device has a trunk consisting of 40-Gbps port 1/2/1 and a 4 x 10-Gbps port 1/2/2 on the top row, and one trunk with the same pair type, 1/2/6 and 1/2/7, on the bottom row. If this device is being connected to the next one in sequence, one of these rows must connect to 2/2/1 and 2/2/2, OR to 2/2/6 and 2/2/7.

FIGURE 15 Trunked stack port configuration on Brocade ICX 6610 devices



- (1) Stack port n/2/1 (40G)
- (2) Trunk 0
- (3) Stack port n/2/2 (4*10G)
- (4) Stack port n/2/6 (40G)
- (5) Trunk 1
- (6) Stack port n/2/7 (4*10G)

The two stacking trunks can form either linear or ring topologies. [Figure 16](#) and [Figure 17](#) show fully connected linear and ring topologies. Both cables of each trunk are connected.

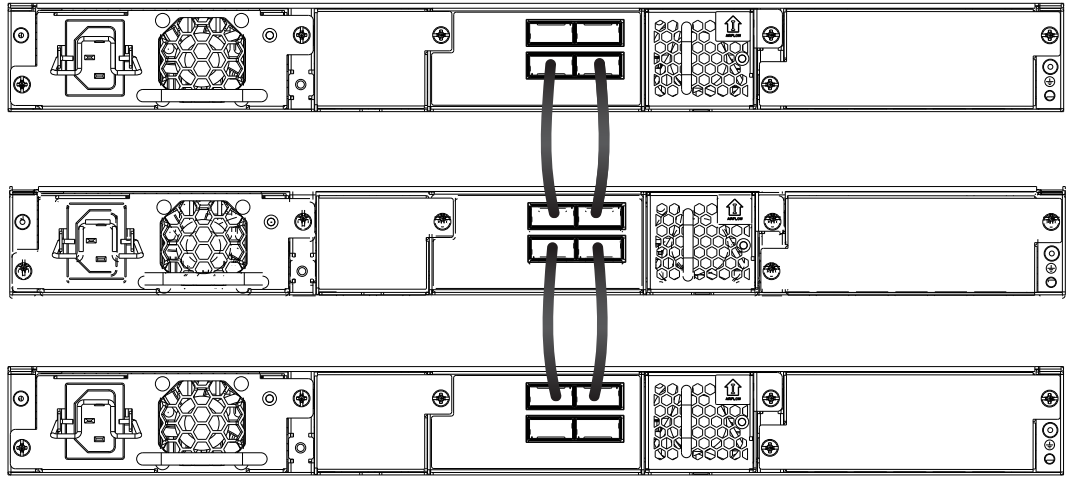
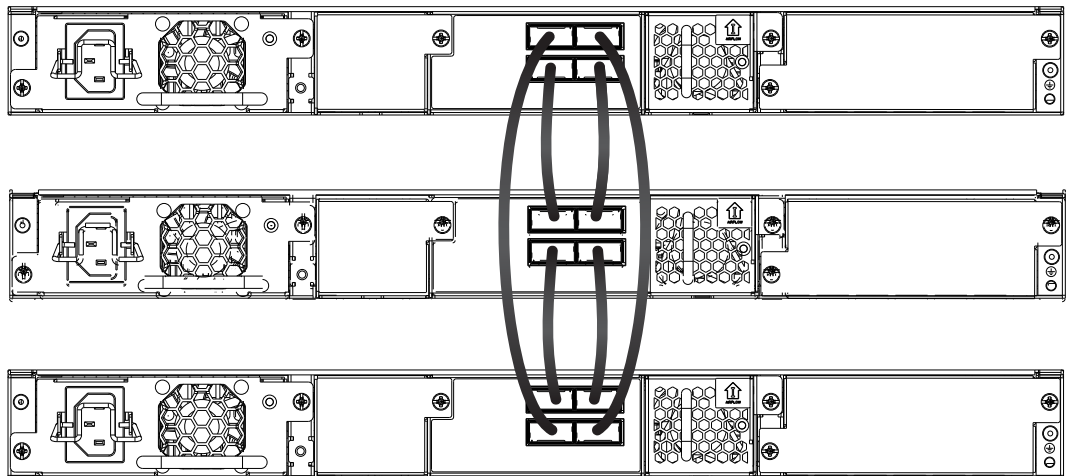
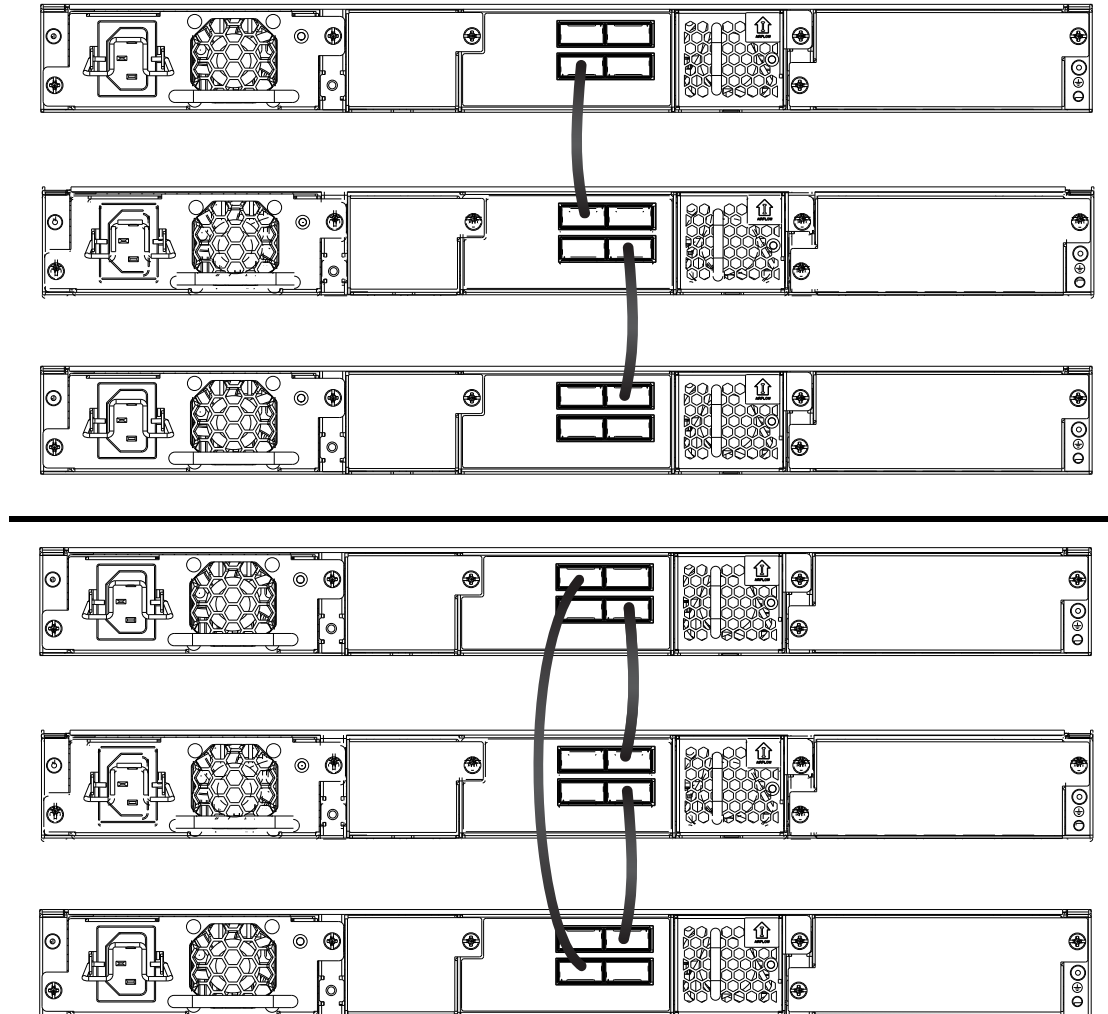
FIGURE 16 ICX 6610 linear stack topology**FIGURE 17** ICX 6610 ring stack topology

Figure 18 shows a linear and ring topology with missing partial trunk cables. They both still work.

FIGURE 18 ICX 6610 linear and ring stack topologies with partially missing cables



Connecting ICX 6450 and ICX 6430 devices in a stack

ICX 6430 and ICX 6450 support linear and ring stack topologies, and can also operate as standalone devices. ICX 6430 and ICX 6450 devices have four ports on the front panel for a stacking configuration. ICX 6430 and ICX 6450 devices ship with two default stacking ports configured. When stacking is enabled, ports 1 and 3 are dedicated to stacking and cannot be used for data ports. Use the **stack-port** command to select only one of these default ports as the stacking port. If you do not select a default port as the stacking port, both default ports will operate as stacking ports. For more information about using the **stack-port** command on ICX 6430 and ICX 6450 devices, refer to [“Error messages encountered during the configuration of an ICX 6430 or ICX 6450 IronStack”](#) on page 269. If stacking is not enabled on the ports, then all four stacking ports

can be used for data or uplink ports. By default, ICX 6430 and ICX 6450 devices are not configured for trunked stacking. You can dynamically configure or remove a stacking trunk port configuration using the **stack-trunk** command or the **multi-trunk** command. For more information about these commands, refer to [“Configuring an ICX 6430 and ICX 6450 IronStack”](#) on page 266.

ICX 6450 and ICX 6430 devices support hitless stacking switchover and failover. The Secure Setup utility is supported for ICX 6450 and ICX 6430 devices.

ICX 6430 device supports up to four units in a stack. ICX 6450 device supports up to eight units in a stack. You cannot mix and match stacking units between ICX 6430 and ICX 6450 devices.

Connecting ICX 6450 devices in a stack

ICX 6450 devices have 24 or 48 10/100/100 Mbps data ports and 4 ports with 1-Gbps or 10-Gbps SFP+ fiber uplink and stacking ports. The top row consists of ports 1 and 3, and the bottom row consists of ports 2 and 4. By default, ports 1 and 3 are 10-Gbps stacking ports. By default, without a license at bootup, ports 2 and 4 come up in 10-Gbps port speed in an error disabled state. To enable ports 2 and 4 to 10-Gbps port speed, purchase the ICX6450-2X10G-LIC-POD license. For more information about enabling ports 2 and 4 to 10-Gbps port speed, refer to [“Licensing for Ports on Demand”](#) on page 208.

Connecting ICX 6430 devices in a stack

ICX 6430 devices have 24 or 48 10/100/100 Mbps data ports and 4 1-Gbps SFP fiber uplink and stacking ports. The default stacking ports are 1 and 3. The 4 1-Gbps ports are not eligible for an upgrade to 10-Gbps port speed. Trunk stacking configuration is supported for ICX 6430 devices. LAG configuration is supported for stacking or uplink ports when a pair of 1-Gbps ports are aggregated.

Trunking configuration considerations for ICX 6430 and ICX 6450 devices

The ICX 6430 and ICX 6450 stacking ports are grouped into two trunks. Follow these guidelines for connecting and configuring the stacking ports:

NOTE

After enabling the ICX 6430 and ICX 6450 trunked stacking ports, it is recommended that you enter the **write memory** command to save your configuration.

- You can connect one or both ports in a trunk. Connecting both ports in a trunk increases stacking bandwidth and provides resiliency.
- You must enable stacking and connect cables properly for the stack to work. The active copper cable lengths for 1-Gbps ports are 1 m (3.2 ft) and 5 m (16.4 ft). The copper cable lengths for 10-Gbps ports are 1 m (3.2 ft), 3 m (9.8 ft), and 5 m (16.4 ft).
- The default stacking ports are always ports 1 and 3. You can trunk (or un-trunk) ports 1 to 2 and or ports 3 to 4. One or both of the two sets of stacking ports can be trunked (or un-trunked).
- When creating a trunk, the ports in the same column are always trunked. For ICX 6450 devices, all stack ports must be configured to 10-Gbps port speed to enable trunking. For ICX 6430 devices, all stack ports must be at 1-Gbps port speed to enable trunking.

For example, you can connect ports 1/2/3 to 1/2/4 to form one trunk on one device, and ports 2/2/1 to 2/2/2 to form a second trunk on another device.

- If you connect both ports in a trunk, both ports must connect to both ports of one trunk on another device.

ICX 6430 and ICX 6450 stack topologies

In a linear stack topology, there is a single stack cable connection between each switch that carries two-way communications across the stack. In a ring stack topology, an extra cable is connected between the top and bottom switches forming a “ring” or “closed-loop.” The closed-loop cable provides a redundant path for the stack link, so if one link fails, stack communications can be maintained.

The following figures show stacking cabling configurations.

[Figure 19](#) shows an example of how the stack cables can be connected between switches if you only connect one port per trunk. The figure shows both linear and ring stacking configurations. The one port per trunk topology is the most commonly configured stacking configuration and is applicable to both ICX 6430 and ICX 6450 switches.

[Figure 20](#) shows how the stack cables are connected between switches in a linear stacking configuration for dual linking. The linear stacking configuration is applicable to both ICX 6430 and ICX 6450 devices.

[Figure 21](#) shows how the stack cables are connected between switches in a ring stacking configuration. The ring stacking configuration is applicable to both ICX 6430 and ICX 6450 devices.

FIGURE 19 ICX 6430 and ICX 6450 stacking with one port per trunk

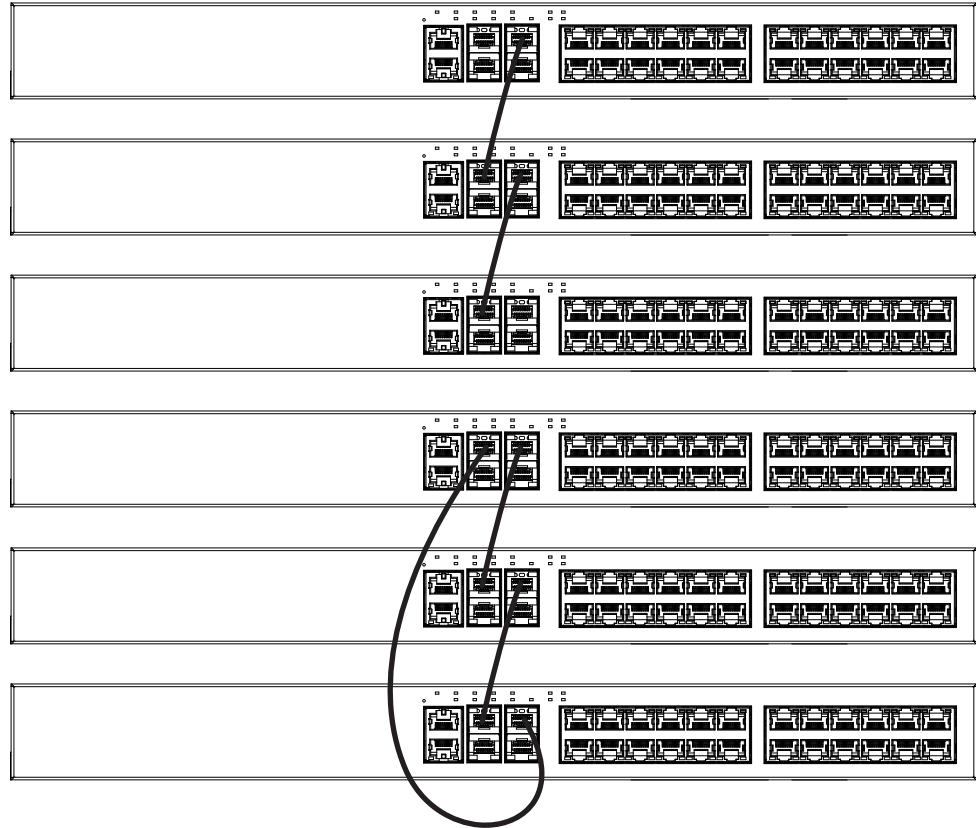


FIGURE 20 ICX 6430 and ICX 6450 linear stacking configuration

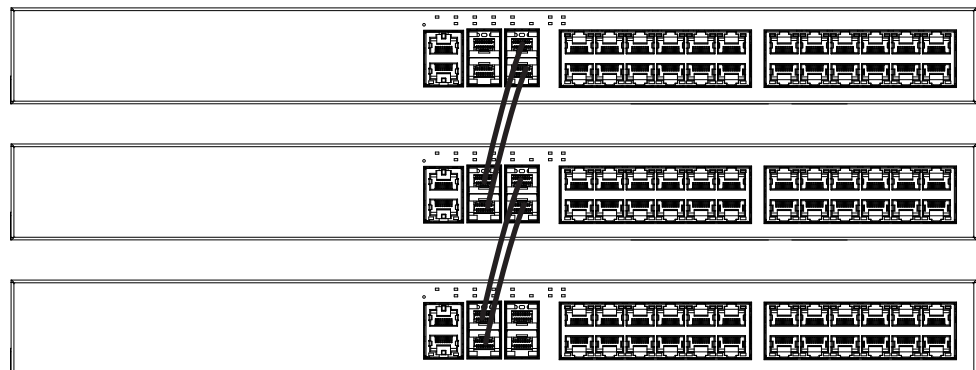
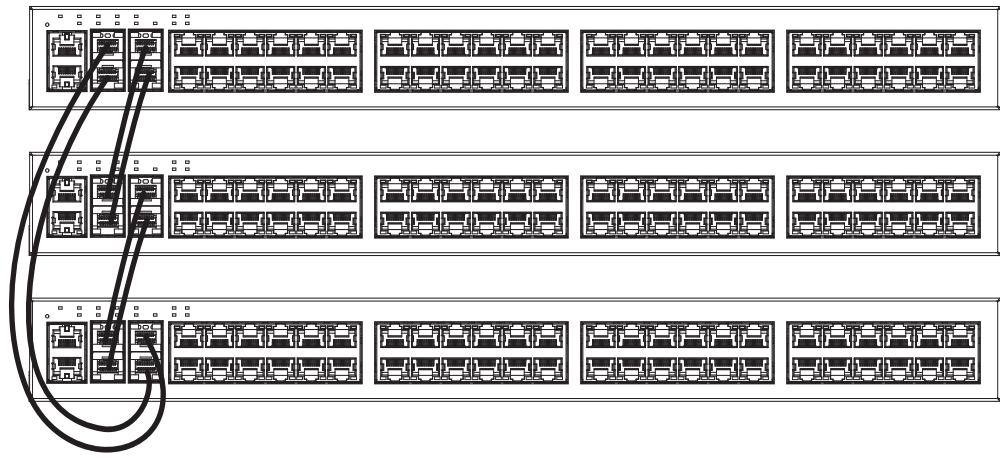


FIGURE 21 ICX 6430 and ICX 6450 ring stacking configuration



Software requirements

All units in an IronStack must be running the same software version. Refer to [“IronStack troubleshooting”](#) on page 315 for more information.

IronStack construction methods

There are three ways to build an IronStack.

1. Use the **secure-setup** utility to form your stack. Secure-setup gives you control over the design of your stack topology and provides security through password verification. For the secure-setup procedure, refer to [“Scenario 1 - Three-member IronStack in a ring topology using secure-setup”](#) on page 250.
2. Automatic stack configuration. With this method, you enter all configuration information, including the module type and the priorities of all members into the unit you decide will be the Active Controller and set its priority to be the highest. When you enable stacking on the Active Controller the stack then forms automatically. This method requires that you start with clean units (except for the Active Controller) that do not contain startup or run time configurations. Refer to [“Scenario 2 - Three-member IronStack in a ring topology using the automatic setup process”](#) on page 254.
3. Manual stack configuration. With this method, you configure every unit individually, and enable stacking on each unit. Once the units are connected together, they will automatically operate as an IronStack. With this method the unit with the highest priority becomes the Active Controller, and ID assignment is determined by the sequence in which you physically connect the units. Refer to [“Scenario 3 - Three member IronStack in a ring topology using the manual configuration process”](#) on page 257.

IronStack configuration notes

Before you configure your IronStack, consider the following guidelines:

- Consider the number of units and how the stacking ports on the units will be connected. For more information, refer to the *Brocade FCX Hardware Installation Guide*.
- The stack should be physically cabled in a linear or ring topology. Connect only those units that will be active in the stack.
- Make sure all units intended for the stack are running the same software version. Refer to [“Confirming IronStack software versions”](#) on page 282.
- When you have a full stack of 8 units, you may need to increase the trap hold time from the default, which is 60 seconds, to five minutes (300 seconds). This will prevent the loss of initial boot traps. To increase the trap hold time, use the following command.

```
Brocade(config)#snmp-server enable traps holddown-time 300
```

Syntax: [no] snmp-server enable traps holddown-time <seconds>

NOTE

The router image requires more time to boot than the switch image.

Scenario 1 - Three-member IronStack in a ring topology using secure-setup

NOTE

For more detailed information about configuring an FCX IronStack, refer to [“FCX IronStack configuration”](#) on page 258.

This scenario describes how to build an IronStack using the secure-setup utility. Secure-setup lets you easily configure your entire stack through the Active Controller, which propagates the configuration to all stack members. Secure-setup is the most secure way to build an IronStack, and gives you the most control over how your stack is built. For example, secure-setup offers three security features that prevent unauthorized devices from accessing or joining an IronStack:

- Authentication of secure-setup packets provides verification that these packets are from genuine Brocade stack unit. MD5-based port verification confirms stacking ports.
- Superuser password is required to allow password-protected devices to become members of an IronStack.
- The **stack disable** command. When this command is issued, a unit does not listen for or send stacking packets, which means that no other device in the network can force the stacking-disabled unit to join an IronStack.

Secure-setup can also be used to add units to an existing IronStack (refer to [“Adding, removing, or replacing units in an IronStack”](#) on page 311) and to change the stack IDs of stack members (refer to [“IronStack unit identification”](#) on page 277).

When secure-setup is issued on a unit that is not already the Active Controller, this unit becomes the Active Controller, and, if it does not have an assigned priority, secure-setup assigns it a priority of 128. Any unit that then tries to join the stack must have an assigned priority less than 128. If secure-setup discovers a unit with a priority of 128 or higher, it changes the priority to 118.

When secure-setup is issued on a unit that is not already the Active Controller, this unit becomes the Active Controller. If this unit does not already have an assigned priority, secure-setup will assign this unit a priority of 128 by default, if no other units in the stack have a priority higher than 128. If another unit in the stack has a priority of 128 or higher, secure-setup will give the Active Controller a priority equal to the highest priority unit in the stack (which is by default the Standby Controller). When the Active Controller and the Standby Controller have identical priorities, during a reset, the old Active Controller cannot reassume its role from the Standby Controller (which has become the Active Controller at the reset).

If the previous Active Controller again becomes active, and you want it to resume the role of Active Controller, you should set the priority for the Standby Controller to a priority lower than 128. If you do not want the previous Active Controller to remain Active Controller, you can set the same priority for both Active and Standby Controllers (higher than, or equal to 128). For details, refer to [“IronStack unit priority”](#) on page 277.

NOTE

Secure-setup works for units within a single stack. It does not work across stacks.

[Figure 12](#) on page 241 shows an IronStack with three units in a ring topology. Refer to this figure as you follow the procedure steps for this scenario.

Configuring a three-member IronStack in a ring topology using secure-setup

1. Connect the devices using the stacking ports and stack cabling. For more information refer to the appropriate hardware installation guides.
2. Power on the units.
3. Connect your console to the intended Active Controller. The unit through which you run secure-setup becomes the Active Controller by default.
4. Issue the **stack enable** command on the intended Active Controller.

```
Brocade# config terminal
Brocade(config)# stack enable
Brocade(config)# exit
```

5. Enter the **stack secure-setup** command. As shown in the following example, this command triggers a Brocade proprietary discovery protocol that begins the discovery process in both upstream and downstream directions. The discovery process produces a list of upstream and downstream devices that are available to join the stack. Secure-setup can detect up to 7 units in each direction (14 total), but since the maximum number of units in a stack is 8, you must select a maximum of 7 units from both directions.

NOTE

To exit the secure-setup, enter **^C** at any time.

You should see output similar to the following.

```
Brocade# stack secure-setup
Brocade# Discovering the stack topology...
```

Current Discovered Topology - RING

Available UPSTREAM units

Hop(s)	Type	Mac Address
1	FCX624	0012.f239.2d40
2	FCX624	0012.f2d5.2100

Available DOWNSTREAM units

Hop(s)	Type	Mac Address
1	FCX624	0012.f2d5.2100
2	FCX624	0012.f239.2d40

Do you accept the topology (RING) (y/n)? : y

If you accept the topology, you will see output similar to the following.

Selected Topology:

Active	Id	Type	Mac Address
	1	FCX648	00e0.52ab.cd00

Selected UPSTREAM units

Hop(s)	Id	Type	Mac Address
1	3	FCX624	0012.f239.2d40
2	2	FCX624	0012.f2d5.2100

Connecting ICX 6450 and ICX 6430 devices in a stack

Selected DOWNSTREAM units

Hop(s)	Id	Type	Mac Address
1	2	FCX624	0012.f2d5.2100
2	3	FCX624	0012.f239.2d40

Do you accept the unit ids (y/n)? : y

To accept the unit ID assignments, type **y**. If you do not want to accept the ID assignments, type **n**. You can use secure-setup to renumber the units in your stack. Refer to [“Renumbering stack units”](#) on page 313.

If you accept the unit IDs, the stack is formed, and you can see the stack topology using the **show stack** command.

```
Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type   Role   Mac Address   Pri State   Comment
1   S FCX648 active  00e0.52ab.cd00 128 local   Ready
2   D FCX624 standby 0012.f2d5.2100 60 remote  Ready
3   D FCX624 member 0012.f239.2d40 0 remote  Ready
```

```

      active      standby
    +---+      +---+      +---+
-2/1| 1 |3/1--2/1| 2 |3/1--2/2| 3 |2/1-
    +---+      +---+      +---+
```

Current stack management MAC is 00e0.52ab.cd00

For ICX devices, it displays an equals sign (=) to show connections between trunk ports, for example:

```
ICX6610-24P POE Router#show stack stack-port
      active      standby
    +---+      +---+      +---+
=2/1| 1 |2/6==2/6| 5 |2/1==2/1| 4 |2/6=
|      +---+      +---+      +---+ |
|                                     |
|-----|
```

NOTE

For field descriptions for the **show stack** command, refer to [“Displaying stack information”](#) on page 293.

NOTE

In this output, D indicates a dynamic configuration. After you perform a **write memory**, this display will change to S, for static configuration.

- The Active Controller automatically checks all prospective stack members to see if they are password protected. If a unit is password protected, you will be asked to enter the password before you can add the unit. If you do not know the password, take one of the following actions:
 - Discontinue secure-setup by entering **^C**.
 - Obtain the device password from the administrator.
 - Continue secure-setup for your stack. The password-protected device and all devices connected behind it will not be included in the setup process.

In the following example, the second unit is password protected, so you are asked for the password.

Connecting ICX 6450 and ICX 6430 devices in a stack

```

Brocade# stack secure-setup
Brocade# Discovering the stack topology...

Verifying password for the password protected units...

Found UPSTREAM units
Hop(s)  Type      Mac Address
1       2       FCX648  001b.ed5e.c480
2       3       FCX648  00e0.5205.0000

Enter password for FCX648 located at 2 hop(s): ****

Enter the number of the desired UPSTREAM units (1-2)[1]: 2

Selected Topology:
Active  Id  Type      Mac Address
      1   FCX624  00e0.5201.4000

Selected UPSTREAM units
Hop(s)  Id  Type      Mac Address
1       2   FCX648  001b.ed5e.c480
2       3   FCX648  00e0.5205.0000

Do you accept the unit id's (y/n)??: y

```

- When the Active Controller has finished the authentication process, you will see output that shows the suggested assigned stack IDs for each member. You can accept these recommendations, or you can manually configure stack IDs. Enter the **show stack** command to verify that all units are in the ready state.

```

Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID  Type  Role      Mac Address      Pri State      Comment
1  S FCX624 active    00e0.5201.4000  128 local      Ready
2  S FCX648 standby  001b.ed5e.c480   0 remote    Ready
3  S FCX648 member  00e0.5205.0000   0 remote    Ready

      active      standby
      +---+      +---+      +---+
-2/1| 1 |3/1--2/1| 2 |3/1--2/2| 3 |2/1-
|      +---+      +---+      +---+
|      |      |      |
|-----|

Current stack management MAC is 00e0.5201.4000
Brocade#

```

For ICX devices, it displays the port up state of all ports of the trunk, for example:

```

ICX6610-24P POE Router#show stack stack-port

      active      standby
      +---+      +---+      +---+
=2/1| 1 |2/6==2/6| 5 |2/1==2/1| 4 |2/6=
|      +---+      +---+      +---+
|      |      |
|-----|

U#  Stack-port1      Stack-port2
1  up (1/2/1-1/2/5)  up (1/2/6-1/2/10)

```

```
up ports: 1/2/1, 1/2/2, 1/2/3, 1/2/4, 1/2/5
up ports: 1/2/6, 1/2/7, 1/2/8, 1/2/9, 1/2/10

4  up (4/2/1-4/2/5)                                up (4/2/6-4/2/10)
   up ports: 4/2/1, 4/2/2, 4/2/3, 4/2/4, 4/2/5
   up ports: 4/2/6, 4/2/7, 4/2/8, 4/2/9, 4/2/10

5  up (5/2/1-5/2/5)                                up (5/2/6-5/2/10)
   up ports: 5/2/1, 5/2/2, 5/2/3, 5/2/4, 5/2/5
   up ports: 5/2/6, 5/2/7, 5/2/8, 5/2/9, 5/2/10
```

NOTE

A 4x10G port consists of four sub-ports and the **show stack-port** command displays all sub-ports. So ports 1/2/2-1/2/5 in the previous code example are sub-ports of port 1/2/2 and 1/2/7-1/2/10 are sub-ports of 1/2/7.

NOTE

For field descriptions for the **show stack** command, refer to [“Displaying stack information”](#) on page 293.

8. Enter the **write memory** command on the Active Controller once all of the stack units are active. This command initiates configuration synchronization, which copies the configuration file of the Active Controller to the rest of the stack units.

NOTE

The secure-setup process may modify your configuration with information about new units, stacking ports, and so on. For this reason, it is very important to save this information by issuing the **write memory** command. If you do not do this, you may lose your configuration information the next time the stack reboots.

The secure-setup process for your stack is now complete.

NOTE

During the secure-setup process, after one minute of inactivity, authentication for stack members will expire and you will need to restart the process.

Scenario 2 - Three-member IronStack in a ring topology using the automatic setup process

FCX devices determine stacking port candidates through the default-port setting. An FCX stackable device with the default port configuration is still considered a clean unit. To ensure that the device remains a clean unit, do not do a write memory on the device.

For more detailed information about configuring an FCX IronStack, refer to [“FCX IronStack configuration”](#) on page 258.

Configuring a three-member IronStack in a ring topology using the automatic setup process

Complete the following steps to configure a three-member IronStack in a ring topology using automatic setup process.

1. Power on the devices.
2. This process requires clean devices (except for the Active Controller) that do not contain any configuration information. To change a device to a clean device, enter the **erase startup-config** command and reset the device. When all of the devices are clean, continue with the next step.

NOTE

The physical connections must be sequential, and must match the stack configuration.

3. Log in to the device that you want to be the Active Controller.
4. Configure the rest of the units by assigning ID numbers and module information on each unit. The stack ID can be any number from 1 through 8.

```
Brocade# config t
Brocade(config)# stack unit 2
Brocade(config-unit-2)# module 1 fcx-24-port-copper-base-module
Brocade(config-unit-2)# module 2 fcx-xfp-1-port-10g-module
Brocade(config-unit-2)# module 3 fcx-xfp-1-port-10g-module
Brocade(config-unit-2)# stack unit 3
Brocade(config-unit-3)# module 1 fcx-24-port-copper-base-module
Brocade(config-unit-3)# module 2 fcx-xfp-1-port-10g-module
Brocade(config-unit-3)# module 3 fcx-xfp-1-port-10g-module
```

NOTE

Each stack unit must have a unique ID number.

5. Assign a priority to the Active Controller using the **priority** command, as shown.

```
Brocade(config)# stack unit 1
Brocade(config-stack-1)# priority 255
```

Syntax: **priority** <num>

- The <num> variable is a value from 0 through 255. 255 is the highest priority.

6. Assign a priority to the unit that will act as Standby Controller.

```
Brocade# config t
Brocade(config)# stack unit 2
Brocade(config-unit-2)# priority 240
```

7. Enter the **write memory** command to save your settings.
8. Enter the **stack enable** command.
9. Physically connect the devices in a stack topology, which triggers an election during which the stack is automatically configured. For more information about cabling the devices, refer to the appropriate hardware installation guides.

NOTE

When you are configuring individual stack units, you can skip ID numbers. However, the sequence in which the units are connected must match the order in which you configure them. For example, you could configure unit 1 as FCX624, unit 3 as FCX648, unit 4 as FCX624, unit 6 as FCX624 and unit 7 as FCX648. The physical connection order must be: Active (FCX624), FCX648 (3), FCX624 (4), FCX624 (6) and FCX648 (7). The Active Controller is stack unit 1.

10. Verify your stack configuration by entering the **show running config** command.

```
Brocade# show running config
Current configuration:
!
ver 05.0.02
!
stack unit 1
  module 1 fcx-24-port-copper-base-module
  module 2 fcx-xfp-1-port-10g-module
  module 3 fcx-xfp-1-port-10g-module
  priority 255
stack unit 2
  module 1 fcx-24-port-copper-base-module
  module 2 fcx-xfp-1-port-10g-module
  module 3 fcx-xfp-1-port-10g-module
  priority 240
stack unit 3
  module 1 fcx-24-port-copper-base-module
  module 2 fcx-xfp-1-port-10g-module
  module 3 fcx-xfp-1-port-10g-module
stack enable
!
```

NOTE

For field descriptions for the **show running config** command, refer to [“Displaying running configuration information”](#) on page 305.

11. To see information about your stack, enter the **show stack** command.

```
Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type   Role   Mac Address   Pri State   Comment
1   S FCX624 active  00e0.5200.0100 255 local   Ready
2   S FCX624 standby 0012.f2eb.afc0 240 remote  Ready
3   S FCX624 member 001b.ed5d.alc0 0 remote   Ready

      active      standby
      +---+      +---+      +---+
-2/1| 1 |3/1--2/1| 2 |3/1--2/2| 3 |2/1-
|   +---+      +---+      +---+ |
|   |-----|-----|-----| |
|   |-----|-----|-----| |

Current stack management MAC is 00e0.5200.0100
Brocade#
```

Results for ICX devices are similar, with an equals signe (=) to show connections between trunk ports, rather than the hyphen symbol (-) symbol showing connection.

NOTE

For field descriptions for the **show stack** command, refer to [“Displaying stack information”](#) on page 293.

Configuration notes for building a stack using the automatic setup process

- If a new unit configuration matches other unit configurations, the Active Controller gives this unit the lowest *sequential* ID. For example, in a stack configuration that contains eight FCX624 configurations, but only units 1, 4 and 8 are currently active, if you place a new FCX624 unit between units 4 and 8, the new unit will be assigned ID 5, even though it might match unused IDs 2, 3, 5, 6, and 7, because 5 is the lowest *sequential* ID.
- In a ring topology, the same new unit might assume either ID if either direction produces sequential IDs. For example, in a four-member stack where IDs 2 and 4 are reserved, a new unit could assume either ID 2 or ID 4 because either ID 1,2,3 or 1, 3, 4 is sequential.

Scenario 3 - Three member IronStack in a ring topology using the manual configuration process

NOTE

For more detailed information about configuring an FCX IronStack, refer to [“FCX IronStack configuration”](#) on page 258

Complete the following steps to configure a three-member IronStack in a ring topology using the manual configuration process.

1. Power on the devices. Do not connect the stacking cables at this point.
2. Assign a priority of 255 to unit 1, and a priority of 240 to unit 3 using the **priority** command. You do not have to assign a priority to the third device. Enter the **stack enable** command on each device. In this example, device 1 will be the Active Controller and device 2 will be the Standby Controller.

Unit 1

```
Brocade# config t
Brocade(config)# stack unit 1
Brocade(config-unit-1)# priority 255
Brocade(config-unit-1)# stack enable
Enable stacking. This unit actively participates in stacking
Brocade(config-unit-1)# write memory
Write startup-config done.
Flash Memory Write (8192 bytes per dot) .Flash to Flash Done.
Brocade(config-unit-1)# end
```

Unit 2

```
Brocade# config t
Brocade(config)# stack enable
Enable stacking. This unit actively participates in stacking
Brocade(config)# Handle election, was standalone --> member, assigned-ID=2,
T=261285 ms.
```

```
Write startup-config done.
```

```
Flash Memory Write (8192 bytes per dot) .Flash to Flash Done.
Brocade(config-unit-1)# end
Brocade# config t
```

Unit 3

```
Brocade# config t
Brocade(config)# stack unit 1
Brocade(config-unit-1)# priority 240
Brocade(config-unit-1)# stack enable
```

```
Enable stacking. This unit actively participates in stacking
Brocade(config-unit-1)# end
```

3. Connect the devices in a stack topology. The Active Controller will retain its ID. The rest of the units are assigned unique ID numbers depending on the sequence in which you connected them.

For more information about cabling the devices, refer to the appropriate hardware installation guides.

NOTE

This method does not guarantee sequential stack IDs. If you want to change stack IDs to make them sequential, you can use secure-setup. Refer to [“Renumbering stack units”](#) on page 313.

FCX IronStack configuration

FCX devices cannot be intermixed with non-FCX devices, and FCX devices have more possible stacking ports.

Every FCX-S and FCXS-F device contains two default 16 Gbps stacking ports on the rear panel and two 10 Gbps ports on the front panel that can also be used as stacking ports.

NOTE

FCX-I and FCX-E devices can only be used for stacking if they have an optional 10 Gbps SFP+ module installed in the front panel. These devices do not have stacking ports on the rear panels.

An FCX IronStack may contain up to eight 24-port and 48-port devices, using any combination of the rear panel stacking ports and the front panel optional stacking ports. For FCX-S and FCXS-F devices, to use ports other than the factory-default 16 Gbps ports, you must define the ports for each device in the run time configuration. You can also configure the 16 Gbps ports to operate as 10 Gbps ports. Refer to [“Configuring FCX stacking ports”](#) on page 259.

An FCX “clean unit” may contain a default port configuration, but it is still considered a clean unit. To preserve this state, do not do a write memory on the unit before you build the stack. An FCX device with the default port configuration is still considered a clean unit. To ensure that the device remains a clean unit, do not do a write memory on the device. (Write memory adds a startup-config, and the device is no longer a clean unit.)

NOTE

The automatic setup process will not work for FCX devices that do not contain the default port information in their clean unit configurations.

Configuring FCX stacking ports

Brocade FCX-S and FCXS-F devices have two 10 Gbps ports on the front panel and two 16 Gbps ports on the rear panel. All of these ports may be used as stacking ports, however the non-default ports must be configured as stacking ports when setting up your FCX-S or FCXS-F IronStack.

FCX-I and FCX-E devices do not have 16 Gbps ports on the rear panel. These devices may be used in an IronStack by installing the 10 Gbps 4-port SFP+ module in the module slot on the front panel. Once you have installed one of these modules, ports 1 and 2 act as the default stacking ports. However, you can also use these ports to pass regular traffic, after disabling the stacking default. Refer to [“Changing default stacking port configurations”](#) on page 261.

NOTE

If you are adding FCX-E or FCX-I devices to a stack containing FCX-S or FCXS-F devices, you must reconfigure the stacking ports on the FCX-S or FCXS-F devices to be the 10 Gbps ports on the front panel. You can then connect all of the devices in a stack using front panel ports.

Changing FCX-S and FCXS-F CX4 ports from 16 Gbps to 10 Gbps

You can configure the 16 Gbps CX4 ports to operate as 10 Gbps ports using the **speed-duplex** command, as shown in the following example.

Syntax: **speed-duplex** [10-full | 10-half | 100-full | 100-half | 1000-full-master | 1000-full-slave | 10g-full | auto]

- **10-full** - 10M, full duplex
- **10-half** - 10M, half duplex
- **100-full** - 100M, full duplex
- **100-half** - 100M, half duplex
- **1000-full-master** - 1G, full duplex, master
- **1000-full-slave** - 1G, full duplex, slave
- **10g-full** - 10G, full duplex
- **auto** - Autonegotiation

NOTE

Both ends of a link must be configured for 10 Gbps for the link to operate as 10 Gbps. If you want the link to operate as a 16 Gbps link, both ends of the link must be configured for 16 Gbps.

```
Brocade(config-if-e10000-cx4-1/2/1)# speed-duplex 10g-full
Brocade(config-if-e10000-cx4-1/2/1)# end
Brocade# show int br | in Up
1/1/4   Up      Forward    Full 1G    None No 1    0    001b.f288.0003
1/2/1   Up      Forward    Full 10G    None No 1    0    001b.f288.0019
1/3/1   Up      Forward    Full 10G    None No N/A  0    001b.f288.001b
3/3/1   Up      Forward    Full 10G    None No N/A  0    0024.3814.9df3
mgmt1   Up      None       Full 1G    None No 1    0    001b.f288.0018
Brocade# show interface e 1/2/1
16GigabitEthernet1/2/1 is up, line protocol is up
Hardware is 16GigabitEthernet, address is 001b.f288.0019 (bia 001b.f288.0019)
Interface type is 16Gig CX4
Configured speed 10Gbit, actual 10Gbit, configured duplex fdx, actual fdx
Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
BPDU guard is Disabled, ROOT protect is Disabled
```

```

Link Error Dampening is Disabled
STP configured to ON, priority is level0, mac-learning is enabled
Flow Control is enabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IP MTU 1500 bytes, encapsulation ethernet
300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runs, 0 giants
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions
Relay Agent Information option: Disabled

```

Changing FCX-S and FCXS-F CX4 ports from 10 Gbps to 16 Gbps

To change the CX4 ports from 10 Gbps back to 16 Gbps, enter the **no speed-duplex 10g** command at the interface level of the CLI, as shown in this example.

```

Brocade(config-if-e10000-cx4-1/2/1)# no speed-duplex 10g
Brocade(config-if-e10000-cx4-1/2/1)# show interface br | in Up
1/1/4   Up      Forward    Full 1G   None No 1    0    001b.f288.0003
1/2/1   Up      Forward    Full 16G   None No 1    0    001b.f288.0019
1/3/1   Up      Forward    Full 10G   None No N/A  0    001b.f288.001b
3/3/1   Up      Forward    Full 10G   None No N/A  0    0024.3814.9df3
mgmt1   Up      None       Full 1G   None No 1    0    001b.f288.0018
Brocade(config-if-e10000-cx4-1/2/1)# show interface e 1/2/1
16GigabitEthernet1/2/1 is up, line protocol is up
Hardware is 16GigabitEthernet, address is 001b.f288.0019 (bia 001b.f288.0019)
Interface type is 16Gig CX4
Configured speed 16Gbit, actual 16Gbit, configured duplex fdx, actual fdx
Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
BPDU guard is Disabled, ROOT protect is Disabled
Link Error Dampening is Disabled
STP configured to ON, priority is level0, mac-learning is enabled
Flow Control is enabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IP MTU 1500 bytes, encapsulation ethernet
300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runs, 0 giants
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions
Relay Agent Information option: Disabled
Brocade(config-if-e10000-cx4-1/2/1)#

```

Configuring default ports on FCX devices

On FCX devices, the **default-port** command is used to define stacking port candidates. A stacking port is always a default port, but a default port may not necessarily be a stacking port. Default ports can become stacking ports using the secure-setup utility, or through automatic stack building.

Secure-setup probe packets can be received by a default port whether or not it is acting as a stacking port. Stacking packets can be only received by a stacking port (which is also always a default port). In order to use stacking ports that are not defined in the default configuration, you must define the port settings for each unit using the **default-port** command, so that secure-setup can discover the topology of the stack.

The 4-byte Ethernet preamble for the Ethernet frame is used when a port is configured as a default stacking port. For non-default ports, the standard 8-byte Ethernet preamble is used. For a default port that is used as a regular data port, the standard 8-byte Ethernet preamble must be explicitly enabled on the port using the **long-preamble** command. For details, refer to [“Configuring a default stacking port to function as a data port”](#) on page 264.

Stackable devices ship with two default stacking ports configured. Use the **stack-port** command to select only one of these factory default ports as the stacking port. If you do not configure **stack-port**, both default ports will operate as stacking ports.

Use the **default-port** command to use ports other than the factory default ports as stacking ports. You must configure **default-port** on each unit before building a stack. Once you have configured **default-port** on all units, you can then use any of the three stack construction methods to build a stack. The Active Controller then learns the port configuration for each unit.

NOTE

You cannot change the setting for a default port if the port is in use.

Changing default stacking port configurations

For FCX-E and FCX-I devices, ports 1 and 2 of the optional 10 Gbps SFP+ module (slot 2) act as the default stacking ports. You can change the default stacking ports to 3 and 4 on this module, or disable stacking, on all of the module ports. The following example changes the default ports on a 10 Gbps module from 1 and 2 to 3 and 4.

```
Brocade 10g-1(config)# stack unit 1
10g-1(config-unit-1)#
10g-1(config-unit-1)# default-ports 1/2/3 1/2/4
```

[Table 40](#) identifies the slot and port designations for each model.

NOTE

FCX-I and FCX-E models cannot be used in an IronStack without the addition of an optional 10 Gbps SFP+ module.

TABLE 40 Slot and port designations for FastIron stackable devices

Device	Slot 1	Slot 2	Slot 3	Slot 4
FCX624S-F	24 10/100/1000 ports on front panel	Two 16 Gbps ports on rear panel	Two 10 Gbps ports on front panel	N/A
FCX648S-F	48 10/100/1000 ports on front panel	Two 16 Gbps ports on rear panel	Two 10 Gbps ports on front panel	N/A
FCX-E devices with four-port 1 Gbps SFP module	Four-port 1 Gbps SFP module plus the first four copper ports act as a combo port. Slot 1 also contains the remaining 20 10/100/1000 ports.	N/A	N/A	N/A
FCX-I devices with four-port 1 Gbps SFP module	Four-port 1 Gbps SFP module plus the first four copper ports act as a combo port. Slot 1 also contains the remaining 20 10/100/1000 ports.	N/A	N/A	N/A
FCX-E devices with four-port 10 Gbps SFP+ module	48 10/100/1000 ports on front panel	Four-port 10 Gbps SFP+ module (supports stacking)	N/A	N/A
FCX-I devices with four-port 10 Gbps SFP+ module	48 10/100/1000 ports on front panel	Four-port 10 Gbps SFP+ module (supports stacking)	N/A	N/A

NOTE

Do not connect stacking ports to non-stacking ports. Stacking ports have a proprietary packet format that renders them incompatible with regular ports even when they are forwarding regular packets. In linear topologies, make sure that end units have only one stacking port configured (secure-setup automatically configures only one stacking port for an end unit).

Configuring a single stack port

To configure a single stack port, enter a command similar to the following.

```
Brocade(config)# stack unit 3
Brocade(config-unit-3)# stack-port 3/2/1
```

Syntax: [no] **stack-port** <stack-unit/slotnum/portnum>

If you enter an incorrect stack port number, you will get an error similar to the following.

```
Brocade(config-unit-3)# stack-port 3/4/1
Error! port 3/4/1 is invalid
Brocade(config-unit-3)# stack-port 3/2/1
```

To return both ports to stacking status, enter the **no stack-port** command on the single stacking port. This converts both ports to stacking ports. By default, if both ports are stacking ports, they are displayed by the system only when stacking is enabled. If only one port is configured as a stacking port, the system always displays this port.

Using secure-setup to build an FCX IronStack

You can use the secure-setup utility to build an FCX IronStack by performing the following step.

Enter **stack enable** and **stack secure-setup** commands when you have designated the desired stacking ports and connected your FCX units together, on stack unit 1, as shown in the following output.

```

Brocade# stack enable
Brocade# stack secure-setup
Brocade# Discovering the stack topology...

Available UPSTREAM units
Hop(s)  Id   Type      Mac Address
1        new FCX648POE  0012.f2d6.0511
2        new FCX624   0200.9999.0000

Enter the number of the desired UPSTREAM units (0-2)[0]: 2

Selected Topology:
Active  Id   Type      Mac Address
      1    FCX624POE  001b.f2e5.0100

Selected UPSTREAM units
Hop(s)  Id   Type      Mac Address
1        2    FCX648POE  0012.f2d6.0511
2        3    FCX624   0200.9999.0000

Do you accept the unit ids (y/n)? : y
Brocade# Election, was alone --> active, assigned-ID=1, total 3 units, my
priority=128
Election, was active, no role change, assigned-ID=1, total 3 units, my
priority=128
reset unit 2: diff bootup id=1
reset unit 3: diff bootup id=1
Election, was alone --> active, assigned-ID=1, total 3 units, my priority=128
Detect stack member 2 POE capable
Detect stack unit 2 has different startup config flash, will synchronize it
Detect stack unit 3 has different startup config flash, will synchronize it
Done hot swap: Set stack unit 3 to Ready
Done hot swap: Set stack unit 2 to Ready
Synchronize startup config to stack unit 2
Flash Memory Write (8192 bytes per dot).Synchronize startup config to stack
unit 3
Flash Memory Write (8192 bytes per dot).POE: Stack unit 2 Power supply 1 with 4
10000 mwatts capacity is up
Stack unit 2 Power supply 2 is down
Stack unit 3 Power supply 1 is up
Stack unit 3 Power supply 2 is down

Config changed due to add/del units. Do write mem if you want to keep it
Election, was active, no role change, assigned-ID=1, total 3 units, my
priority=128

Brocade#
Config changed due to add/del units. Do write mem if you want to keep it

Brocade# PoE Info: PoE module 1 of Unit 2 on ports 2/1/1 to 2/1/48 detected.
Initializing....
PoE Info: PoE module 1 of Unit 2 initialization is done.
```

```

Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type      Role      Mac Address      Pri State      Comment
1   S FCX624POE active   001b.f2e5.0100  128 local      Ready
2   D FCX648POE standby  0012.f2d6.0511   0 remote     Ready
3   D FCX624   member   0200.9999.0000   0 remote     Ready

          standby      active
+---+      +---+      +---+
| 3 |3/1--3/1| 2 |2/1--2/1| 1 |
+---+      +---+      +---+

Current stack management MAC is 001b.f2e5.0100
Brocade# write mem
Write startup-config done.
Flash Memory Write (8192 bytes per dot) .Flash to Flash Done.

Brocade#show stack
alone: standalone, D: dynamic config, S: static config
ID   Type      Role      Mac Address      Pri State      Comment
1   S FCX624POE active   001b.f2e5.0100  128 local      Ready
2   S FCX648POE standby  0012.f2d6.0511   0 remote     Ready
3   S FCX624   member   0200.9999.0000   0 remote     Ready

          standby      active
+---+      +---+      +---+
| 3 |3/1--3/1| 2 |2/1--2/1| 1 |
+---+      +---+      +---+

Current stack management MAC is 001b.f2e5.0100
Brocade#

```

Configuring a default stacking port to function as a data port

You can configure one of the two default stacking ports as a stacking port and the other port as a regular data port. By default, the 4-byte Ethernet preamble for the Ethernet frame is used when a port is configured as a default stacking port. This is done to compensate for extra overhead caused by stacking protocol. To use a default stacking port as a regular data port, the Ethernet preamble must be set to 8 bytes.

To configure a default port to use the long preamble, enter the **long-preamble** command at the Interface level of the CLI.

```

Brocade(config)#interface ethernet 1/2/1
Brocade(config-if-e10000-1/2/1)#long-preamble

```

Syntax: [no] long-preamble

Use the **no** form of the command to revert to the 4-byte Ethernet preamble.

Configuring an ICX 6610 IronStack

ICX 6610 devices can be stacked using the methods and topologies described in [“Supported IronStack topologies”](#) on page 239. This section describes how stacking ports on the ICX 6610 devices can be trunked.

ICX 6610 trunked stacking ports configuration

A trunk doubles the stacking port bandwidth, and provides better resilience. As long as at least one port of the trunk is connected properly, the communication between the neighboring units will work. Traffic is load balanced to the trunk ports. With large number of sessions, traffic should be well load balanced between 40G and 4*10G ports. Traffic should also be well distributed among the four sub-ports of a 4*10G port. If a 4*10G sub-port fails due to hardware failure, traffic is re-distributed to other ports. So, the stack system should still work. Periodical background diagnosis will print out warning messages that any 4*10G sub-port is down.

When a stacking port goes down, it should not cause stack election or topology change as long as the other port of the same trunk is up. The traffic interruption time should be sub-second for the system to detect the port down event and reprogram HW.

Stacking ports are trunked by default in Brocade ICX 6610 Series. You do not need to configure anything. The system automatically generate trunk configuration. You must enable stacking and connect cables properly for the stack to work.

The **show trunk** command shows user-configured or LACP trunks, and does not show stacking trunks.

At least one port of a trunk must be connected. If multiple ports of a trunk are connected, they must be connected to the same trunk of other unit. For example, it is invalid that the two ports of a trunk connect to two different units, or to different trunks of the same unit. Wrong connection might result in stacking formation failure or other problems. Some invalid connections still form a stack but with forwarding problems. Stack probe packets use the first connected port of a stacking trunk. So, as long as the first connected port of a trunk is connected properly, a stack can form. However, packet forwarding may not work for some streams that are hashed into the wrongly connected trunk port. In such case, periodical background diagnosis task could detect wrong connections in about 20 minutes, and it prints error messages.

Configuration notes for ICX stack topologies

- ICX devices do not allow users to change "stack-ports" configuration. Secure-setup sets the endpoints of a linear FCX stack to data ports, but it does not do so for ICX.
- ICX devices support up to 124 user-configured or LACP trunks. HW trunk ID of 0 is reserved, IDs 125-127 are used in trunk stacking ports and the internal cascade trunk in the 48-port unit.

NOTE

ICX 40G and 4x10G ports cannot be used as data ports, even when stacking is not enabled. In standalone mode, they drop all packets, except stacking probe packets. Secure Setup and unit replacement can therefore still discover a standalone unit.

Periodic background stack diagnosis for ICX 6610 devices

After a stack forms, the system periodically probes the topology to check the connections between units of this stack. It can detect the following errors. The purposes are to detect user's connection problem and hardware failures. Error messages are printed about every 10 minutes. If there is no printout, there is no problem. This diagnosis runs in the background.

Manually triggering stack diagnosis

You can manually trigger the diagnosis using the **show stack connection** command.

1. Ports of the same trunk connect to different units.
2. Ports of the same trunk connect to different trunks of the same unit.
3. Sub-ports of the 4*10G are down.
4. One end of a 10G port is up and the other end is down.
5. Communication problems between units of the stack

The most common connection error is connecting a 40G port to a 4*10G port. They use the same type of cable. In such case, the system might show one end is up, and one end is down. The stack cannot form, and the periodical background diagnosis task won't run at all.

Another connection error is that the two cables of a trunk go to different units, or to different trunks of the same unit as in case 1 and 2. Stack might still form in these cases. But it may cause internal forwarding loop, or have forwarding problem.

Cases 3 to 5 are due to hardware failures or software problems.

For example, the active unit console pops up the following messages if sub-ports of 4*10G are down. The stack should still work because of trunks. If all sub-ports of 4*10G are down, the diagnosis prints nothing because it cannot distinguish hardware failure or the cable is not connected.

```
*** Warning! miss 4*10G link 5/2/8(down)to 1/2/8(down). Stack can still work.
*** Warning! U1, dir=1, 4*10G ports: 1/2/8 are down.
*** Warning! U5, dir=1, 4*10G ports: 5/2/8 are down.
Please use "show stack conn" to view detailed connections
You can suppress the error messages by configuring "stack suppress-warning"
```

Configuring an ICX 6430 and ICX 6450 IronStack

ICX 6430 and ICX 6450 devices can be stacked using the methods and topologies described in [“Connecting ICX 6450 and ICX 6430 devices in a stack”](#) on page 244. The [“Configuring ICX 6430 or ICX 6450 trunked stacking ports”](#) section describes how stacking ports on the ICX 6430 and ICX 6450 devices can be trunked to form a single trunk-to-port connection on two directly connected stack units. The [“Configuring ICX 6430 or ICX 6450 multi-trunked stacking ports”](#) section describes how to configure a double port trunk configuration on the ICX 6430 and ICX 6450 devices. The [“Error messages encountered during the configuration of an ICX 6430 or ICX 6450 IronStack”](#) section describes the error messages that may occur when configuring an ICX 6430 or ICX 6450 IronStack.

Configuring ICX 6430 or ICX 6450 trunked stacking ports

NOTE

Brocade advises using the **stack-trunk** command in a new environment on first deployment. Brocade recommends using the **multi-trunk** command in a production environment. For more information about the **multi-trunk** command refer to [“Configuring ICX 6430 or ICX 6450 multi-trunked stacking ports”](#) on page 267.

The **stack-trunk** command forms a single trunk-to-port connection on two connected stack units. A trunk-to-port connection is formed when one side of the ports forms a trunk, and the other side of the ports does not. You can use the **stack-trunk** command to configure a stack trunk if one or both of the units are provisional units. You must enable stacking and connect cables properly for the stack to work. To enable the **stack-trunk** command, the primary port in the trunk must be configured under the **stack-port** command configuration.

For ICX 6450 devices, the data ports on both units must be configured to 10 Gbps port speed to enable trunking. To upgrade to 10 Gbps port speed on ports 2 and 4, use the ICX6450-2X10G-LIC-POD license. If the data port is not enabled for 10 Gbps port speed, the port status is down. For more information about configuring ports to 10 Gbps port speed, refer to [“Licensing for Ports on Demand”](#) on page 208.

Configure a stack truck by entering the following command under the stack unit configuration level.

```
Brocade (config)#stack unit 1
Brocade (config-unit-1)#stack-trunk 1/2/3 to 1/2/4
```

The following warning is displayed on the CLI if the **stack-trunk** command results in a trunk-to-port connection.

```
Error- this command will result in a port-to-trunk connection between stack 1 and
2. Please use "multi-trunk" command instead.
```

Use the **multi-trunk** command to configure a stack trunk on two directly connected stack units to ensure that a trunk-to-trunk connection is formed on both units at the same time.

Syntax: [no] **stack-trunk**

Use the **no** form of the command to disable the stack trunk configuration.

To save the configuration, enter the **write memory** command.

Configuring ICX 6430 or ICX 6450 multi-trunked stacking ports

To upgrade from a single to a double port trunk configuration, use the **multi-trunk** command. The **multi-trunk** command is used to ensure that a stack trunk is formed on two directly connected stack units at the same time. The **multi-trunk** command can only be enabled on the active controller unit. By configuring a multi-trunk on two sets of connected stack ports, a trunk-to-port connection is avoided. A trunk-to-port connection is formed when one side of the ports forms a trunk, and other side of the ports does not. A trunk-to-port connection can result in packet drops and can potentially break a stacking link.

The primary ports of the stack trunk must be connected and in an up status to enable the **multi-trunk** command. If the primary stack trunk ports are not connected and not in an up status, the command is rejected and the following error message is displayed on the CLI.

Error- Primary trunk port 1/2/3 is not UP; removing the trunk might break the stack

To upgrade to a double port trunk configuration, enter the **multi-trunk** command under the stack unit configuration level. Enter the following command.

```
Brocade(config)#stack unit 1
Brocade(config-unit-1)# multi-trunk 1/2/3 to 1/2/4 and 2/2/1 to 2/2/2
Brocade(config-unit-1)#stack unit 2
Brocade(config-unit-2)# multi-trunk 2/2/3 to 2/2/4 and 3/2/1 to 3/2/2
```

Syntax: [no] multi-trunk

Use the **no** form of the command to disable the configuration of the two connected stack trunk ports, and convert the secondary stack ports to data ports.

Displaying multi-trunked stacking configuration

When upgrading to a double port trunk configuration using the **multi-trunk** command, the configuration is not saved in the running or start-up configuration. Instead, the configuration for the **multi-trunk** command is saved as stack-trunk ports. Use the **show stack** command to display the configuration for multi-trunked stacking ports. For example, the following command is entered.

```
Brocade (config-unit-1)# multi-trunk 1/2/3 to 1/2/4 and 2/2/1 to 2/2/2
```

The running or start-up configuration displays the following configuration for the **show stack** command.

```
Brocade#show stack
stack unit 1
module 1 icx6450-48p-poe-port-management-module
  module 2 icx6450-sfp-plus-4port-40g-module
    stack-trunk 1/2/3 to 1/2/4
    stack-port 1/2/1 1/2/3
stack unit 2
module 1 icx6450-24-port-management-module
module 2 icx6450-sfp-plus-4port-40g-module
default-ports 2/2/1 2/2/3
stack-trunk 2/2/1 to 2/2/2
stack-port 2/2/1 2/2/3
```

Syntax: show stack

Periodic background stack diagnosis for ICX 6430 and ICX 6450 devices

After a stack forms, the system periodically probes the topology to check the connections between units of this stack. It can detect the following errors. The purposes are to detect user's connection problem and hardware failures. Error messages are printed about every 10 minutes. If there is no printout, there is no problem. This diagnosis runs in the background.

Manually triggering stack diagnosis

You can manually trigger the diagnosis using the **show stack connection** command.

1. Ports of the same trunk connect to different units.
2. Ports of the same trunk connect to different trunks of the same unit.
3. One end of a 10G port is up and the other end is down.
4. Communication problems between units of the stack

Another connection error is that the two cables of a trunk go to different units, or to different trunks of the same unit as in case 1 and 2. Stack might still form in these cases. But it may cause internal forwarding loop, or have forwarding problem.

Cases 3 to 5 are due to hardware failures or software problems.

Error messages encountered during the configuration of an ICX 6430 or ICX 6450 IronStack

The following error messages may occur when configuring an ICX 6430 or ICX 6450 IronStack:

- In you form a multi-trunk connection on two sets of stack ports that are not directly connected on neighboring units, the **multi-trunk** command is rejected and an error message is displayed. The following is an example of an error message.

```
Error- 1/2/3 and 1/2/4 are not connected; please use directly connected
stack-ports to form stack-trunks.
```

- When an unit joins or leaves a stack, or a stack trunk is configured using the **stack-trunk** command, a stack election is triggered. The **multi-trunk** command triggers a stack election among the stack units and reprograms (or removes) the stack trunk port in the hardware. A timer is set on all units to coordinate a stack election. The traffic interruption time generally takes less than 5 seconds for the system to detect the port down event and reprogram HW. You must wait for the stack election to be completed before entering another command. If you do not wait for the stack election to finish, the following warning message is displayed.

```
Stack port or trunk change is in progress, please try later.
```

You cannot enter the **stack-switchover** command until the stack election is completed. You can enter the **stack-switchover** command following the **multi-trunk** command configuration. If you enter the **stack-switchover** command before the stack election is completed, the following warning message is displayed.

```
Please try later, reason: during stack port or trunk deployment.
```

- If ports 2 and 4 of the ICX 6450-24 device are not configured to 10 Gbps port speed, then the **multi-trunk** command and the **stack-trunk** command are rejected with the following error message.

```
Error! port 1/2/2 is not configured as 10G
```

You must first enable the port to 10 Gbps port speed using the **speed-duplex 10g-full** command. For more information about configuring ports to 10 Gbps port speed, refer to [“Licensing for Ports on Demand”](#) on page 208.

- You cannot use the **stack-port** command to remove a stacking port if the port is part of a stack trunk. You must first remove the stack trunk and then remove the stack port. Use the **stack-trunk** command or the **multi-trunk** command to remove the stack trunk. If you attempt to remove the stack port before removing the stack trunk, an error message is displayed. The following is an example of an error message.

Please remove stack-trunk 1/2/3 - 1/2/4 using "stack-trunk" or "multi-trunk" command before removing stack port 1/2/3.

Verifying an IronStack configuration

Complete the following step to verify your IronStack configuration.

Log in to the Active Controller and verify the stack information by entering the **show running-config** and **show stack** or **show stack detail** commands. If your stack is configured properly, you should see the following:

- One Active Controller, one Standby Controller, and stack members.
- All stack members show a status of Ready

The following output shows an example configuration of an FCX IronStack.

```
Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type      Role      Mac Address      Pri State      Comment
2   S FCX648POE standby 00e0.5202.0000    0 remote Ready
3   S FCX624POE member 00e0.5203.0000    0 remote Ready
4   S FCX648   member 00e0.5204.0000    0 remote Ready
5   S FCX648POE member 0000.0000.0000    0 remoteReady
6   S FCX648POE active 00e0.5201.0000  128 local  Ready

active                                     standby
+---+ +---+ +---+ +---+
-2/1| 8 |2/2--2/1| 4 |2/2--2/1| 3 |2/2--2/1| 2 |2/2-
| +---+ +---+ +---+ +---+ |
|-----|
Current stack management MAC is 00e0.5201.0000
```

Results for ICX devices are similar, with an equals sign (=) to show connections between trunk ports, rather than the hyphen symbol (-) showing connections.

The next example shows output from the **show version** command for the same FCX stack.

```
Brocade# show version
Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
UNIT 8: compiled on Jun 17 2009 at 06:23:29 labeled as FCX06000a359
(3578117 bytes) from Primary FCX06000a359.bin
SW: Version 06.0.00a359T7f1
UNIT 2: compiled on Jun 17 2009 at 06:23:29 labeled as FCX06000a359
(3578117 bytes) from Primary FCX06000a359.bin
SW: Version 06.0.00a359T7f1
UNIT 3: compiled on Jun 17 2009 at 06:23:29 labeled as FCX06000a359
(3578117 bytes) from Primary FCX06000a359.bin
SW: Version 06.0.00a359T7f1
UNIT 4: compiled on Jun 17 2009 at 06:23:29 labeled as FCX06000a359
(3578117 bytes) from Primary FCX06000a359.bin
SW: Version 06.0.00a359T7f1
Boot-Monitor Image size = 365257, Version:06.0.00T7f5 (grz06000)
HW: Stackable FCX648P-POE
=====
UNIT 2: SL 1: FCX-48G POE 48-port Management Module
P-ENGINE 0: type DB90, rev 01
P-ENGINE 1: type DB90, rev 01
```

```

=====
UNIT 2: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 3: SL 1: FCX-24G POE 24-port Management Module
        P-ENGINE 0: type DB90, rev 01
=====
UNIT 3: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 3: SL 3: FCX-2XG 2-port 10G Module (2-XFP)
=====
UNIT 4: SL 1: FCX-48G 48-port Management Module
        P-ENGINE 0: type DB90, rev 01
        P-ENGINE 1: type DB90, rev 01
=====
UNIT 4: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
UNIT 4: SL 3: FCX-2XG 2-port 10G Module (2-XFP)
=====
UNIT 8: SL 1: FCX-48G POE 48-port Management Module
        P-ENGINE 0: type DB90, rev 01
        P-ENGINE 1: type DB90, rev 01
=====
UNIT 8: SL 2: FCX-2XGC 2-port 16G Module (2-CX4)
=====
        800 MHz Power PC processor (version 33/0022) 144 MHz bus
65536 KB flash memory
        256 MB DRAM
Monitor Option is on
STACKID 8 system uptime is 21 hours 2 minutes 23 seconds
STACKID 2 system uptime is 21 hours 2 minutes 22 seconds
STACKID 3 system uptime is 21 hours 2 minutes 23 seconds
STACKID 4 system uptime is 21 hours 2 minutes 22 seconds
The system : started=warm start reloaded=by "reload"
My stack unit ID = 8, bootup role = active
*** NOT FOR PRODUCTION ***

```

NOTE

For field descriptions for the **show running config** command, refer to [“Displaying running configuration information”](#) on page 305.

NOTE

For field descriptions for the **show stack** and **show stack detail** commands, refer to [“Displaying stack information”](#) on page 293.

The output from the **show stack** command contains a visual diagram of the stack. The dashed line between ports 1/2/1 and 3/2/1 indicates that this stack is configured in a ring topology. If the link between ports 1/2/1 and 3/2/1 is lost, the stack topology changes to linear, and the diagram changes to resemble the following.

```

      active      standby
      +---+      +---+      +---+
-2/1| 1 |3/1--2/1| 2 |3/1--2/2| 3 |2/1-
      +---+      +---+      +---+

```

The interfaces at either of a stack member are stacking ports. If no interface is displayed, it indicates that there is no stacking port configured. For example, the following diagram shows that stack units 1 and 3 each have only one stacking port configured.

Verifying an IronStack configuration

```
active      standby
+---+      +---+      +---+
| 1 |3/1--2/1| 2 |3/1--2/2| 3 |
+---+      +---+      +---+
```

For more detailed information, you can enter the **show stack detail** command.

Brocade IronStack management

Your Brocade IronStack can be managed through a single IP address. You can manage the stack using this IP address even if you remove the Active Controller or any member from the stack. You can also connect to the Active Controller through Telnet or SSH using this address. All management functions, such as SNMP, use this IP address to acquire MIB information and other management data.

A Brocade IronStack can be configured and managed using the command line interface (CLI) over a serial connection to a console port.

Logging in through the CLI

You can access the IronStack and the CLI in two ways:

- Through a direct serial connection to the console port
- Through a local or remote Telnet session using the stack IP address

You can initiate a local Telnet or SNMP connection by attaching a cable to a port and specifying the assigned management station IP address.

The stacking commands in the CLI are organized into the following levels:

- **Global** – Commands issued in the global mode are applied to the entire stack.
- **Stack Member Configuration Mode** – Commands issued in this mode apply to the specified stack member. Configuration information resides in the Active Controller.
- **Configuration Mode** – This is where you make configuration changes to the unit. To save changes across reloads, you need to save them to the Active Controller startup-config file. The configuration mode contains sub-levels for individual ports, for VLANs, for routing protocols, and other configuration areas.

NOTE

By default, any user who can open a serial or Telnet connection to the IronStack can access all of these CLI levels. To secure access, you can configure Enable passwords or local user accounts, or you can configure the Active Controller to use a RADIUS or TACACS/TACACS+ server for authentication. Refer to [Chapter 4, “Security Access”](#).

Logging in through the console port

When a device becomes a stack member in the IronStack, it establishes a remote connection to a virtual console port on the Active Controller. Input and output are relayed between the physical console port on the stack member and the virtual console port on the Active Controller. Since each stack member connects to an independent virtual console port on the Active Controller, the console ports on multiple stack units may be used simultaneously. However, messages displayed on the Active Controller physical console port during a reload will not be visible on the console ports of the stack members because the remote connections are not established until the software loading process is complete. It is preferable to connect a cable to the console port on the stack unit that will normally be the Active Controller, rather than to the console port of one of the other stack units.

When a stack unit establishes communication with the Active Controller, it also establishes a remote console session to the Active Controller. In a normally functioning IronStack, a console cable may be connected to any of the stack units and provide access to the same commands on the Active Controller.

You can terminate a session by entering **Ctrl+O** followed by **x** or **X**, or by entering the **exit** command from the User EXEC level, or by entering the **logout** command at any level.

NOTE

For the rconsole connections from the stack units to the Active Controller, the escape sequence and other methods of terminating the session are not available.

NOTE

Error messages that are generated during a reload of the Active Controller will not appear on rconsole connections from the stack units to the Active Controller. To see these error messages, you must connect a console cable to the Active Controller itself.

To establish an rconsole session, enter the **rconsole** command as shown:

```
Brocade# rconsole 1
```

Syntax: **rconsole** <stack-unit>

The following example shows how to establish rconsole sessions to stack members. Notice that the **show stack** command on the stack members displays different information than what is shown when the **show stack** command is entered on the Active Controller.

To see the status of your stack units, enter the **show stack** command on the Active Controller.

```
Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID  Type    Role      Mac Address      Pri  State   Comment
1  S FCX648P active    0012.f2de.8100   128  local   Ready
2  S FCX624P standby   0012.f2e2.ba40   0     remote  Ready
3  S FCX624P member    001b.ed7a.22c0   0     remote  Ready
```

```

      active      standby
      +---+      +---+      +---+
-2/1| 1 |3/1--2/1| 2 |3/1--2/2| 3 |2/1-
|   +---+      +---+      +---+ |
|   |           |           |   |
|   |-----|           |   |

```

```
Current stack management MAC is 0012.f2de.8100
Brocade#
```

NOTE

For field descriptions for the **show stack** command, refer to [“Displaying stack information”](#) on page 293.

Establish a remote console session with stack unit 2.

```
Brocade# rconsole 2
Connecting to unit 2... (Press Ctrl-O X to exit)
rconsole-2@Brocade#show stack
ID  Type    Role      Mac Address      Prio State   Comment
2  S FCX624P standby   0012.f2e2.ba40   0     local   Ready
```



```
rconsole-2@Brocade# exit
rconsole-2@Brocade> exit
Disconnected. Returning to local session...
```

Establish a remote console session with stack unit 3.

```
Brocade# rconsole 3
Connecting to unit 3... (Press Ctrl-O X to exit)
rconsole-3@Brocade# show stack
ID  Type   Role      Mac Address      Prio State   Comment
3  S  FCX624P  member          001b.ed7a.22c0    0   local   Ready

rconsole-3@Brocade# logout
Disconnected. Returning to local session...
Brocade#
```

IronStack management MAC address

The IronStack is identified in the network by a single MAC address, usually the MAC address of the Active Controller (the default). If a new Active Controller is elected, the MAC address of the new Active Controller (by default) becomes the MAC address for the entire stack. However, you can manually configure your stack to use a specified MAC address. Refer to [“Manually allocating the IronStack MAC address”](#) on page 275.

In an IronStack, the management MAC address is generated by the software, and is always the MAC address of the first port of the Active Controller. This ensures that the management MAC address remains consistent across stack reboots, and helps prevent frequent topology changes as a result of protocol enable, disable, and configuration changes.

When you are configuring Layer 2 protocols on stack units, such as STP, RSTP, and MSTP, the management MAC address of the Active Controller acts as the Bridge ID.

You can also configure the IronStack to retain its original MAC address, or wait for a specified amount of time before assuming the address of a new Active Controller, using the Persistent MAC Address feature (refer to [“Persistent MAC address for the IronStack”](#) on page 286).

Manually allocating the IronStack MAC address

You can manually configure your IronStack to use a specific MAC address. This overrides the default condition where the stack uses the MAC address of whatever unit is currently serving as Active Controller.

NOTE

The **stack mac** command is useful for administration purposes, however it should be used with caution to prevent duplication of MAC addresses.

NOTE

For hitless stacking failover, Brocade recommends that you configure the IronStack MAC address using the **stack mac** command. Without this configuration, the MAC address of the stack will change to the new base MAC address of the Active Controller. This could cause a spanning tree root change. Even without a spanning tree change, a client (for example, a personal computer) pinging the stack might encounter a long delay depending on the client MAC aging time. The client won't work until it ages out the old MAC address and sends ARP requests to relearn the new stack MAC address.

To configure a stack MAC address manually, enter the following command.

```
Brocade(config)# stack mac 0000.0000.0011
```

Syntax: [no] **stack mac** <mac-address>

The <mac-address> variable is a hexadecimal MAC address in the xxxx.xxxx.xxxx format.

Enter the **no** form of this command to return the MAC address to that of the Active Controller.

Output for this command resembles the following.

```
Brocade(config)# stack mac 0000.0000.0011
Brocade(config)# show running-config
Current configuration:
!
ver 05.0.01 100T7e1
!
stack 1
  module 1 fcx-48-port-copper-base-module
  module 2 fcx-cx4-1-port-10g-module
  priority 80
stack 2
  module 1 fcx-24-port-copper-base-module
  module 2 fcx-cx4-1-port-10g-module
  module 3 fcx-cx4-1-port-10g-module
stack enable
stack mac 0000.0000.0011
```

To display the stack MAC address, enter the **show chassis** command.

```
Brocade# show chassis
The stack unit 1 chassis info:

Power supply 1 (NA - AC - Regular) present, status ok
Power supply 2 not present

Fan 1 ok
Fan 2 ok
Exhaust Side Temperature Readings:
  Current temperature : 35.5 deg-C
  Warning level.....: 80.0 deg-C
  Shutdown level.....: 90.0 deg-C
Intake Side Temperature Readings:
  Current temperature : 33.5 deg-C
Boot Prom MAC: 0012.f2de.9440
Management MAC: 0000.0000.0011

The stack unit 2 chassis info:

Power supply 1 (NA - AC - Regular) present, status ok
Power supply 2 not present

Fan 1 ok
Fan 2 ok
--More--, next page: Space, next line: Return key, quit: Control-c
```

NOTE

For field descriptions for the **show chassis** command, refer to [“Displaying IronStack chassis information”](#) on page 291.

Removing MAC address entries

You can remove the following types of learned MAC address entries from the Brocade system MAC address table:

- All MAC address entries
- All MAC address entries for a specified Ethernet port
- All MAC address entries for a specified VLAN
- A specified MAC address entry in all VLANs

For example, to remove entries for the MAC address 000d.cb80.00d in all VLANs, enter the following command at the Privileged EXEC level of the CLI.

```
Brocade# clear mac-address 000d.cb80.00d0
```

Syntax: **clear mac-address** <mac-address> | **ethernet** <port> | **vlan** <number>

- If you enter the **clear mac-address** command without any parameters, the software removes all MAC entries.
- Use the <mac-address> variable to remove a specified MAC address from all VLANs. Specify the MAC address in the following format: HHHH.HHHH.HHHH.
- Use the **ethernet** <port> parameter to remove all MAC addresses for a specified Ethernet port. Specify the <port> variable in the format <stack-unit/slotnum/portnum>.
- Use the **vlan** <number> parameter to remove all MAC addresses for a specified VLAN.

IronStack unit identification

Stack units are identified by numbers 1 through 8. You can display stack unit IDs by entering the **show stack** command (refer to [“Displaying IronStack information”](#) on page 289).

A new device (one that has not been connected in an IronStack or has not been manually assigned a stack unit number) ships with a default number of 1. Once you enable stacking and the unit becomes part of an IronStack, its default stack unit number changes to the lowest available number in the stack.

Stack units must each have a unique identification number. Every stack member, including any standalone units, retains its stack unit number unless that number is already being used in the stack, or until you manually renumber the unit using secure-setup. For more information about how to renumber stack IDs using secure-setup, refer to [“Renumbering stack units”](#) on page 313.

IronStack unit priority

A unit with a higher priority is more likely to be elected Active Controller. The priority value can be 0 to 255 with a priority of 255 being the highest. The default priority value assigned to the Active Controller and Standby is 128.

You can assign the highest priority value to the stack unit you want to function as the Active Controller. When you enter a new priority value for a stack unit, that value takes effect immediately, but does not affect the current Active Controller until the next reset. However, if you enable hitless stacking failover, the stack unit with the highest priority will become the active controller in about five minutes (2 minutes in the case of ICX 6430 devices). For details, refer to [“Changing the priority of a stack unit”](#) on page 278.

You can give your Active and Standby Controllers the same priority, or different priorities (Active highest, Standby second-highest). When Active and Standby Controllers have the same priority, if the Active fails and the Standby takes over, then the original Active becomes operational again, it will not be able to resume its original role if the new Active Controller has more members.

In the same situation, when the priorities of the Active and Standby Controllers are different, the old Active Controller will regain its role and will reset the other units.

For example, suppose both Active and Standby Controllers have the same priority. If there are more than two units in a stack and the Active Controller leaves and comes back, it cannot win back the Active role because the new Active Controller has more members than the old Active Controller, which has no members. In this case, both the old Active Controller and new Active Controller have no members, so the unit with the longer up time wins the Active role. If the old Active Controller is reset, it cannot win. If the old Active Controller is not reset, it could win due to longer up time on up time or lower unit ID.

If you want to assign the same priority to the Active and Standby Controllers, you must do so after the stack is formed. This prevents the intended Standby Controller from becoming the Active Controller during stack construction.

Changing the priority of a stack member will trigger an election that takes effect immediately unless the Active Controller role changes. If this is the case, the changes will not take effect until after the next stack reload. However, if you enable hitless stacking failover, the stack unit with the highest priority will become the active controller without reload.

To display stack member priority values, enter the **show stack** command.

```
Brocade(config-unit-3)# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type  Role   Mac Address   Pri State   Comment
1 S   FCX624 active  0012.f2eb.a900 128 local    Ready
2 S   FCX624 standby 00f0.424f.4243 0  remote   Ready, member after reload
3 S   FCX624 member  001b.ed5d.a100 200 remote   Ready, active after reload
Brocade(config-unit-3)#
```

Changing the priority of a stack unit

To change the priority value for a stack unit, enter the **priority** command.

```
Brocade(Config)# stack unit 1
Brocade(Config-unit-1)# priority 128
```

Once a change in priority value has taken effect, if you have enabled hitless stacking failover, the stack unit with the highest priority will become the active controller without reload.

Syntax: **priority** <num>

The <num> variable is a value from 0 through 255. 255 is the highest priority.

CLI command syntax for stack units

CLI syntax that refers to stack units must contain all of the following parameters:

`<stack-unit>/<slotnum>/<portnum>`

- `<stack-unit>` - If the device is operating as a standalone, the stack-unit will be 1. Stack IDs can be any number from 1 through 8.
- `<slotnum>` - Refers to a specific group of ports on each device.
- `<portnum>` - A valid port number.

IronStack CLI commands

CLI commands specific to stacking are listed in [Table 41](#), with a link to the description for each command. For more information about CLI commands and syntax conventions, refer to [1](#), “Management Applications”.

TABLE 41 Stacking CLI commands

Command	Description location
copy flash flash	“Copying the flash image to a stack unit from the Active Controller” on page 283
clear stack ipc	“Troubleshooting an unsuccessful stack build” on page 315
cx4-10g	“Changing FCX-S and FCXS-F CX4 ports from 16 Gbps to 10 Gbps” on page 259
kill console	“Configuring TACACS/TACACS+ for devices in a Brocade IronStack” on page 140
priority	“Changing the priority of a stack unit” on page 278
rconsole	“Logging in through the console port” on page 273
reload stack unit	“Reloading a stack unit” on page 283
show chassis	“Displaying IronStack chassis information” on page 291
show flash	“Displaying IronStack flash information” on page 289
show memory	“Displaying IronStack memory information” on page 290
show module	“Displaying stack module information” on page 292
show running-config	“Displaying running configuration information” on page 305
show stack	“Displaying stack information” on page 293
show stack detail	“Displaying stack information” on page 293
show stack flash	“Displaying stack flash information” on page 296
show stack ipc	“Troubleshooting an unsuccessful stack build” on page 315
show stack neighbors	“Displaying stack neighbors information” on page 303
show stack resource	“Displaying stack information” on page 293
show stack rel-ipc stats	“Displaying stack rel-IPC statistics” on page 297
show stack rel-ipc stats unit #	“Displaying stack rel-IPC statistics for a specific stack unit” on page 301
show stack stack-port	“Displaying stack port information” on page 304
show statistics stack-port	“Displaying stacking port statistics” on page 309
show interfaces stack-ports	“Displaying stacking port interface information” on page 308

TABLE 41 Stacking CLI commands (Continued)

Command	Description location
show version	"Displaying software version information" on page 307
stack enable	"Enabling the stacking mode" on page 280
stack disable	"Enabling the stacking mode" on page 280
stack mac [mac-address]	"IronStack management MAC address" on page 275
stack persistent-mac-timer	"Persistent MAC address for the IronStack" on page 286
stack-port	"Changing default stacking port configurations" on page 261
default-ports	"Changing default stacking port configurations" on page 261
stack secure-setup	"Scenario 1 - Three-member IronStack in a ring topology using secure-setup" on page 250
stack unconfigure	"Unconfiguring an IronStack" on page 287
stack unconfigure rollback	"Unconfiguring an IronStack" on page 287
hitless-failover enable	"Enabling hitless stacking" on page 341
stack switch-over	"Executing a hitless stacking switchover" on page 344
debug stacking sync_rel_msg	"Displaying information about hitless stacking" on page 350
show stack failover	"Displaying information about stack failover" on page 350
show stack link-sync	"Displaying information about link synchronization status" on page 350

Enabling the stacking mode

When a unit is stack-enabled or joins a stack either actively or passively, it reserves priority queue 7 for stacking traffic control, assigns buffers for the stacking ports, and configures the first two 10 Gbps ports as stacking ports.

NOTE

Designated stacking ports cannot contain any configuration information, such as VLAN membership, etc. If configuration information exists, **stack enable** will fail. You must remove all configuration information from the port and re-issue the **stack enable** command.

To enable stacking mode on a new unit before you add it to the stack, enter the following command.

```
Brocade(config)# stack enable
Enable stacking. This unit actively participates in stacking
```

Syntax: [no] stack enable

To see the configuration of the stack at any time, enter the **show stacking configuration** command.

To remove stacking capability, enter the **no stack enable** command. This prevents the unit from actively sending out probe messages, however the unit could still be called to join a stack by an Active Controller. To prevent this, enter the **stack disable** command.

The **stack disable** command prevents a unit from sending or listening for any stacking probe messages. In this mode, the unit cannot be forced to join a stack.

```
Brocade(config)# stack disable
```

Syntax: [no] stack disable

To remove this restriction, enter the **no stack disable** command.

Important notes about stacking images

Consider the notes in this section when upgrading from a pre-stacking release to a stacking release, or when reverting from a stacking release to a pre-stacking release.

Refer to the release notes for instructions about upgrading the software.

Converting from a pre-stacking image to a stacking image

When you boot a stacking image (release 05.X or later) on a device with a pre-stacking (pre-release 05.X) startup-config.txt file, the system automatically converts the interface format from 0/X/X to 1/X/X. In addition, when a **write memory** command is issued, the pre-stacking startup-config.txt file is renamed to start-config.v4 and saved as a backup file.

NOTE

If you enter the **erase startup-config** command or **stack unconfigure clean** command, all startup-config.txt-related files, such as startup-config.v4 and startup-config.old are erased. You will no longer be able to recover pre-stacking startup-config.txt files.

Reverting from a stacking image to a pre-stacking image

To convert a device running a stacking image (release 05.X or later) to a pre-stacking (pre-release 05.X) image, use the **stack unconfigure rollback** command, which recovers the startup-config.v4 file that was saved when the **write memory** command was issued at the time the device was upgraded to a stacking image. For more information, refer to [“Recovering an earlier version of a stack configuration”](#) on page 288.

Encountering a problem after upgrading and reloading the software

If you encounter a problem after upgrading and reloading the software, make sure the device has the correct boot code version and the following (if applicable) are installed correctly:

- EEPROM
- Memory DIMM

NOTE

If you do not have the correct EEPROM for the upgrade, you will need to recover your pre-stacking image. For information about how to do this, refer to [“Recovering an earlier version of a stack configuration”](#) on page 288.

If the stacking EEPROM is missing or is not installed correctly, or if you have installed the wrong EEPROM, you will see output similar to the following.

```
FCX MEM size: 0x10000000
FCX Flash config....
FCX Boot Code Version 05.0.01
Enter 'b' to stop at boot....
BOOT INFO: load monitor from primary, size=103408
BOOT INFO: load image from primary.....
BOOT INFO: bootparam at 000543e8, mp_flash_size=002ee6c5
BOOT INFO: code decompression completed
```

```

BOOT INFO: branch to 00400100
Starting Main Task.....
*****
ERR: This software needs License PROM to be installed in the system
*****
System Reset!

```

If your memory DIMM is not installed correctly, you will see output similar to the following.

```

FCX Mem size: 0x8000000
Flash Config...
FCX Boot Code Version 05.0.01
Enter 'b' to stop at boot....
BOOT INFO: load monitor from primary, size = 103380
BOOT INFO: debug enabled!!
BOOT INFO: load image from primary...
BOOT INFO: bootparam at 00054338 mp_flash_size = 002flaeb
BOOT INFO: code decompression completed
BOOT INFO: branch to 00400100
Starting Main Task...

*****
ERR: This software requires 256M memory to be installed in the system.
*****
System Reset!

```

When you have confirmed that your hardware upgrade is installed correctly, restart the system and check the software version using the **show version** command. Refer to [“Confirming IronStack software versions”](#) in the next section.

Confirming IronStack software versions

All units in an IronStack must be running the same software image. To confirm this, check the software version on all devices that you want to add to your IronStack. Upgrade any units that are running older versions of the software before you build your stack.

1. Telnet, SSH, or connect to any of the console ports in the stack.
2. Enter the **show version** command. Output similar to the following is displayed.

```

Brocade# show version
Copyright (c) 1996-2010 Brocade Communications Systems, Inc.
  UNIT 1: compiled on Jan 26 2010 at 22:16:08 labeled as FCX07001
          (2441570 bytes) from Primary fcx07001.bin
          SW: Version 07.0.0151T7e1
  UNIT 2: compiled on Jan 26 2010 at 22:16:08 labeled as FCX07001
          (2441570 bytes) from Primary fcx07001.bin
          SW: Version 07.0.0151T7e1
  UNIT 3: compiled on Jan 26 2010 at 22:16:08 labeled as FCX07001
          (2441570 bytes) from Primary fcx07001.bin
          SW: Version 07.0.0151T7e1
  UNIT 4: compiled on Jan 26 2010 at 22:16:08 labeled as FCX07001
          (2441570 bytes) from Primary fcx07001.bin
          SW: Version 07.0.0151T7e1

```


NOTE

If any unit in the IronStack is running an incorrect version of the software, it will appear as non-operational. You must install the correct software version on that unit for it to operate properly in the stack. For more information, refer to [“Copying the flash image to a stack unit from the Active Controller”](#) in the next section.

Copying the flash image to a stack unit from the Active Controller

To copy the flash image to a stack unit from the Active Controller primary or secondary flash, enter the following command.

```
Brocade# copy flash flash unit-id-pri 2
```

Syntax: `copy flash flash [primary | secondary | unit-id-pri <unit-num> | unit-id-sec <unit-num>]`

- **primary** - Copy secondary to primary
- **secondary** - Copy primary to secondary
- **unit-id-pri** - Copy active primary image to unit ID
- **unit-id-sec** - Copy active secondary image to unit ID

The **unit-id-pri** and **unit-id-sec** keywords are used to copy images to a stack member from the Active Controller primary and secondary flash, respectively. For **<unit-num>**, enter a value from 1 through 8.

NOTE

You do not have to manually copy the flash image to a mismatched stack unit. For more information, refer to [“Auto Image Copy for stack units”](#) on page 321.

Reloading a stack unit

To reload a stack unit, enter the following command.

```
Brocade# reload
```

Syntax: `reload [after | at | cancel | unit-id <unit-list>]`

- **after** - Schedule reloading after certain time period
- **at** - Schedule reloading at an exact later time
- **cancel** - Cancel scheduled reload
- **unit-id** - Stack members to reload. The **<unit-list>** variable can be a combination, such as 2,4-6,8. Tokens must be separated by a comma and there is no space.

Controlling stack topology

Because Stackable devices allow you to use one of the two ports intended for stacking as a regular data port, you can control the size of your stack. The following example shows a stack where the existing ring topology is changed so that only one unit in the upstream direction is connected through a stacking port, which limits the size of the stack to two units.

```

Brocade# stack secure-setup
Brocade# Discovering the stack topology...

Current Discovered Topology - RING

Available UPSTREAM units
Hop(s)  Type      Mac Address
1       FCX624   0012.f2d5.2100
2       FCX624   001b.ed5d.9940

Available DOWNSTREAM units
Hop(s)  Type      Mac Address
1       FCX624   001b.ed5d.9940
2       FCX624   0012.f2d5.2100

Do you accept the topology (RING) (y/n)? : n

Available UPSTREAM units
Hop(s)  Type      Mac Address
1       FCX624   0012.f2d5.2100
2       FCX624   001b.ed5d.9940

Available DOWNSTREAM units
Hop(s)  Type      Mac Address
1       FCX624   001b.ed5d.9940
2       FCX624   0012.f2d5.2100

Enter the number of the desired UPSTREAM units (0-2)[0]: 1
Enter the number of the desired DOWNSTREAM units (0-1)[0]:

Selected Topology:
Active  Id   Type      Mac Address
      1    FCX624   0012.f239.2d40

Selected UPSTREAM units
Hop(s)  Id   Type      Mac Address
1       2    FCX624   0012.f2d5.2100

Do you accept the unit ids (y/n)? : y
Brocade#Election, was alone --> active, assigned-ID=1
reset unit 2: diff bootup id=1

Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID  Type   Role   Mac Address   Pri State   Comment
1  S FCX624 active  0012.f239.2d40 128 local   Ready
2  S FCX624 standby 0012.f2d5.2100 0 remote   Ready

```

Managing IronStack partitioning

When a unit in an IronStack with a linear topology fails, the IronStack divides (partitions) into two or more separate stacks that all have the same configuration. This may cause an IP address conflict in the network. If you want to keep the stacks separate, you will need to change the IP address of each new stack.

When a stack breaks into partitions, the partition with the Active Controller remains operational. If a partition contains the Standby Controller, this partition will become operational because the Standby Controller will assume the Active role and will reload the partition units. A partition without an Active or Standby Controller will not function. To reconfigure these units to act in standalone mode, you must first do a **stack unconfigure me** command on each unit. Refer to [“Unconfiguring an IronStack”](#) on page 287.

To reverse the partitioning, reconnect all of the units into the original stack topology using the stacking ports. This is the same as merging stacks. If the original Active Controller again has the highest priority, it will regain its role. If two partition Active Controllers have the same priority, the Active Controller with the most stack members will win the election. This process helps minimize traffic interruption.

Ring topology stacks do not partition in the event of a member failure. Operation is interrupted briefly while the stack recalculates a new path. Ring topologies are more stable than linear topologies because they provide redundant pathways in case of accidental failure.

Merging IronStacks

IronStacks may be merged, but the total number of stack units must not exceed eight. For example, you could combine two stacks with four units each into a single stack of eight units.

You can merge stacks by connecting them together using the stacking ports. Before doing this, make sure that none of the stacking ports have been reconfigured as data ports (for example, ports on an end unit in a linear stack topology). You cannot use secure-setup to merge stacks because secure-setup does not work across stack boundaries.

When stacks are merged, an election is held among the Active Controllers. The winner retains its configuration and the IDs of all of its original stack members. The remaining stack units lose their configuration and are reset. If the IDs of the losing stack units conflict with the IDs of the winning units they may change, and the IDs will no longer be sequential. You can use secure-setup to renumber the members in the newly merged stack. The following examples show how stack merging works:

- If a stack partitions into multiple stacks because of a connection failure, and you fix the connection, the stack partitions will merge back into the original stack with no change to stack IDs, because in this case all stack IDs are distinct.
- In a linear stack topology, the end units of the stack will have only one stacking port configured. Before you can merge two linear stacks, you must reconfigure the end units so that both ports are stacking ports.

MIB support for the IronStack

All statistics about packets received and sent, RMON, jumbo frames, runts, giants, and other instances are gathered through the stack interfaces and are accessible through SNMP. The functionality for an IronStack is the same as that for a standard 10 Gbps interface. Information includes types of modules, including optics modules.

NOTE

A type counter has been added to count the number of packets greater than 1518 bytes (jumbo frames).

For detailed information about stacking MIBs, refer to the *MIB Reference Guide*.

Persistent MAC address for the IronStack

The MAC address for the entire IronStack is determined by the MAC address of the Active Controller. When an Active Controller is removed from the stack, and a new Active Controller is elected, by default the MAC address of the new Active Controller becomes the MAC address for the IronStack. When you enable the Persistent MAC Address feature, you configure a time delay before the stack MAC address changes. During this configured interval, if the previous Active Controller is reinstalled in the stack, the stack continues to use the MAC address of this unit, even though it may no longer be the Active Controller. If the previous Active Controller does not rejoin the stack during the specified time interval, the stack assumes the address of the new Active Controller as the stack MAC address.

The Persistent MAC Address feature allows you to configure a period of time during which the original base MAC address will not change if the Active Controller fails, or is removed for maintenance. This timer is triggered when the Standby Controller becomes the Active Controller. When the timer expires, the new Active Controller will change the previous MAC address to its base MAC address and advertise this MAC address to management VLANs to update the ARP peer table. If you want to use the new address, you will have to re-enter the **stack-persistent-mac-timer** command again to reactivate the persistent MAC address,

To enable Persistent MAC Address, enter the following command.

```
Brocade(config)# stack persistent-mac-timer 120
```

Syntax: [no] **stack persistent-mac-timer** <number>

The <number> variable is the number of minutes during which the IronStack will retain the original MAC Address if the Active Controller fails or is removed for service. The valid value range is from 5 through 6000 minutes. If you enter a 0, it means “keep this address forever”. The default is 60 minutes.

To disable Persistent MAC Address, enter the following command.

```
Brocade(config)# no stack persistent-mac-timer
```

NOTE

If you enter the **no** form of this command while the persistent MAC address timer is active, the stack will disregard the persistent MAC address and will assume the MAC address of the new Active Controller.

NOTE

Persistent MAC and stack MAC cannot be used together.

In the following example, the persistent MAC timer has been set to the default of 60 minutes.

```
Brocade(config)# stack persistent-mac 60

Brocade(config)# show running-config
Current configuration:
!
ver 05.0.011T7e1
!
stack 1
  module 1 fcx-48-port-copper-base-module
  module 2 fcx-cx4-1-port-10g-module
  priority 80
stack 2
```

```

module 1 fcx-24-port-copper-base-module
module 2 fcx-cx4-1-port-10g-module
module 3 fcx-cx4-1-port-10g-module
stack 3
  module 1 fcx-48-port-management-module
  module 2 fcx-cx4-2-port-10g-module
  priority 40
stack enable
stack persistent-mac 60

```

To display the stack MAC addresses, enter the **show stack** command.

```

Brocade(config)# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type   Role      Mac Address      Prio State   Comment
1   S   FCX648p  active          0012.f2d5.9380   80   local   Ready
2   S   FCX648   member          00e0.6666.8880   0    remote  Ready
3   S   FCX624   standby         0012.f2dc.0ec0   40   remote  Ready
Current persistent MAC is 0012.f2d5.9380
Brocade(config)# stack mac 111.111.111
Error: persistent stacking MAC address timer is configured
Brocade(config)#

```

The following example shows what the Persistent MAC information looks like in the output of the **show stack** command when the Standby Controller becomes the Active Controller.

```

Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type   Role      Mac Address      Prio State   Comment
1   S   FCX648P  active          0000.0000.0000   80   reserved
2   S   FCX648   standby         00e0.6666.8880   0    remote  Ready
3   S   FCX624   master         0012.f2dc.0ec0   40   local   Ready
Brocade#Persistent MAC timer expires in 59 minutes 52 seconds.

Current persistent MAC is 0012.f2d5.9380

```

Unconfiguring an IronStack

The **stack unconfigure** command is a run time command that returns stack units to their pre-stacking state. When a stack unit is unconfigured, its stacking flash is removed, and its startup-config.txt flash file is recovered. These actions apply to all units to which this command is applied, regardless of the role of the unit in the stack.

When the **stack unconfigure** command is applied to the Active Controller, it removes **stack enable** from the run time configuration but not from the startup configuration. If you want to remove **stack enable** from the Active Controller permanently, you must enter the **write memory** command.

When the **stack unconfigure** command is applied to the Standby Controller or a stack member (besides the Active Controller) it removes **stack enable** from the recovered startup-config.txt and resets the unit.

When a stack unit that did not have an original startup-config file is unconfigured, it becomes a clean unit. It is possible that this unit could automatically rejoin the stack if its module configuration matches that of the Active Controller. To prevent this from happening accidentally, it is best to first disconnect the unit which has to be unconfigured, and then issue the **stack unconfigure me** command on it.

To remove the configuration from a specific IronStack unit, or from the entire stack, enter a command similar to the following.

```
Brocade# stack unconfigure me
```

Syntax: `stack unconfigure [<stack-unit> | all | me | clean | rollback]`

- `<stack-unit>` - Unconfigure the stack member with this ID
- `all` - Unconfigure every unit including this unit
- `me` - Unconfigure this unit only
- `clean` - Removes all startup configuration files including v4 and v5 and makes this a clean unit
- `rollback` - Recovers the earlier version (4 or 3) of a startup configuration (refer to [“Recovering an earlier version of a stack configuration”](#) on page 288)

NOTE

The `stack unconfigure me` command is available to all units, while `stack unconfigure all` and `stack unconfigure <stack-unit>` are available on the Active Controller only.

The following example shows a session where stack unit 2 is unconfigured.

```
Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type   Role   Mac Address   Pri State   Comment
1 S   FCX624 active  0012.f2eb.a900 128 local   Ready
2 S   FCX648 standby 00f0.424f.4243  0 remote Ready
3 S   FCX624 member 00e0.5201.0100  0 remote Ready

Brocade# stack unconfigure 2
Will recover pre-stacking startup config of this unit, and reset it. Are you sure?
(enter 'y' or 'n'): y
Stack 2 deletes stack bootup flash and recover startup-config.txt from .old

Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID   Type   Role   Mac Address   Pri State   Comment
1 S   FCX624 active  0012.f2eb.a900 128 local   Ready
2 S   FCX648 member 0000.0000.0000  0 reserved
3 S   FCX624 standby 00e0.5201.0100  0 remote   Ready
```

When the `stack unconfigure 2` command is issued, stack unit 2 recovers the `startup-config.txt` from the `startup-config.old` configuration file that was saved when this unit downloaded its configuration from the Active Controller. As the output shows, stack member 2 has been removed from the stack, and ID 2 is now reserved for a replacement unit. Stack member 3 is now the Standby Controller.

Recovering an earlier version of a stack configuration

The `stack unconfigure rollback` command recovers the earlier, non-stacking version (4.X or 3X) of a startup configuration. When a system that is running stacking (05.X and later) discovers that its startup configuration is an earlier version, it saves this configuration to a `startup-config.v4` file when a `write memory` command is issued, or when the Active Controller tries to synchronize all startup configurations. The `stack unconfigure rollback` command is similar to the `stack unconfigure` command except that it does not remove “stack enable” from the run time or startup configuration.

The **stack unconfigure** and **stack unconfigure rollback** commands are unrelated and recover different startup-config.txt files. Both commands permanently delete the current startup-config.txt and replace it with a pre-stacking (pre-05.X) startup-config.txt file.

NOTE

When you issue the **stack unconfigure rollback** command to recover the previous startup-config.v4 file, DO NOT issue a **write memory** command, as **write memory** will overwrite the recovered file. You should reboot from a pre-5.X image without doing a **write memory**.

If you enter the **erase startup-config** or **stack unconfigure clean** commands, all startup-config text-related files, such as startup-config.v4 and startup-config.old are erased. You will no longer be able to recover pre 5.X startup-config.txt files.

Syntax: **stack unconfigure rollback** [<stack-unit> | **all** | **me**]

- <stack-unit>- Recover the version 4 startup-config file for the stack member with this ID
- **all** - Recover the version 4 startup-config file for every unit including this unit
- **me** - Recover the version 4 startup-config file for this unit only

Displaying IronStack information

This section describes the **show** commands for an IronStack, including output examples and field descriptions.

Displaying IronStack flash information

Use the **show flash** command to display flash memory information for all members of a stack, or for a specified stack member.

Syntax: **show flash** <stack-unit>

Output from the **show flash** command for a stack resembles the following (for a stack with three members).

From the Active Controller for the entire stack:

```
Brocade# show flash
Stack unit 1:
Compressed Pri Code size = 3034232, Version 05.0.00T7e1 (fcx05000.bin)
Compressed Sec Code size = 2873568, Version 04.2.00T7e1 (fcx04200.bin)
Compressed BootROM Code size = 405217, Version 04.0.00T7e5
Code Flash Free Space = 2146304
Stack unit 2:
Compressed Pri Code size = 3034232, Version 05.0.00T7e1 (fcx05000.bin)
Compressed Sec Code size = 2873523, Version 04.2.00aT7e1 (fcx04200a.bin)
Compressed BootROM Code size = 403073, Version 03.0.00T7e5
Code Flash Free Space = 24117248
Stack unit 3:
Compressed Pri Code size = 3034232, Version 05.0.00T7e1 (fcx05000.bin)
Compressed Sec Code size = 2873568, Version 04.2.00T7e1 (fcx04200.bin)
Compressed BootROM Code size = 405217, Version 04.0.00T7e5
Code Flash Free Space = 2252800
Brocade#
```

For stack member 3 only:

```
Brocade# show flash unit 3
Stack unit 3:
Compressed Pri Code size = 3034232, Version 05.0.00T7e1 (fcx05000.bin)
Compressed Sec Code size = 2873568, Version 04.2.00T7e1 (fcx04200.bin)
Compressed BootROM Code size = 405217, Version 04.0.00T7e5
Code Flash Free Space = 2252800
Brocade#
```

Table 42 describes the fields displayed in this example.

TABLE 42 Field definitions for the **show flash** command

Field	Description
Compressed Pri Code size	The compressed size, version, and image name for the Primary Code
Compressed Sec Code size	The compressed size, version, and image name for the Secondary Code
Compressed BootROM Code size	The compressed size and version for the BootROM Code
Code Flash Free Space	The amount of available free space on the Flash memory

Displaying IronStack memory information

The **show memory** command displays information about stack units. The following example shows output from this command for a stack with eight units.

```
Brocade# show memory
Stack unit 1:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 182820476 bytes free, 23% used
Stack unit 2:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 172751776 bytes free, 27% used
Stack unit 3:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 172751776 bytes free, 27% used
Stack unit 4:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 172751776 bytes free, 27% used
Stack unit 5:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 107140664 bytes free, 54% used
Stack unit 6:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 172751740 bytes free, 27% used
Stack unit 7:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 182820504 bytes free, 23% used
Stack unit 8:
  Total DRAM: 268435456 bytes
  Dynamic memory: 238026752 bytes total, 182811440 bytes free, 23% used
Brocade#
```

Syntax: **show memory**

Table 43 describes the fields displayed in this output example.

TABLE 43 Field definitions for the show memory command

Field	Description
Total DRAM	The size (in bytes) of DRAM
Dynamic memory	The total number of bytes in dynamic memory, including the number of bytes that are available (free, or unused), and the percentage of memory used.

Displaying IronStack chassis information

The **show chassis** command displays chassis information for each stack unit. Output resembles the following (in this example, a three member stack).

```

Brocade# show chassis
The stack unit 1 chassis info:

Power supply 1 (NA - AC - Regular) present, status ok
Power supply 2 not present

Fan 1 ok
Fan 2 ok
Exhaust Side Temperature Readings:
    Current temperature : 33.0 deg-C
    Warning level.....: 85.0 deg-C
    Shutdown level.....: 90.0 deg-C
Intake Side Temperature Readings:
    Current temperature : 31.0 deg-C
Boot Prom MAC: 0012.f2e4.6e00
Management MAC: 0012.f2e4.6e00

The stack unit 2 chassis info:

Power supply 1 (NA - AC - Regular) present, status ok
Power supply 2 not present

Fan 1 ok
Fan 2 ok
Exhaust Side Temperature Readings:
    Current temperature : 32.5 deg-C
    Warning level.....: 85.0 deg-C
    Shutdown level.....: 90.0 deg-C
Intake Side Temperature Readings:
    Current temperature : 31.0 deg-C
Boot Prom MAC: 0012.f2e3.11c0

The stack unit 3 chassis info:

Power supply 1 (NA - AC - Regular) present, status ok
Power supply 2 not present

Fan 1 ok
Fan 2 ok
Exhaust Side Temperature Readings:
    Current temperature : 31.5 deg-C
    Warning level.....: 85.0 deg-C
    Shutdown level.....: 90.0 deg-C

```

```
Intake Side Temperature Readings:
    Current temperature : 32.0 deg-C
Boot Prom MAC: 0012.f2db.e500
```

Syntax: show chassis

Table 44 describes the fields displayed in this output example.

TABLE 44 Field definitions for the show chassis command

Field	Description
Power Supply 1	The status of the primary power supply.
Power Supply 2	The status of the secondary power supply, if present.
Fan 1 and Fan 2	The status of the cooling fans
Exhaust Side Temperature Readings	From the air exhaust side of the chassis, the current temperature reading, the warning level temperature setting, and the shutdown level temperature setting.
Intake Side Temperature Reading	The current temperature reading from the air intake side of the chassis.
Boot Prom MAC	The MAC address of the boot prom
Management MAC	For the Active Controller only, the management MAC address

Displaying stack module information

The **show module** command displays information about the stack unit modules. Output resembles the following.

```
Brocade(config)# show module
Module                                     Status Ports Starting MAC
S1:M1 FCX-24G 24-port Management Module + PoE      OK      24      00e0.5201.4000
S1:M2 FCX-2XGC 2-port 10G Module (2-CX4)           OK       2      00e0.5201.4018
S1:M3 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      00e0.5201.401a
S3:M1 FCX-48G 48-port Management Module            OK      48      001b.ed5e.c480
S3:M2 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      001b.ed5e.c4b0
S3:M3 FCX-1XGC 1-port 10G Module (1-CX4)           OK       1      001b.ed5e.c4b1
S4:M1 FCX-48G 48-port Management Module            OK      48      001b.ed5e.ac00
S4:M2 FCX-1XGC 1-port 10G Module (1-CX4)           OK       1      001b.ed5e.ac30
S4:M3 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      001b.ed5e.ac31
S5:M1 FCX-24G 24-port Management Module            OK      24      001b.ed5d.a180
S5:M2 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      001b.ed5d.a198
S5:M3 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      001b.ed5d.a199
S5:M4 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      001b.ed5d.a19a
S6:M1 FCX-24G 24-port Management Module            OK      24      00e0.5200.3000
S6:M2 FCX-1XGC 1-port 10G Module (1-CX4)           OK       1      00e0.5200.3018
S6:M3 FCX-1XGC 1-port 10G Module (1-CX4)           OK       1      00e0.5200.3019
S7:M1 FCX-48G 48-port Management Module            OK      48      00e0.4444.0000
S7:M2 FCX-1XGC 1-port 10G Module (1-CX4)           OK       1      00e0.4444.0030
S7:M3 FCX-1XGC 1-port 10G Module (1-CX4)           OK       1      00e0.4444.0031
S8:M1 FCX-48G 48-port Management Module            OK      48      0012.f2eb.d540
S8:M2 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      0012.f2eb.d570
S8:M3 FCX-1XG 1-port 10G Module (1-XFP)            OK       1      0012.f2eb.d571
Brocade(config)#
```

Syntax: `show module`

Table 45 describes the fields displayed in this output example.

TABLE 45 Field definitions for the **show module** command

Field	Description
Module	Identifies the module, by stack unit ID, module number, module type
Status	The status of this module
Ports	The number of ports in this module
Starting MAC	The starting MAC address for this module

Displaying stack resource information

Use the **show stack resource** command to display stack resource information, as shown in the example output from a Brocade ICX 6610 switch.

```
Brocade# show stack resource
alloc in-use avail get-fail limit get-mem size init
register attribute 4800 2710 2090 0 556800 4810 334 2400
general 12B data 32 10 22 0 7424 12 12 32
RB-tree node 4096 2714 1382 0 237568 3026 18 1024
variable length link 3905 4 3901 0 905960 4 8 3905
AU msg dev0 4092 0 4092 0 16368 0 16 4092
AU msg dev1 4092 0 4092 0 16368 0 16 4092
```

Syntax: `show stack resource`

Table 46 describes the output fields for this command. This command displays the following information for register attributes, general 12B data, and RB-tree node.

TABLE 46 Field definitions for the **show stack resource** command

Field	Description
alloc	Memory allocated
in-use	Memory in use
avail	Available memory
get-fail	The number of get requests that have failed.
limit	The maximum memory allocation
get-mem	The number of get-memory requests
size	The size
init	The number of requests initiated.

Displaying stack information

You can display information about any and all of the members in an IronStack by entering **show** commands from the Active Controller console port. If you enter **show** commands from a unit that is not the Active Controller, the information may not be displayed correctly.

The **show stack** command displays general information about an IronStack, for all members, for a specified member, and with additional detail if required.

The following output covers the entire stack, as shown in this example output from a Brocade ICX 6610 switch.

```
Brocade#show stack
```

alone: standalone, D: dynamic config, S: static config

ID	Type	Role	Mac Address	Pri	State	Comment
1	S ICX6610-24P	member	748e.f834.5238	0	remote	Ready
2	S ICX6610-48P	member	748e.f834.4800	0	remote	Ready
3	S ICX6610-24F	member	001b.f385.0124	0	remote	Ready
4	S ICX6610-48P	active	748e.f834.4930	200	local	Ready
5	S ICX6610-48P	standby	748e.f834.4d14	200	remote	Ready
6	S ICX6610-24P	member	748e.f834.50b4	0	remote	Ready
7	S ICX6610-24P	member	748e.f834.504c	0	remote	Ready
8	S ICX6610-24F	member	0000.0000.0000	200	reserve	

```

active
+---+      +---+      +---+      +---+      +---+      +---+
=2/1| 4 |2/6==2/6| 3 |2/1==2/1| 2 |2/6==2/6| 1 |2/1==2/1| 7 |2/6==2/6| 6 |2/1=
+---+      +---+      +---+      +---+      +---+      +---+
|
|
|
|
|
standby
+---+
-----2/1| 5 |2/6=
+---+

```

```
Standby u5 - protocols ready, can failover or manually switch over
Current stack management MAC is 001b.1234.1234
```

If you add a stack member ID, output is displayed for that member only.

```
Brocade# show stack 1
```

ID	Type	Role	Mac Address	Prio	State	Comment	
1	S	FCX648	active	0012.f2eb.a900	130	local	Ready

```
Brocade# show stack 2
```

ID	Type	Role	Mac Address	Prio	State	Comment	
2	S	FCX648	standby	00f0.424f.4243	0	remote	Ready, member after reload

```
Brocade#show stack 3
```

ID	Type	Role	Mac Address	Prio	State	Comment	
3	S	FCX624	member	00f0.424f.4243	0	remote	Ready

If you add **detail** to the **show stack** command, output resembles the following on a Brocade ICX 6610 switch.

```
Brocade(config)# show stack detail
```

alone: standalone, D: dynamic config, S: static config

ID	Type	Role	Mac Address	Pri	State	Comment	
1	S	ICX6610-24P	member	748e.f834.5238	0	remote	Ready
2	S	ICX6610-48P	member	748e.f834.4800	0	remote	Ready
3	S	ICX6610-24F	member	001b.f385.0124	0	remote	Ready
4	S	ICX6610-48P	active	748e.f834.4930	200	local	Ready
5	S	ICX6610-48P	standby	748e.f834.4d14	200	remote	Ready
6	S	ICX6610-24P	member	748e.f834.50b4	0	remote	Ready
7	S	ICX6610-24P	member	748e.f834.504c	0	remote	Ready
8	S	ICX6610-24F	member	0000.0000.0000	200	reserve	

$$\begin{array}{ccccccccc} \text{active} & & & & & & & & \\ +---+ & & +---+ & & +---+ & & +---+ & & +---+ \\ =2/1\mid 4\mid 2/6==2/6\mid 3\mid 2/1==2/1\mid 2\mid 2/6==2/6\mid 1\mid 2/1==2/1\mid 7\mid 2/6==2/6\mid 6\mid 2/1= \end{array}$$

```

| +---+ +---+ +---+ +---+ +---+ +---+ |
|                                     standby |
|                                     +---+  |
|-----2/1| 5 |2/6=
|                                     +---+

```

Standby u5 - protocols ready, can failover or manually switch over
 Current stack management MAC is 001b.1234.1234

Stack Port Status		Neighbors	
Unit#	Stack-port1	Stack-port2	Stack-port1
1	up (1/2/1-1/2/5)	up (1/2/6-1/2/10)	unit7 (7/2/1-7/2/5)
unit2	(2/2/6-2/2/10)		
2	up (2/2/1-2/2/5)	up (2/2/6-2/2/10)	unit3 (3/2/1-3/2/5)
unit1	(1/2/6-1/2/10)		
3	up (3/2/1-3/2/5)	up (3/2/6-3/2/10)	unit2 (2/2/1-2/2/5)
unit4	(4/2/6-4/2/10)		
4	up (4/2/1-4/2/5)	up (4/2/6-4/2/10)	unit5 (5/2/1-5/2/5)
unit3	(3/2/6-3/2/10)		
5	up (5/2/1-5/2/5)	up (5/2/6-5/2/10)	unit4 (4/2/1-4/2/5)
unit6	(6/2/1-6/2/5)		
6	up (6/2/1-6/2/5)	up (6/2/6-6/2/10)	unit5 (5/2/6-5/2/10)
unit7	(7/2/6-7/2/10)		
7	up (7/2/1-7/2/5)	up (7/2/6-7/2/10)	unit1 (1/2/1-1/2/5)
unit6	(6/2/6-6/2/10)		

Unit# System uptime

1	5 days 6 hours 58 minutes 20 seconds
2	5 days 6 hours 58 minutes 20 seconds
3	5 days 6 hours 58 minutes 20 seconds
4	5 days 6 hours 58 minutes 21 seconds
5	1 days 11 hours 45 minutes 37 seconds
6	5 days 6 hours 58 minutes 21 seconds
7	5 days 6 hours 58 minutes 21 seconds

The system started at 23:43:14 GMT+00 Thu Oct 20 2011

Syntax: `show stack <stack-unit> | detail`

Table 47 describes the fields displayed by the **show stack** command.

TABLE 47 Field descriptions for the **show stack** command

Field	Description
alone: Standalone	This device is operating as a standalone device
S: static configuration	The configuration for this unit is static (has been saved with a write memory command).
D: dynamic configuration	The configuration for this unit is dynamic and may be overwritten by a new stack unit. To change to a static configuration, enter the write memory command.
ID	The stack identification number for this unit.
Type	The model of this unit.
Role	The role of this unit within the stack.
MAC address	The MAC address of this unit.
Priority	The priority assigned to this unit.

TABLE 47 Field descriptions for the **show stack** command (Continued)

Field	Description
State	The operational state of this unit.
Comments	Additional information about this unit (optional).

NOTE

The Active Controller removes the dynamic configuration of a unit when the unit leaves. However, if there is a static trunk configuration associated with the unit, the Active Controller cannot remove the dynamic configuration. In this case, you must remove the static trunk and use the **no stack unit x** to manually remove the configuration.

[Table 48](#) describes the output from the **show stack detail** command (in addition to the **show stack** command fields shown in the previous table).

TABLE 48 Field descriptions for the **show stack detail** command

Field	Description
Stack Port Status	Indicates stacking port status for each stack unit.
Neighbors	Identifies stack neighbors (by unit ID) for each stack unit.
ID	The stack identification number for this unit.
Stack-port 1	Indicates the port state (up or down) and identifies the port by number (stack-ID/slot/port).
Stack-port 2	Indicates the port state (up or down) and identifies the port by number (stack-ID/slot/port).

Displaying stack flash information

Use the **show stack flash** command to display information about flash memory for stack members, as shown in this example output from a Brocade ICX 6610 switch.

```
Brocade# show stack flash
There is no startup-config.old
Stack flash that was read in bootup:
ICX6610-48P, ID =4, role= active, pri=200, config=1, jumbo=X PPVLAN=X S2M=0 FIPS=X
stack p: [0]=4/2/1 [1]=4/2/6 default p: 4/2/1(5) 4/2/6(5), , , hash-chain=X
vlan#=X ve#=X stp#=X
active-chg=0

Current written stack flash:
ICX6610-48P, ID =4, role= active, pri=200, config=1, jumbo=X PPVLAN=X S2M=0 FIPS=X
stack p: [0]=4/2/1 [1]=4/2/6 default p: 4/2/1(5) 4/2/6(5), , , hash-chain=X
vlan#=X ve#=X stp#=X
active-chg=0
```

Syntax: show stack flash

Table 49 describes the output from the **show stack flash** command.

TABLE 49 Field descriptions for the **show stack flash** command

Field	Description
ID	Device ID
role	The role of this device in the stack
priority	The priority of this device in the stack
config	Indicates the port state (up or down) and identifies the port by number (stack-ID/slot/port).
<i>The rest of the fields are used for debug purposes only.</i>	

Displaying stack rel-IPC statistics

Use the **show stack rel-ipc stats** command to display session statistics for stack units. The following output is observed on a Brocade ICX 6610 switch.

```
Brocade# show stack rel-ipc stats
Reliable IPC statistics:

Global statistics:
Pkts rcvd w/no session: 0
Msgs rcvd w/no handler: 0

Unit statistics:

Unit 2 statistics:
Msgs sent: 41384  Msgs received: 14052, Pkt sends failed: 0

Message types sent:
    [9]=21674,      [10]=19703,      [11]=2,      [13]=5,

Message types received:
    [9]=14016,      [10]=2,      [11]=28,      [13]=6,

Session statistics: base-channel, unit 2, channel 0:
Session state: established (last established 15 hours 33 minutes 31 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 14636, Msgs received: 14039
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 30892, Pkts received: 30842
Msg bytes sent: 1828190, Msg bytes received: 1232988
Pkt bytes sent: 2659848, Pkt bytes received: 1763028
Flushes requested: 30, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 888, ACK: 14010, WND: 437, ACK+WND: 0
DAT: 15556, DAT+ACK: 1, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 1069, Zero-window probes sent: 0
Dup ACK pkts rcvd: 1224, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: image-transfer, unit 2, channel 1:
```

```

Session state: established (last established 15 hours 11 minutes 2 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 9850, Msgs received: 1
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 9899, Pkts received: 10606
Msg bytes sent: 10124076, Msg bytes received: 8
Pkt bytes sent: 10341308, Pkt bytes received: 127284
Flushes requested: 1, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 1, ACK: 1, WND: 0, ACK+WND: 0
DAT: 9897, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 49, Zero-window probes sent: 0
Dup ACK pkts rcvd: 757, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: ACL, unit 2, channel 3:
Session state: established (last established 15 hours 33 minutes 31 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 7011, Msgs received: 4
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 7588, Pkts received: 7617
Msg bytes sent: 629316, Msg bytes received: 5840
Pkt bytes sent: 802504, Pkt bytes received: 107508
Flushes requested: 0, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 1, ACK: 1, WND: 0, ACK+WND: 2
DAT: 7584, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 573, Zero-window probes sent: 0
Dup ACK pkts rcvd: 596, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: sync-reliable, unit 2, channel 4:
Session state: established (last established 15 hours 32 minutes 27 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 27, Msgs received: 1
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 53, Pkts received: 40
Msg bytes sent: 39420, Msg bytes received: 1460
Pkt bytes sent: 73836, Pkt bytes received: 1944
Flushes requested: 0, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 2, ACK: 1, WND: 0, ACK+WND: 0
DAT: 50, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 22, Zero-window probes sent: 0
Dup ACK pkts rcvd: 6, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: rconsole-server-to-2, unit 2, channel 6:
Session state: established (last established 15 hours 33 minutes 30 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 5, Msgs received: 6
Atomic batches sent: 0, Atomic batches received: 0

```


Pkts sent: 14, Pkts received: 40
 Msg bytes sent: 183, Msg bytes received: 56
 Pkt bytes sent: 384, Pkt bytes received: 1052
 Flushes requested: 5, Suspends: 0, Resumes: 0
 Packets sent with data (DAT), ACKs, and window updates (WND):
 Other: 4, ACK: 5, WND: 0, ACK+WND: 0
 DAT: 5, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
 Data retransmits done: 0, Zero-window probes sent: 0
 Dup ACK pkts rcvd: 0, Pkts rcvd w/dup data: 0
 Pkts rcvd w/data past window: 0

Unit 3 statistics:

Msgs sent: 41356 Msgs received: 14007, Pkt sends failed: 0

Message types sent:

[9]=21623, [10]=19703, [11]=29, [13]=1,

Message types received:

[9]=14003, [10]=2, [13]=2,

Session statistics: base-channel, unit 3, channel 0:

Session state: established (last established 15 hours 33 minutes 49 seconds ago)

Connections established: 1

Remote resets: 0, Reset packets sent: 0

Connection statistics (for current connection, if established):

Msgs sent: 14647, Msgs received: 14003

Atomic batches sent: 0, Atomic batches received: 0

Pkts sent: 31055, Pkts received: 31403

Msg bytes sent: 1801742, Msg bytes received: 1232204

Pkt bytes sent: 2402644, Pkt bytes received: 1877788

Flushes requested: 32, Suspends: 0, Resumes: 0

Packets sent with data (DAT), ACKs, and window updates (WND):

Other: 1269, ACK: 13911, WND: 437, ACK+WND: 0

DAT: 15346, DAT+ACK: 92, DAT+WND: 0, DAT+ACK+WND: 0

Data retransmits done: 966, Zero-window probes sent: 0

Dup ACK pkts rcvd: 661, Pkts rcvd w/dup data: 0

Pkts rcvd w/data past window: 0

Session statistics: image-transfer, unit 3, channel 1:

Session state: established (last established 15 hours 11 minutes 2 seconds ago)

Connections established: 1

Remote resets: 0, Reset packets sent: 0

Connection statistics (for current connection, if established):

Msgs sent: 9850, Msgs received: 1

Atomic batches sent: 0, Atomic batches received: 0

Pkts sent: 9930, Pkts received: 10599

Msg bytes sent: 10124076, Msg bytes received: 8

Pkt bytes sent: 10457352, Pkt bytes received: 127200

Flushes requested: 1, Suspends: 0, Resumes: 0

Packets sent with data (DAT), ACKs, and window updates (WND):

Other: 1, ACK: 1, WND: 0, ACK+WND: 0

DAT: 9928, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0

Data retransmits done: 140, Zero-window probes sent: 0

Dup ACK pkts rcvd: 798, Pkts rcvd w/dup data: 0

Pkts rcvd w/data past window: 0

Session statistics: ACL, unit 3, channel 3:

Session state: established (last established 15 hours 33 minutes 49 seconds ago)

Connections established: 1

Remote resets: 0, Reset packets sent: 0

```

Connection statistics (for current connection, if established):
Msgs sent: 7004, Msgs received: 0
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 7447, Pkts received: 7300
Msg bytes sent: 616352, Msg bytes received: 0
Pkt bytes sent: 774304, Pkt bytes received: 87600
Flushes requested: 0, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 2, ACK: 0, WND: 0, ACK+WND: 0
DAT: 7445, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 441, Zero-window probes sent: 0
Dup ACK pkts rcvd: 295, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: rconsole-server-to-3, unit 3, channel 7:
Session state: established (last established 15 hours 33 minutes 48 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 1, Msgs received: 2
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 3, Pkts received: 2
Msg bytes sent: 35, Msg bytes received: 20
Pkt bytes sent: 76, Pkt bytes received: 52
Flushes requested: 1, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 1, ACK: 1, WND: 0, ACK+WND: 0
DAT: 1, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 0, Zero-window probes sent: 0
Dup ACK pkts rcvd: 0, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Unit 4 statistics:
Msgs sent: 41337 Msgs received: 14035, Pkt sends failed: 0

Message types sent:
    [9]=21632,    [10]=19702,    [11]=2,    [13]=1,

Message types received:
    [9]=14031,    [10]=2,    [13]=2,

Session statistics: base-channel, unit 4, channel 0:
Session state: established (last established 15 hours 33 minutes 49 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 14630, Msgs received: 14031
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 30186, Pkts received: 31052
Msg bytes sent: 1801548, Msg bytes received: 1234680
Pkt bytes sent: 2325044, Pkt bytes received: 1857824
Flushes requested: 30, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 1199, ACK: 13879, WND: 434, ACK+WND: 4
DAT: 14522, DAT+ACK: 148, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 197, Zero-window probes sent: 0
Dup ACK pkts rcvd: 560, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: image-transfer, unit 4, channel 1:

```

```

Session state: established (last established 15 hours 11 minutes 2 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 9850, Msgs received: 1
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 9852, Pkts received: 10675
Msg bytes sent: 10124076, Msg bytes received: 8
Pkt bytes sent: 10284896, Pkt bytes received: 128112
Flushes requested: 1, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 1, ACK: 1, WND: 0, ACK+WND: 0
DAT: 9850, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 2, Zero-window probes sent: 0
Dup ACK pkts rcvd: 826, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: ACL, unit 4, channel 3:
Session state: established (last established 15 hours 33 minutes 49 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 7004, Msgs received: 0
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 7051, Pkts received: 7240
Msg bytes sent: 616352, Msg bytes received: 0
Pkt bytes sent: 733028, Pkt bytes received: 86880
Flushes requested: 0, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 3, ACK: 0, WND: 0, ACK+WND: 0
DAT: 7048, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 44, Zero-window probes sent: 0
Dup ACK pkts rcvd: 234, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics: rconsole-server-to-4, unit 4, channel 8:
Session state: established (last established 15 hours 33 minutes 48 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 1, Msgs received: 2
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 5, Pkts received: 8
Msg bytes sent: 35, Msg bytes received: 20
Pkt bytes sent: 140, Pkt bytes received: 264
Flushes requested: 1, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 2, ACK: 1, WND: 0, ACK+WND: 0
DAT: 2, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 1, Zero-window probes sent: 0
Dup ACK pkts rcvd: 1, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

```

Syntax: show stack rel-ipc stats

Displaying stack rel-IPC statistics for a specific stack unit

To display IPC statistics for a specific unit, enter the following command:

Brocade IronStack management

```
Brocade# show stack rel-ipc stats unit 3

Unit 3 statistics:
Msgs sent: 1217  Msgs received: 509, Pkt sends failed: 0

Message types sent:
[9]=1182,          [10]=2,          [11]=2,          [13]=2,
[19]=29,

Message types received:
[9]=506,          [10]=1,          [13]=2,

Session statistics, unit 3, channel 0:
Session state: established (last established 32 minutes 19 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 971, Msgs received: 506
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 1205, Pkts received: 1088
Msg bytes sent: 44281, Msg bytes received: 19308
Pkt bytes sent: 238004, Pkt bytes received: 34652
Flushes requested: 59, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 2, ACK: 504, WND: 7, ACK+WND: 0
DAT: 691, DAT+ACK: 1, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 129, Zero-window probes sent: 0
Dup ACK pkts rcvd: 18, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics, unit 3, channel 2:
Session state: established (last established 32 minutes 17 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 0, Msgs received: 0
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 1, Pkts received: 7
Msg bytes sent: 0, Msg bytes received: 0
Pkt bytes sent: 12, Pkt bytes received: 84
Flushes requested: 0, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 1, ACK: 0, WND: 0, ACK+WND: 0
DAT: 0, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 0, Zero-window probes sent: 0
Dup ACK pkts rcvd: 7, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics, unit 3, channel 3:
Session state: established (last established 32 minutes 19 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 242, Msgs received: 0
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 243, Pkts received: 246
Msg bytes sent: 8712, Msg bytes received: 0
Pkt bytes sent: 12596, Pkt bytes received: 2952
Flushes requested: 0, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
```

```

Other: 1, ACK: 0, WND: 0, ACK+WND: 0
DAT: 242, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 0, Zero-window probes sent: 0
Dup ACK pkts rcvd: 4, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0

Session statistics, unit 3, channel 6:
Session state: established (last established 32 minutes 17 seconds ago)
Connections established: 1
Remote resets: 0, Reset packets sent: 0
Connection statistics (for current connection, if established):
Msgs sent: 2, Msgs received: 2
Atomic batches sent: 0, Atomic batches received: 0
Pkts sent: 8, Pkts received: 13
Msg bytes sent: 123, Msg bytes received: 20
Pkt bytes sent: 232, Pkt bytes received: 296
Flushes requested: 2, Suspends: 0, Resumes: 0
Packets sent with data (DAT), ACKs, and window updates (WND):
Other: 5, ACK: 1, WND: 0, ACK+WND: 0
DAT: 2, DAT+ACK: 0, DAT+WND: 0, DAT+ACK+WND: 0
Data retransmits done: 0, Zero-window probes sent: 0
Dup ACK pkts rcvd: 6, Pkts rcvd w/dup data: 0
Pkts rcvd w/data past window: 0
Brocade#

```

Syntax: show stack rel-ipc unit *<num>*

Displaying stack neighbors information

The **show stack neighbors** command displays information about stack member neighbors. The following output is from a Brocade ICX 6610 switch.

```

Brocade# show stack neighbors
U#   Stack-port1                Stack-port2
1    unit7 (7/2/1-7/2/5)       unit2 (2/2/6-2/2/10)
2    unit3 (3/2/1-3/2/5)       unit1 (1/2/6-1/2/10)
3    unit2 (2/2/1-2/2/5)       unit4 (4/2/6-4/2/10)
4    unit5 (5/2/1-5/2/5)       unit3 (3/2/6-3/2/10)
5    unit4 (4/2/1-4/2/5)       unit6 (6/2/1-6/2/5)
6    unit5 (5/2/6-5/2/10)      unit7 (7/2/6-7/2/10)
7    unit1 (1/2/1-1/2/5)       unit6 (6/2/6-6/2/10)

```

Topology: Ring, 7 unit(s), order: 4 3 2 1 7 6 5

[illegible]

```

|                                     +---+ |
-----2/1| 5 |2/6=
                                     +---+

```

Syntax: `show stack neighbors`

[Table 50](#) describes the output from the `show stack neighbors` command.

TABLE 50 Field descriptions for the `show stack neighbors` command

Field	Description
U	The stack identification number for this unit.
Stack-port1	Identifies the neighbor stack unit for stack-port1 for this unit ID.
Stack-port2	Identifies the neighbor stack unit for stack-port2 for this unit ID.

Displaying stack port information

The `show stack stack-ports` command displays information about stack port status.

```

Brocade(config)# show stack stack-ports
ID      Stack-port1      Stack-port2
1       up (1/2/1)       up (1/2/2)
2       up (2/2/1)       up (2/2/2)
3       up (3/2/1)       up (3/3/1)
4       up (4/2/1)       up (4/3/1)
5       up (5/2/1)       up (5/3/1)

```

For ICX devices, it displays an equals sign (=) to show connections between trunk ports and is enhanced to display the port up state of all trunked ports. The following example is observed on a Brocade ICX 6610 switch.

```

Brocade#show stack stack-ports
active
      +---+      +---+      +---+      +---+      +---+      +---+
=2/1| 4 |2/6==2/6| 3 |2/1==2/1| 2 |2/6==2/6| 1 |2/1==2/1| 7 |2/6==2/6| 6 |2/1=
|      +---+      +---+      +---+      +---+      +---+      +---+
|                                     standby
|                                     +---+
-----2/1| 5 |2/6=
                                     +---+

U#  Stack-port1      Stack-port2
1   up (1/2/1-1/2/5)  up (1/2/6-1/2/10)
   up ports: 1/2/1, 1/2/2, 1/2/3, 1/2/4, 1/2/5
   up ports: 1/2/6, 1/2/7, 1/2/8, 1/2/9, 1/2/10

2   up (2/2/1-2/2/5)  up (2/2/6-2/2/10)
   up ports: 2/2/1, 2/2/2, 2/2/3, 2/2/4, 2/2/5
   up ports: 2/2/6, 2/2/7, 2/2/8, 2/2/9, 2/2/10

3   up (3/2/1-3/2/5)  up (3/2/6-3/2/10)
   up ports: 3/2/1, 3/2/2, 3/2/3, 3/2/4, 3/2/5
   up ports: 3/2/6, 3/2/7, 3/2/8, 3/2/9, 3/2/10

4   up (4/2/1-4/2/5)  up (4/2/6-4/2/10)
   up ports: 4/2/1, 4/2/2, 4/2/3, 4/2/4, 4/2/5

```

```

up ports: 4/2/6, 4/2/7, 4/2/8, 4/2/9, 4/2/10

5  up (5/2/1-5/2/5)                                up (5/2/6-5/2/10)
   up ports: 5/2/1, 5/2/2, 5/2/3, 5/2/4, 5/2/5
   up ports: 5/2/6, 5/2/7, 5/2/8, 5/2/9, 5/2/10

6  up (6/2/1-6/2/5)                                up (6/2/6-6/2/10)
   up ports: 6/2/1, 6/2/2, 6/2/3, 6/2/4, 6/2/5
   up ports: 6/2/6, 6/2/7, 6/2/8, 6/2/9, 6/2/10

7  up (7/2/1-7/2/5)                                up (7/2/6-7/2/10)
   up ports: 7/2/1, 7/2/2, 7/2/3, 7/2/4, 7/2/5
   up ports: 7/2/6, 7/2/7, 7/2/8, 7/2/9, 7/2/10

```

Syntax: show stack stack-ports

[Table 51](#) describes the output from the **show stack stack-ports** command.

TABLE 51 Field descriptions for the **show stack stack-ports** command

Field	Description
ID	The stack identification number for this unit
Stack-port1	Indicates port state (up or down) and identifies the port by number (stack-ID/slot/port)
Stack-port 2	Indicates port state (up or down) and identifies the port by number (stack-ID/slot/port)

Displaying running configuration information

The **show running-config** command displays information about the current stack configuration.

```

Brocade(config)# show running-config
Current configuration:
!
ver 05.0.00T7e1
!
stack unit 1
  module 1 fcx-24-port-management-module
  module 2 fcx-cx4-2-port-10g-module
  module 3 fcx-xfp-1-port-10g-module
  stack-port 1/2/1 1/3/1
stack unit 2
  module 1 fcx-48-port-management-module
  module 2 fcx-xfp-2-port-10g-module
stack unit 3
  module 1 fcx-48-port-copper-base-module
  module 2 fcx-xfp-1-port-10g-module
  module 3 fcx-cx4-1-port-10g-module
stack unit 4
  module 1 fcx-48-port-copper-base-module
  module 2 fcx-cx4-1-port-10g-module
  module 3 fcx-xfp-1-port-10g-module
  priority 128
stack enable
!

```

For ICX devices, with stacking enabled, for example:

```

stack unit 1
  module 1 icx6610-24p-poe-port-management-module
  module 2 icx6610-qsfp-10-port-160g-module

```

```

module 3 icx6610-8-port-10g-dual-mode-module
priority 128
stack-trunk 1/2/1 to 1/2/2
stack-trunk 1/2/6 to 1/2/7
stack-port 1/2/1 1/2/6
stack unit 4
module 1 icx6610-48p-poe-port-management-module
module 2 icx6610-qsfp-10-port-160g-module
module 3 icx6610-8-port-10g-dual-mode-module
priority 100
stack-trunk 4/2/1 to 4/2/2
stack-trunk 4/2/6 to 4/2/7
stack-port 4/2/1 4/2/6
stack unit 5
module 1 icx6610-48-port-management-module
module 2 icx6610-qsfp-10-port-160g-module
module 3 icx6610-8-port-10g-dual-mode-module
priority 128
stack-trunk 5/2/1 to 5/2/2
stack-trunk 5/2/6 to 5/2/7
stack-port 5/2/1 5/2/6
stack enable

```

Syntax: `show running-config`

Table 52 describes the output from the **show running-config** command.

TABLE 52 Field descriptions for the `show running-config` command

Field	Description
stack unit <#>	The stack identification number for this unit.
module <#>	Identifies the configuration for modules on this unit.
priority	Indicates that a priority has been assigned to this stack unit
stack-trunk	Indicates the trunk configuration ¹

1. For ICX devices, software version 7.3 always shows two primary ports of two stacking trunks. The user cannot define just one stacking port as for FCX devices.

Displaying configured stacking ports

The stacking ports may display in the output from the **show running-config** command in three different ways.

1. When stacking is enabled, the output shows both stacking ports.

```

stack unit 1
module 1 fcx-24-port-management-module
module 2 fcx-cx4-2-port-10g-module
module 3 fcx-xfp-1-port-10g-module
stack-port 1/2/1 1/3/1

```

2. When stacking is not enabled, neither stacking port is displayed.

```

stack unit 1
module 1 fcx-24-port-management-module
module 2 fcx-cx4-2-port-10g-module
module 3 fcx-xfp-1-port-10g-module

```


3. If one stacking port is configured, that port will be displayed whether or not stacking is enabled.

```
stack unit 1
  module 1 fcx-24-port-management-module
  module 2 fcx-cx4-2-port-10g-module
  module 3 fcx-xfp-1-port-10g-module
stack-port 1/3/1
```

Displaying software version information

The **show version** command shows the software version that the stack is running. Note that the last line of this output shows the bootup ID and role for this unit. Output resembles the following.

```
Brocade(config)# show version
SW: Version 05.0.02T7e1 Copyright (c) 2009 Brocade Communications Systems, Inc.
   Compiled on Jul 23 2008 at 02:38:03 labeled as FCX05002
   (3054675 bytes) from Primary fcx05002.bin
STACKID 1: compiled on Jul 23 2008 at 02:38:03 labeled as FCX05000
           (3054675 bytes) from Primary fcx05000.bin
STACKID 2: compiled on Jul 23 2008 at 02:38:03 labeled as FCX05000
           (3054675 bytes) from Primary fcx05000.bin
STACKID 3: compiled on Jul 23 2008 at 02:38:03 labeled as FCX05000
           (3054675 bytes) from Primary fcx05000.bin
   BootROM: Version 04.0.00T7e5 (FEv2)
HW: Chassis FCX648
=====
STACKID 1: SL 1: FCX-24G 24-port Management Module + PoE
           Serial #: PR11060248
           P-ASIC 0: type D804, rev 01
=====
STACKID 1: SL 2: FCX-2XGC 2-port 10G Module (2-CX4)
=====
STACKID 1: SL 3: FCX-1XG 1-port 10G Module (1-XFP)
=====
STACKID 2: SL 1: FCX-48G 48-port Management Module
           Serial #: AN07510010
           P-ASIC 0: type D804, rev 01
           P-ASIC 1: type D804, rev 01
=====
STACKID 2: SL 2: FCX-1XG 1-port 10G Module (1-XFP)
=====
STACKID 2: SL 3: FCX-1XGC 1-port 10G Module (1-CX4)
=====
STACKID 3: SL 1: FCX-48G 48-port Management Module
           Serial #: AN07510269
           P-ASIC 0: type D804, rev 01
           P-ASIC 1: type D804, rev 01
=====
STACKID 3: SL 2: FCX-1XGC 1-port 10G Module (1-CX4)
=====
STACKID 3: SL 3: FCX-1XG 1-port 10G Module (1-XFP)
=====
           400 MHz Power PC processor 8248 (version 130/2014) 66 MHz bus
           512 KB boot flash memory
          30720 KB code flash memory
           128 MB DRAM
Monitor Option is on
The system uptime is 18 minutes 4 seconds
STACKID 1 system uptime 18 minutes 4 seconds
```

```

STACKID 2 system uptime 18 minutes 3 seconds
STACKID 3 system uptime 18 minutes 3 seconds
The system started at 21:08:51 GMT+00 Fri Jul 25 2008
The system : started=warm start    reloaded=by "reload"
My stack unit ID = 1, bootup role = active

```

Syntax: `show version`

Displaying stacking port interface information

The `show interfaces stack-ports` command displays information about the stacking ports on all stack units.

```
ICX6610-48 Router#show interfaces stack-ports
```

Port	Link	State	Dupl	Speed	Trunk	Tag	Pvid	Pri	MAC	Name
1/2/1	Up	Forward	Full	40G	None	No	N/A	0	748e.f834.1db5	
1/2/2	Up	Forward	Full	10G	None	No	N/A	0	748e.f834.1db6	
1/2/6	Up	Forward	Full	40G	None	No	N/A	0	748e.f834.1db7	
1/2/7	Down	None	None	None	None	No	N/A	0	748e.f834.1db8	
2/2/1	Down	None	None	None	None	No	N/A	0	0000.0000.0000	
2/2/2	Down	None	None	None	None	No	N/A	0	0000.0000.0000	
2/2/6	Down	None	None	None	None	No	N/A	0	0000.0000.0000	
2/2/7	Down	None	None	None	None	No	N/A	0	0000.0000.0000	
3/2/1	Down	None	None	None	None	No	N/A	0	748e.f834.266d	
3/2/2	Up	Forward	Full	10G	None	No	N/A	0	748e.f834.266e	
3/2/6	Up	Forward	Full	40G	None	No	N/A	0	748e.f834.266f	
3/2/7	Up	Forward	Full	10G	None	No	N/A	0	748e.f834.2670	
5/2/1	Down	None	None	None	None	No	N/A	0	748e.f834.11ad	
5/2/2	Up	Forward	Full	10G	None	No	N/A	0	748e.f834.11ae	
5/2/6	Up	Forward	Full	40G	None	No	N/A	0	748e.f834.11af	
5/2/7	Down	None	None	None	None	No	N/A	0	748e.f834.11b0	

Syntax: `show interfaces stack-ports`

[Table 53](#) describes the fields displayed by the `show interfaces stack-ports` command.

TABLE 53 Field descriptions for the `show interfaces stack-ports` command

Field	Description
Port	The stack identification number for this unit.
Link	Identifies the configuration for modules on this unit.
State	Indicates that a priority has been assigned to this stack unit
Dupl	Indicates whether the port is configured as half or full duplex
Speed	Indicates the port speed
Trunk	Indicates whether the port is part of a trunk
Tag	Indicates whether the port is tagged or untagged
P	Port priority
MAC	The MAC address of the port
Name	An optional name assigned to the port

NOTE

If a unit is provisional (reserved; and does not have a physical unit associated with the unit ID), its interface MAC address shows as 0000.0000.0000

Displaying stacking port statistics

The **show statistics stack-ports** command displays information about all stacking ports in an IronStack topology.

```
Brocade# show statistics stack-ports
```

Port	In Packets	Out Packets	In Errors	Out Errors
1/2/1	22223	4528	0	0
1/2/2	35506	3844	0	0
2/2/1	3161	34173	0	0
2/2/2	24721	3676	0	0
3/2/1	3048	23881	0	0
3/2/2	13540	2857	0	0
4/2/1	2862	13537	0	0
4/2/2	3626	3184	0	0
5/2/1	3183	3621	0	0
5/2/2	3265	13508	0	0
6/2/1	14020	3655	0	0
6/3/1	3652	17705	0	0
7/2/1	17705	3658	0	0
7/3/1	4047	21802	0	0
TOTAL	154559	153629	0	0

Syntax: show statistics stack-ports

[Table 54](#) describes the fields displayed by the **show statistics stack-ports** command.

TABLE 54 Field definitions for the **show statistics stack-ports** command

Field	Description
Port	The stack identification number for this unit.
In Packets	The number of incoming packets on this port
Out Packets	The number of outgoing packets on this port
In Errors	The number of incoming errors on this port
Out Errors	The number of outgoing errors on this port

Displaying stacking topology

The **show stack connection** command displays the topology and prints out a detailed connection report. It also prints connection errors or hardware failures, as shown in the following example output from a Brocade ICX 6610 switch.

```
Brocade#show stack connection
Probing the topology. Please wait ...
Brocade#
    active
      +---+      +---+      +---+      +---+      +---+      +---+
=2/1| 4 |2/6==2/6| 3 |2/1==2/1| 2 |2/6==2/6| 1 |2/1==2/1| 7 |2/6==2/6| 6 |2/1=
| +---+      +---+      +---+      +---+      +---+      +---+
|
```

```

|
|
-----standby-----
+----+
2/1| 5 |2/6=
+----+

```

```

trunk probe results: 7 links
Link 1: u7 -- u1, num=5
    1: 1/2/1 (T0) <----> 7/2/1 (T0)
    2: 1/2/2 (T0) <----> 7/2/2 (T0)
    3: 1/2/3 (T0) <----> 7/2/3 (T0)
    4: 1/2/4 (T0) <----> 7/2/4 (T0)
    5: 1/2/5 (T0) <----> 7/2/5 (T0)
Link 2: u2 -- u1, num=5
    1: 1/2/6 (T1) <----> 2/2/6 (T1)
    2: 1/2/7 (T1) <----> 2/2/7 (T1)
    3: 1/2/8 (T1) <----> 2/2/8 (T1)
    4: 1/2/9 (T1) <----> 2/2/9 (T1)
    5: 1/2/10(T1) <----> 2/2/10(T1)
Link 3: u3 -- u2, num=5
    1: 2/2/1 (T0) <----> 3/2/1 (T0)
    2: 2/2/2 (T0) <----> 3/2/2 (T0)
    3: 2/2/3 (T0) <----> 3/2/3 (T0)
    4: 2/2/4 (T0) <----> 3/2/4 (T0)
    5: 2/2/5 (T0) <----> 3/2/5 (T0)
Link 4: u4 -- u3, num=5
    1: 3/2/6 (T1) <----> 4/2/6 (T1)
    2: 3/2/7 (T1) <----> 4/2/7 (T1)
    3: 3/2/8 (T1) <----> 4/2/8 (T1)
    4: 3/2/9 (T1) <----> 4/2/9 (T1)
    5: 3/2/10(T1) <----> 4/2/10(T1)
Link 5: u5 -- u4, num=5
    1: 4/2/1 (T0) <----> 5/2/1 (T0)
    2: 4/2/2 (T0) <----> 5/2/2 (T0)
    3: 4/2/3 (T0) <----> 5/2/3 (T0)
    4: 4/2/4 (T0) <----> 5/2/4 (T0)
    5: 4/2/5 (T0) <----> 5/2/5 (T0)
Link 6: u6 -- u5, num=5
    1: 5/2/6 (T1) <----> 6/2/1 (T0)
    2: 5/2/7 (T1) <----> 6/2/2 (T0)
    3: 5/2/8 (T1) <----> 6/2/3 (T0)
    4: 5/2/9 (T1) <----> 6/2/4 (T0)
    5: 5/2/10(T1) <----> 6/2/5 (T0)
Link 7: u7 -- u6, num=5
    1: 6/2/6 (T1) <----> 7/2/6 (T1)
    2: 6/2/7 (T1) <----> 7/2/7 (T1)
    3: 6/2/8 (T1) <----> 7/2/8 (T1)
    4: 6/2/9 (T1) <----> 7/2/9 (T1)
    5: 6/2/10(T1) <----> 7/2/10(T1)
CPU to CPU packets are fine between 7 units.

```

Table 55 describes the fields displayed by the **show stack connection** command.

TABLE 55 Field definitions for the **show stack connection** command

Field	Description
T0	Identifies Trunk 0
T1	Identifies Trunk 1

Syntax: show stack connection

Adding, removing, or replacing units in an IronStack

The following sections describe how to add, remove, or replace units in an IronStack. The recommended method is to connect units to the stack before you supply power to the units; however, you can also connect powered units.

Installing a new unit in an IronStack using secure-setup

This method can be applied to clean units, or units that have existing configurations.

1. Connect the new unit to the stack by connecting the 10 Gbps stacking ports.
2. Run secure-setup on the Active Controller and assign an ID to the new unit. The Active Controller will reset the new unit.
3. Once the new unit boots and joins the stack, do a **write memory** on the Active Controller.

Installing a new unit using static configuration

If the new unit is a clean unit and the connection is sequential you can add it using the static setup process.

1. Enter the module configuration of the new unit into the Active Controller configuration.
2. Connect the new unit to the stack using the 10Gbps stacking ports. The sequence in which you connect the unit must match that of the Active Controller configuration. The Active Controller automatically resets the unit.
3. Once the new unit boots and joins the stack, do a **write memory** on the Active Controller. You should see the following message.

```
Done hot swap: Set stack unit 3 to Fully-Operational:16
```

Configuration notes

Configuration on a new unit can be accomplished in three ways:

- If the Active Controller has no configuration information for the new unit, it learns the new unit's configuration. This is a dynamic configuration and will disappear if the new unit leaves the stack. In order for the configuration to stay on the Active Controller (to make it a static configuration), you must do a **write memory** on the Active Controller.
- If the Active Controller has configuration information for a new unit, and it matches the base module (module 1) of the new unit, no action is necessary. If configuration information for non-base modules on the new unit does not match the information on the Active Controller, the Active Controller learns the configuration for the new unit module types and merges it with the information it has for the base module. This merged configuration remains static and will stay on the Active Controller even if the new unit leaves the stack.
- If the Active Controller has configuration information for the new unit, but it does not match the base module of the new unit, a configuration mismatch can occur where the configuration related to this unit is removed even after the mismatch is resolved. Refer to [“Recovering from a stack unit mismatch”](#) on page 323 for more information.

Removing a unit from an IronStack

To remove a unit from the stack, disconnect the cables from the stacking ports. This can be done whether the units are powered-on or powered-off. When you remove a unit that is powered-on, it is still in stacking enabled mode. To remove the stacking files, enter the **stack unconfigure me** or **stack unconfigure clean** command. When the unit reboots, it will operate as a standalone unit. Refer to “[Unconfiguring an IronStack](#)” on page 287.

When a unit is removed from a stack, the Active Controller deletes this unit configuration if it is dynamically learned. Refer to “[Brocade IronStack terminology](#)” on page 237 for definitions of static and dynamic configurations.

Replacing an IronStack unit (unit replacement)

Replacing with a clean IronStack unit

If the stack unit ID numbering is sequential, you can easily swap a failed unit with an identical clean unit using this procedure.

1. Remove the old unit from the stack.
2. Make sure that the hardware (module) configuration of the replacement unit is identical to that of the failed unit.
3. Connect the new unit to the stack using the same stacking ports used by the old unit.
4. If the configuration of the replacement unit matches the configuration on the Active Controller, the Active Controller resets the new unit, which automatically becomes active in the stack, and the stack retains its original topology.

Replacing with multiple clean IronStack units

If you are replacing multiple old units with clean units, the Active Controller replaces the unit with the lowest ID first. For example, if you remove units 5 and 6 (which are FCX624P-STK devices), the Active Controller assigns ID 5 to the first new FCX624P-STK device you install. If you wanted this particular unit to replace unit 6, instead of unit 5, you must use secure-setup.

You must use secure-setup if the replacement is not a clean unit, the connection is not sequential, or you do not want the Active Controller to trigger an automatic replacement. Use the following steps.

1. Remove the old stack unit from the stack
2. Connect the new unit to the existing stack using the same stacking ports used by the old unit.
3. Run secure-setup to select the ID of the old unit for the new unit. The Active Controller resets the unit, and it joins the stack.

NOTE

Adding, removing or replacing a stack unit which is not at the end of linear topology may cause the other units in the stack to reset if these units lose their path to the Active Controller during the process. Adding or removing a unit in a ring topology should not cause the other units to reset because each unit can still find a path to the Active Controller.

Moving a unit to another stack

Moving a member from a stack and to another stack can result in non-sequential ID assignment. The Active Controller will honor the new unit original ID if that ID is not being used in the new stack. The Active Controller will assign a new ID if the original ID is already being used. To prevent non-sequential stack ID assignments, configure the unit that is moving as a clean unit before adding it to the new stack.

Removing an Active Controller from a powered stack

To remove an Active Controller from a powered stack, disconnect the Active Controller. The Standby Controller waits for 30 seconds and then assumes the role of Active Controller. A single Active Controller device functions as a standalone unit even it is still stacking-enabled. You do not have to issue a **stack unconfigure me** command for an Active Controller.

Renumbering stack units

You can use secure-setup to renumber stack units in a previously constructed stack. In the following example, three units make up a stack, yet two of the units are numbered 5 and 6 (the Active Controller is numbered 1). Since this stack is only going to contain 3 units, you can renumber the other units so that they are unit 2 and unit 3.

The most effective way to number your stack members is sequentially. You can skip numbers, but they should still be sequential, from 1 to 8. Sequential numbering makes it easy to replace stack units, and easier to troubleshoot issues.

NOTE

In a ring topology, 1, 2, 4, 5, and 1, 5, 4, 2 are both sequential.

Example

```
Brocade# stack secure-setup
Brocade#Discovering the stack topology...

Available UPSTREAM units
Hop(s)  Type      Mac Address
1       FCX624    0012.f2d5.2100
2       FCX624    001b.ed5d.9940

Enter the number of the desired UPSTREAM units (1-2)[1]: 2

Selected topology:
Active id  Type      Mac Address
1         FCX624    0012.f239.2d40

Selected UPSTREAM units
Hop(s)  id  Type      Mac Address
1       5  FCX624    0012.f2d5.2100
2       6  FCX624    001b.ed5d.9940

Do you accept the unit ids? (y/n)? : n

Enter an unused id for the UPSTREAM FCX623 unit a 1 hop(s) (1-8)[5]: 2
Enter an unused id for the UPSTREAM FCX624 unit at 2 hop(s) (1-8) [6]: 3
Brocade# Election, was active, no role change, assigned-ID=1
reset unit 2: diff bootup id=5
```

```
reset unit 3: diff bootup id=6
Election, was active, no role change, assigned-ID=1
```

```
Brocade# show stack
```

ID	Type	Role	Mac Address	Pri	State	Comment	
1	S	FCX624	active	0012.f239.2d40	128	local	Ready
2	S	FCX624	standby	0012.f2d5.2100	0	remote	Ready
3	S	FCX624	member	001b.ed5d.9940	0	remote	Ready

Configuration notes for renumbering stack units

- Renumbering may result in the removal of a unit configuration if the stack unit base module does not match the configuration on the Active Controller. However, secure-setup renumbering never changes the interface configuration. For example, if you switch the IDs of identical units 2 and 3, the Active Controller does not change 2/1/5 to 3/1/5 and vice versa.
- If the configuration for the ID you select for a specific unit does not match the configuration on that unit, secure-setup will change the static configuration into a dynamic configuration so it can be overwritten by the learned configuration.
- When swapping IDs for two or more identical units - for example, if units 2, 3, and 4 are identical, changing 2 to 3, 3 to 4, and 4 to 2 will not affect the configurations of the units except that the units will reset and assume the new IDs.
- If you swap IDs for two units that are not identical - for example, unit 2 is an FCX648 and unit 3 is an FCX624, you may cause a configuration mismatch. If this happens, the Active Controller removes the configurations and resets both units. When both units boot with new IDs, the Active Controller learns their module types and creates new *unit* configurations for both. However, all *interface configuration* information related to units 2 and 3 is gone.
- When you renumber identical units using secure-setup, the configurations are not mapped to the new units (since the configurations match exactly). However, if you switch the IDs of units that are not identical, a configuration mismatch occurs. Refer to [“Recovering from a stack unit mismatch”](#) on page 323
- When you assign an unused ID to a stack unit, the unit is reset with the new ID. All unit and interface configuration information related to the old stack ID is deleted. The Active Controller learns the configuration for the new unit (instead of creating interface configuration for the new unit).
- Release 5.0 does not support user changes to the Active Controller ID.
- Secure-setup does not swap configuration information for units that have had their IDs changed. For example, it does not change the 2/1/3 interface configuration or VLAN membership information into 3/1/3 information if the unit ID changes from 2 to 3.
- If the configuration for a unit being replaced does not match the new unit type, the Active Controller removes the unit configuration and associated interface configuration.
- All learned configurations due to mismatches or the addition of new units are dynamic configurations. To convert them into static configurations, do a **write memory** to preserve the configurations if a unit is removed from the stack.

Syslog, SNMP, and traps for stack units

Syslog messages from stack units are forwarded to, and can be viewed from, the Active Controller.

All stack units support SNMP gets, sets, and traps, which are managed by the Active Controller. An SNMP trap is sent from a stack unit to the stack Active Controller, and forwarded from the Active Controller to an SNMP-configured server. An external network management station can execute SNMP gets and sets for MIBs and collect information about any port on the stack.

SNMP traps can be configured for the insertion or removal of a stack unit or uplink module, and for optic identification.

For more information about Syslog messages, refer to [Appendix A, “Syslog messages”](#).

Configuring SNMP for an IronStack

SNMP server and feature configuration is the same for an IronStack as it is for standalone units. In an IronStack, SNMP gets and sets are processed by the Active Controller for the Standby Controller and all stack members. SNMP traps generated by the Standby Controller and stack members are propagated to the configured SNMP server through the Active Controller. For more information about how to configure an SNMP server for FastIron devices, refer to [Chapter 10, “SNMP Access”](#).

SNMP engine IDs for stackable devices

For Brocade stacking devices, if an engine ID is not manually created or a stack MAC address is not specified and saved, the stack will lose its engine ID if the Active Controller fails and the Standby Controller takes over, because the Standby Controller creates a new engine ID at bootup. To prevent this from happening, you will need to either create a new engine ID or create a new stack MAC address to ensure that the engine ID is saved to the startup configuration. This should be done before the SNMPv3 user is created.

If a new Active Controller is elected (for example, the Standby Controller becomes the Active Controller) you will see the following results:

- If you have configured the engine ID saved it to the startup configuration file, the new stack configuration will use the saved engine ID.
- If you have not configured an engine ID, but a stack MAC address is configured, the new stack configuration will retain the original engine ID since it is based on the stack MAC address.
- If you have not configured an engine ID, and no stack MAC address is configured, the new stack configuration will use the default engine ID, which is based on its own management MAC address of the new Active Controller. Since the engine ID will have changed, any SNMPv3 Clients will need to be reconfigured with the new engine ID.

IronStack troubleshooting

The most common reason for an unsuccessful stack build is either a software configuration mismatch, a hardware configuration mismatch, or a combination of both.

The following sections describe common troubleshooting procedures for an IronStack.

Troubleshooting an unsuccessful stack build

If you are unable to build a stack, (for example, the **show stack** command does not display any stack units), perform the following steps.

1. Enter the **show run** command on each unit to make sure the configuration contains “stack enable”. If it does not, enter the **stack enable** command on the unit. Before a stack is formed, you can still access the console port on each device. Once a stack is successfully formed, you are redirected to the Active Controller.

NOTE

If you are building a stack using secure-setup, you do not have to enter stack enable on each unit.

2. Check that all of your stacking port connections are secure and working properly. Enter the **show interface stack** on each device to confirm that the stacking port links are up and the ports are in the forward state.

```
Brocade# show interfaces stack
Port  Link  State   Dupl Speed  Trunk  Tag  P  MAC      Name
1/2/1 Up    Forward Full 10G    None   No   1 0012.f2eb.a902
1/2/2 Up    Forward Full 10G    None   No   1 0012.f2eb.a904
```

3. Confirm that all of the devices are running the same software image
4. Use the **show log** command to display any IPC version mismatch messages. These messages appear in one minute when receiving mismatched probe packets, and then once every 10 minutes.
5. Use the **show stack ipc** command to see if any traffic has been sent or received. Enter **clear stack ipc** to clear the traffic statistics and then enter **show stack ipc** again so you can easily see differences in traffic flow. The following output is from a Brocade ICX 6610 switch.

```
Brocade# show stack ipc
V15, G1, Recv: SkP0: 3749372, P1: 3756064, MAIL: 184291175, sum: 191796611,
t=457152.2
Message types have callbacks:
 1 : Reliable IPC message 2 : Reliable IPC atomic 4 : fragmentation, jumbo
 5 : probe by mailbox    6 : rel-mailbox      7 : test ipc
 8 : disable keep-alive  9 : register cache   10: ipc dnld stk
11: chassis operation   12: ipc stk boot     13: Rconsole IPC message
14: auth msg           15: ipc erase flash  16: unconfigure
17: ipc stk boot       18: ss set           19: sFlow IPC message
21: SYNC download reques 23: SYNC download 1 spec 28: SYNC client hello
30: SYNC dy chg error    32: active-uprintf    33: test auth msg
34: probe KA            39: unrel-mailbox     40: trunk-probe

Send message types:
 [1]=2342639, [4]=44528, [5]=961830, [6]=37146,
 [9]=73104634, [11]=137082, [14]=487007, [20]=2304,
 [22]=1395, [25]=23, [26]=1901701, [29]=415888,
 [34]=1827543, [39]=30451, [40]=289420,

Recv message types:
 [1]=2016251, [4]=1352759, [5]=470884, 475144,
 [6]=114459, 114572, [9]=367644144, [11]=1785229,
 [14]=973285, 974177, [21]=1395, [30]=25,
 [34]=912972, 914086, [39]=973492, 973440, [40]=700313,

Statistics:
 send pkt num      : 34068433,   recv pkt num      : 191796609,
 send msg num      : 79756048,   recv msg num      : 379902767,
 send frag pkt num : 22264,      recv frag pkt num : 493860,
 pkt buf alloc     : 34068433,

Reliable-mail      send success recveive duplic
```

```

target ID          1          1          0          0
target MAC         15230      15230      0          0
unrel target ID    7615              0
There is 1 current jumbo IPC session

```

Possible errors:

```
*** recv from non-exist unit 2 times: unit 5
```

If the send message types: field is empty, it means that stack enable has not been configured. If the number of Recv IPC packets increases, but there are no Recv message types, then the packets are being dropped for various reasons, including the wrong IPC version, or a checksum error. The Possible errors field will list reasons for packet loss.

NOTE

A small “***state not ready” count is normal, but if it continues to increase a problem is indicated.

6. If the results of a **show stack** command show other stack members, but lists them as non-operational, this could be due to an image mismatch, or a configuration mismatch. In the event of an image mismatch, you can download the correct images to the entire stack from the Active Controller. Refer to [“Configuration mismatch for stack units”](#) on page 320 for more information about configuration mismatches.

NOTE

If your intended stacking ports are connected in a ring topology, they will not all appear to be in the forwarding state because of spanning tree, but secure-setup can still build the stack.

7. If you run out of flash memory while doing a **write memory**, your stack devices may contain very large startup-config.v4 or startup-config.old files, which are preserved for recovery purposes (refer to [“Unconfiguring an IronStack”](#) on page 287 for more information). If you do not need these files, you can delete them using the **flash delete** command. Enter the **show dir** command to see all flash files.
8. Check to be sure you do not have any stacking to non-stacking connections. If you see the following message.

```

Warning! Proc ??? packet in 2m from 0012.f2222.8300, Wrong dev/port: dev=4,
port=18, DSA=4971100 497--E
You might have stacking to non-stacking port connections

```

This indicates that you may have a connection between a stacking port and a non-stacking port. This message will appear every 10 minutes after the first display. If you see this message once only, and your connections are correct, your stack should be operating properly. Only repeat displays of this message indicate a problem.

Troubleshooting a stacking upgrade

After you upgrade your device to support stacking, restart the device with the upgraded software. If you encounter a problem at this step, make sure the memory DIMM and stacking EEPROM are installed correctly. If they are not installed correctly, you may see output similar to the following.

```

FCX MEM size: 0x10000000
FCX Flash config....
FCX Boot Code Version 05.0.00
Enter 'b' to stop at boot....

```

Stack mismatches

```
BOOT INFO: load monitor from primary, size=103408
BOOT INFO: load image from primary.....
BOOT INFO: bootparam at 000543e8, mp_flash_size=002ee6c5
BOOT INFO: code decompression completed
BOOT INFO: branch to 00400100
Starting Main Task.....
*****
ERR: This software needs License PROM to be installed in the system
*****
System Reset!
```

If your memory DIMM is not installed correctly, you will see output similar to the following.

```
FCX Mem size: 0x8000000
Flash Config...
FCX Boot Code Version 05.0.00
Enter 'b' to stop at boot....
BOOT INFO: load monitor from primary, size = 103380
BOOT INFO: debug enabled!!
BOOT INFO: load image from primary...
BOOT INFO: bootparam at 00054338 mp_flash_size = 002flaeb
BOOT INFO: code decompression completed
BOOT INFO: branch to 00400100
Starting Main Task ...

*****
ERR: This software requires 256M memory to be installed in the system.
*****
System Reset!
```

Check your upgraded hardware for the following situations:

- EEPROM is installed incorrectly in the socket. Make sure Pin 1 on the EEPROM matches the Pin 1 hole in the socket.
- Make sure your memory DIMM is securely installed in the memory DIMM socket. Refer to the hardware installation guide or the instructions that came with your upgrade kit for more information.

Troubleshooting image copy issues

The **copy tftp flash** command copies the image to all stack units including the Active Controller. The **copy flash flash** command copies the image from the primary or secondary flash on the Active Controller to the primary or secondary flash image of a stack member, respectively. If you are unable to copy an image to one or more stack units, check the following:

- Make sure the unit is actually part of the stack. Use the **show stack** command.
- If a unit joins a stack after the **image copy** command was issued, you will need to copy the image to this unit separately.

Stack mismatches

When a stack mismatch occurs, the Active Controller can put any stack member into a non-operational state, which disables all of the ports except the stacking ports. Stack mismatches can occur for a variety of reasons, which are discussed in this section.

NOTE

The Active Controller can still download an image to the non-operational unit.

The Active Controller generates a log message whenever it puts a stack unit into a non-operational state. The following examples describe the types of mismatches and the related log message:

- **Advanced feature mismatch** - The Active Controller is enabled for advanced features (such as BGP) and the stack unit is not enabled.

```
Stack: Unit 2 00e0.1020.0100 doesn't have the matching advanced feature
privileges
```

- **Image mismatch** - A stack unit is running a different software image than that of the Active Controller.

```
Stack: Unit 2 00c0.1020.0100 image mismatch
```

- **Configuration mismatch** - The module configuration for a stack unit does not match the reserved configuration on the Active Controller.

```
Stack: Unit 2 00e0.1020.0100 config mismatch
```

- **Memory allocation mismatch** - The Active Controller does not have enough memory to accommodate the stack unit.

```
Stack: Malloc failure for unit 2.00e0.1020.0100
```

These mismatches are described in the following sections.

Image mismatches

Advanced feature privileges (FCX devices only)

Advanced feature privileges must be enabled to run advanced features such as BGP. Both Active and Standby units must be enabled for advanced features for these features to operate across the stack. A unit that is not enabled for these features is put into a non-operational state.

If the Active Controller is *not* enabled for advanced features, these features will not operate on the stack.

IronStack technology requires that all stack units run the same version of the software image. In cases where the software version differs, there are two levels of mismatch, major and minor.

Major mismatch for stack units

A major mismatch indicates an Interprocessor Communications (IPC)-related data structure change, or an election algorithm change, or that a version of the software that does not support stacking is installed on a unit. This can happen when the software undergoes a major change (such as a change from 05.0.00 to 05.1.00). When a major mismatch occurs, the system logs and prints a message similar to the following.

```
Warning! Recv 424 IPC in 1m from 0012.f21b.a900 e1/1/25: wrong version 5 !=6.
Please make sure all units run the same image.
```

In a major mismatch, the stack cannot be built and will not operate. You must download the correct version of the software to the mismatched units individually.

Minor mismatch for stack units

With a minor mismatch, an operating stack can still exist, but traffic is dropped from all ports except for the stacking ports for units with the mismatched software. You can download the correct image to the mismatched devices from the Active Controller. A minor software mismatch means that there is no IPC or election algorithm change, but there is a release version disparity. Minor software mismatches can happen with patch release upgrades. The system logs and prints a message similar to the following.

```
Warning! put stack unit 2 to non-operational reason=image mismatch
```

The **show stack** command displays output similar to the following.

```
Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID  Type   Role   Mac Address      Pri   State   Comment
1  S  FCX624  active  0012.f2eb.a900   128   local   Ready
2  S  FCX648  standby 00f0.424f.4243   0     remote NON-OP: image mismatch
3  S  FCX624  member  00e0.5201.0100   0     remote   Ready
```

If the configuration of a stack unit does not match the configuration of the Active Controller, the stack unit will not function. In this example, unit 2 is non-operational due to an image mismatch. To correct this situation, use the **copy flash flash** command (refer to [“Copying the flash image to a stack unit from the Active Controller”](#) on page 283).

Configuration mismatch for stack units

Generally, when a stack unit is added to or removed from the stack, its static configuration is not overwritten by the Active Controller. On the other hand, the Active Controller deletes the dynamic configuration for a unit if it leaves. (Refer to [Table 48](#) for definitions of static and dynamic configurations.)

A configuration mismatch occurs when the base module configuration for a replacement stack unit does not match the run time configuration on the Active Controller. If the configuration on the Active Controller is static, it cannot be overwritten by the new configuration, and a configuration mismatch occurs.

Configuration mismatches can happen during manual setups, or when moving a unit from one stack to another stack or a unit's ID is changed by Secure Setup.

When you renumber identical units using secure-setup, the configurations are not mapped to the new units (since they match exactly). However, if you switch the IDs of units that are not identical, a configuration mismatch occurs.

When a configuration mismatch occurs, port-related functions on all ports are disabled on the mismatched unit (except for the stacking ports). All other functions are unaffected. For example, the Active Controller can still copy the unit's image or reset the unit. Refer to [“Recovering from a stack unit mismatch”](#) on page 323.

Auto Image Copy for stack units

The Auto Image Copy feature ensures that all units in a stack are running the same flash image after a stack merge. This feature also enables automatic reload of the stack units. It prevents the image mismatch that occurs when one or more member units join the stack with a different running image and signature than that of the master and standby units.

Auto Image Copy is enabled by default on Brocade devices and the user does not have to manually copy the master's running image to the mismatched members.

This feature is available on the following Brocade devices:

- Brocade FCX Series (FCX) Stackable Switch
- Brocade ICX 6610 Series (ICX 6610) Stackable Switch
- Brocade ICX 6430 Series (ICX 6430) Stackable Switch
- Brocade ICX 6450 Series (ICX 6450) Stackable Switch

NOTE

To manually copy the master image to the stack units, refer to [“Copying the flash image to a stack unit from the Active Controller”](#) on page 283 and [“Reloading a stack unit”](#) on page 283 for more information.

Auto Image Copy limitations

The following limitations apply to the Auto Image Copy feature:

- This feature is applicable to those stack units that are in a non-operational image mismatch state only.
- Auto Image Copy does not work if the image version of the IPC is different from the stack unit version in the case of a major image mismatch.
- If a stack unit with a newer image is merged with a stack running an older version of the software, the newly formed stack will still be running the older version of the software.
- Auto Image Copy does not work if the user has copied a different image into the flash, creating a mismatch in versions between the flash and the running image on the active unit of the stack.

Disabling Auto Image Copy

Auto Image Copy is enabled by default. However, if you wish to disable this feature, enter the following command in global configuration mode.

```
Brocade(config)# image-auto-copy disable
```

Syntax: no image-auto-copy disable

Run the **no image-auto-copy disable** command to enable Auto Image Copy. This restarts the Auto Image Copy immediately and ensures that all stack units have the same image.

NOTE

You can run the **show running-config** or the **show stack detail** command to see if Auto Image Copy is disabled.

Verifying Auto Image Copy

Use the **show stack detail** command to verify if there are any units in the mismatch state after the stack is formed. Refer to [“Displaying stack information”](#) on page 293 for more information about the command.

Memory allocation failure

A memory allocation (malloc) failure occurs when the Active Controller does not have enough memory to run a stack unit. This failure may occur if you configure large numbers (for example, 4 K of VLANs, or STP instances (for example, 255).in the router image. This message means that the Active Controller is low on memory after allocating these resources and does not have enough remaining memory to control a stack member. You can correct this by reducing the number of VLANs or STP instances.

NOTE

After you make configuration changes such as number of VLANs or STP instances, you must reset the stack.

Recovering from a stack unit mismatch

When a configuration mismatch occurs, the Active Controller logs and displays a configuration mismatch message, and puts the mismatched unit into a non-operational state. In the following example, the original stack unit 3 has failed, and a replacement unit has been installed that does not match the configuration of the original unit. You should see the following message.

```
Warning! put stack unit 3 to non-operational reason= config mismatch
```

Complete the following steps to recover from a configuration or image mismatch.

1. Enter the **show stack** command to see the status of the stack, and a **show running-config** command to see the configurations of the stack units.

```
Brocade# show stack
alone: standalone, D: dynamic config, S: static config
ID Type    Role    Mac Address    Pri State    Comment
1  FCX624  active  0012.f2eb.a900 128 local    Ready
2  FCX648  member  00f0.424f.4243  0 remote  Ready
3  FCX624  standby 00e0.5201.0100  0 remote  NON-OP: config mismatch
```

```
Brocade# show running config
stack unit 1
  module 1 fcx-24-port-copper-base-module
  module 3 fcx-cx4-1-port-10g-module
  module 4 fcx-xfp-1-port-10g-module
  priority 128

stack unit 2
  module 1 fcx-24-port-management-module
  module 3 fcx-xfp-1-port-10g-module

stack unit 3
  module 1 fcx-48-port-copper-base-module
  module 2 fcx-cx4-1-port-10g-module
  module 3 fcx-cx4-1-port-10g-module
stack enable
```

2. To resolve the mismatch, you must remove the configuration for stack unit 3. Use the following command in configuration mode:

```
Brocade(config)# no stack unit 3
```

This removes all configuration related to Unit 3.

If you are unable to remove the configuration because of a multi-slot trunk configuration, you must first manually remove the multi-slot trunk configuration.

3. When you have successfully deleted the mismatched stack unit, a re-election is triggered, and the Active Controller learns the correct module configuration from the Standby Controller or from other stack members. No reload is required.

Complete the following steps to recover from an image mismatch.

1. Use the **copy flash flash** command to replace a mis-matched image with the correct image. Refer to [“Copying the flash image to a stack unit from the Active Controller”](#) on page 283.
2. Reset the unit. After the reset, the unit will contain the new image and the mis-match condition will not exist. To verify, use the **show stack** command.

Troubleshooting secure-setup

Secure-setup can be used to form linear and ring stack topologies. For information about the procedure, refer to [“Scenario 1 - Three-member IronStack in a ring topology using secure-setup”](#) on page 250. During this procedure, if secure-setup does not detect all the units that should be detected, perform the following checks:

- Make sure that all the cables are properly connected
- Make sure that all the relevant ports are in UP state
- Make sure that all the units are running the same image
- Make sure that you issue the **stack enable** command only on the unit that will serve as the Active Controller
- Make sure that **stack disable** is not configured on any prospective members
- Make sure that the connection is sequential (refer to [“Brocade IronStack terminology”](#) on page 237, Sequential Connection)

If secure-setup times out (this may happen due to inactivity), you will not be able to make any changes in your configuration or stack topology until you restart the session by entering the **stack secure-setup** command.

The unit discovery process is triggered when secure-setup is initiated. However, if the stack unit is placed in a topology where another unit in the stack is already running the discovery process, the current discovery process is terminated. If this is the case, you will see a message similar to the following.

```
"Topology discovery is already in progress originated from <mac-address>. Please try later."
```

This means a discovery process is already active and was initiated from the unit with the `<mac-address>` mentioned in the message. You will need to re-issue secure-setup.

If there is already an active discovery process, secure-setup may not discover all the intended units. If this is the case, you will need to restart the secure-setup process.

Troubleshooting unit replacement issues

If you are unsuccessful in building a stack using the automatic setup process (refer to [“Scenario 2 - Three-member IronStack in a ring topology using the automatic setup process”](#) on page 254), or adding or replacing a unit in a stack, consider the following issues:

- Make sure that the number of units in your stack does not exceed the maximum of 8
- Make sure that the replacement unit is a clean unit (does not contain a startup-config.txt file)
- Make sure that the replacement unit running configuration does not contain “stack enable”
- Make sure the replacement unit running configuration does not contain “stack disable”
- Make sure that the configurations of the stack ports on the Active Controller match the physical connections to the unit

More about IronStack technology

This section discusses stacking technology in greater detail than the information presented in Section 1.

Configuration, startup configuration files, and stacking flash

Stacking system behavior is defined by the run time configuration, which can be displayed using the **show run** command. The **write memory** command stores the run time configuration in a flash file called *startup-config.txt*. During bootup, the system reads and applies the *startup-config.txt* file to the run time configuration. The *startup-config.txt* file can be shown using the **show config** command.

The stacking system installs a *stacking.boot* file on each unit that tells the unit what its role is during the boot process. The *stacking.boot* file is generated whenever there is an election that defines the roles for all units.

When an Active Controller is booted, or a **write memory** command is issued, the Active Controller synchronizes its *startup-config.txt* file to every stack unit. The original *startup-config.txt* files in the Standby Controller and other stack members are renamed to *startup-config.old*. If you issue the **stack unconfigure me** command on the Standby Controller or stack member directly, these units will recover their original *startup-config.txt* files and reboot as standalone devices. If you enter the **stack unconfigure all** command from the Active Controller all devices will recover their old *startup-config.txt* files and become standalone devices. When this happens, the *startup-config.old* file is renamed to *startup-config.txt*, and the *stacking.boot* file is removed. For more information, refer to “[Unconfiguring an IronStack](#)” on page 287.

Whenever a change is made to a stack unit's configuration, such as priority, (which could affect stack elections) an election is held, and the result is written into the *stacking.boot* file. A prompt message appears on the console that suggests you do a **write memory**. For an Active Controller role change to take effect, you will need to reset the entire stack.

If you do **not** do a **write memory**, and reset the stack, the stack units will continue to operate in their roles as defined by the *stacking.boot* file. After the reset, each unit readjusts based on the current run time configuration. However, you may get different results depending on what has not been saved. If you have renumbered the stack unit IDs, you may see a configuration mismatch, because your changes no longer match the Active Controller configuration.

If you change priorities to elect an Active Controller, the new Active Controller will assume its role after a reboot whether you have done a **write memory** or not. If you do not save your priority change before the next reboot, the reboot will trigger an election that may result in a different winner based on the priority in the unsaved configuration. The new winner assumes its role after the next reboot.

If you change the stacking port configuration and do not save your changes, you may encounter connectivity errors. To recover from a configuration error, run Secure Startup to define the correct stacking port.

NOTE

You should always do a **write memory** after making stacking-related configuration changes such as priority and stacking ports. If you do not want to keep the changes, change the configuration back to the previous version, and do a **write memory**. Do not discard configuration changes by using the reset without a **write memory**.

IronStack topologies

Brocade IronStack technology supports both linear and ring stack topologies. Because the unicast switching follows the shortest path in a ring topology, this topology offers the strongest redundancy. When the ring is broken, the stack recalculates the forwarding path and resumes the flow of traffic within a few seconds. In a ring topology, all stack members must have two stacking ports, however, in a linear topology, both end units use only one stacking port, leaving the other port available as a data port. To see an illustrated example of each topology, refer to [“Brocade IronStack topologies”](#) on page 239.

Port down and aging

If a unit is powered down, or the stacking link is removed, the system immediately detects the port down and knows that its neighbor is gone. That unit is immediately removed from the Active Controller. If a unit is gone or no longer stack-enabled, but its stacking link is still on, it will take 20 seconds to age the neighbor out. The following message will be logged and displayed.

```
Warning! my mac=00f0.424f.4243, age out up-stream
```

Brocade IronStack device roles and elections

There are three distinct roles played by units that are part of an IronStack:

- Active Controller
- Standby Controller
- Stack member

Active Controller

The Active Controller contains the saved and running configuration files for each stack member. The configuration files include the system-level settings for the stack, and the interface-level settings for each stack member, as well as MIB counters and port status. The Standby Controller also has a synchronized copy of the Active Controller startup config file for use in the event the Active Controller fails.

When a stack is formed, the console function for each stack member is automatically redirected to the Active Controller console. The Active Controller console port handles all stack management functions, as well as ping, Telnet sessions, and tftp image downloads for every stack member. If you connect to the console port on a stack member that is not the Active Controller, you are automatically directed through the console of the Active Controller.

The Active Controller synchronizes its start-up configuration with the Standby Controller and the rest of the stack members. You can recover the previous flash configuration of the Standby Controller and the stack members by issuing the **stack unconfigure** command. For an example of this command and the output generated, refer to [“Unconfiguring an IronStack”](#) on page 287.

The Active Controller may reset the rest of the stack members, if necessary. However, if the Active Controller itself must be reset because of a role or ID change, you must issue the **reset** command.

If the Active Controller fails, the Standby Controller waits 30 seconds, and then takes over as Active Controller, resetting itself and all other stack members. If the old Active Controller becomes operational, it may or may not resume its role as Active, depending on the configured priorities. If hitless stacking failover is enabled, the standby unit can take over immediately without reloading any unit.

Standby Controller

In addition to the Active Controller, another stack member is elected as the Standby Controller. After a default interval of 30 seconds, the Standby Controller takes over if the Active Controller fails. If hitless stacking failover is enabled, the standby unit can take over immediately without reloading any unit.

NOTE

Because it can take as long as 20 seconds to age out a neighbor, the Standby takeover period may last up to 50 seconds. Refer to [“Port down and aging”](#) on page 326.

The Standby Controller synchronizes its configuration with the Active Controller at each reset.

Bootup role

When a stack unit boots, it boots in a particular role, such as standalone, Active Controller, Standby Controller, or stack member. When the bootup role is Standby Controller or stack member, the CLI available to the unit is limited to **show** and **stack** commands. A unit in the role of Standby or stack member will not act without instructions from the Active Controller. To convert a Standby Controller or stack member into a standalone device, use the **stack unconfigure me** command (refer to [“Unconfiguring an IronStack”](#) on page 287).

The last line of the **show version** output identifies the unit role unless the unit is in standalone mode.

Example

```
My stack unit ID = 1, bootup role = active
My stack unit ID = 3, bootup role = standby
```

Active Controller and Standby Controller elections

Whenever there is a topology change in the stack (a reset, unit failure, or the addition or removal of members), elections are held to determine the status of the Active Controller and Standby Controller. The results of the election take effect after the next stack reset.

The following conditions, in the order shown, determine which units will serve as Active Controller and Standby Controller after an election:

- **Boot as Active Controller** - Indicates that a unit was previously Active Controller before the current boot sequence and will again assume the role of Active Controller when two standalone units are combined into a stack. When a third standalone unit joins the stack, a current Active Controller becomes subject to the other factors in this list. The reason for this hierarchy of factors is to achieve a predictable winner regardless of the boot up sequence for a unit. You can upgrade your current Active Controller to “boot as active controller” status by performing a **write memory**. The system interprets the write memory action as a directive to maintain the current Active Controller role regardless of resets or a new unit joining the stack.
- **Priority** - The unit with the highest priority value.

- **Greater number of members** - The unit that has control over the greater number of stack members.
- **Longer up time** - An up time that is more than 30 seconds longer than the next one in size is considered. Where up times are compared, there is no effect if the difference is less than 30 seconds.
- **Lowest boot stack ID** - The unit that has the lowest boot stack ID (1-8, 1 is the lowest).
- **MAC address** - The member with the lowest MAC address.

Active Controller and Standby Controller resets

If the Active Controller is reset or removed from the stack, the entire stack reloads and Active Controller and Standby Controller elections are initiated. If the unit functioning as the previous Active Controller is no longer part of the stack, the Standby Controller unit becomes the new Active Controller. After a reset, if no stack member qualifies as Active Controller, the existing Standby Controller waits 30 seconds and then assumes the role of Active Controller.

NOTE

The details in the preceding paragraph apply to the default setup, with hitless stacking failover not enabled.

If both Active and Standby Controllers are removed the rest of the stack will continue to function because they are operating on whatever is programmed in the hardware. The stack members will not be able to learn any new addresses. You will see the following message every few minutes.

```
Stack member is non-operational because of no Active or Standby Controller
You can recover to standalone mode by "stack unconfigure me"
```

Use **stack unconfigure me** to restore the units into standalone devices with a pre-stacking configuration.

Selecting a standby stack unit

You can choose a Standby Controller by configuring a stack unit priority to be the second highest, or the same as the Active Controller. If the priorities are configured the same for both, when the original Active Controller fails, the Standby Controller takes over. If the original Active Controller becomes active again, it will not win back its active role, which helps to minimize interruption of the stack. However, if the original Active Controller has the higher priority, it will win back its role and reset all of the stack units.

Standby Controller election criteria

The Standby Controller election is based on the following criteria.

- The highest priority
- Bootup as Active Controller
- Bootup as Standby Controller
- The lowest boot ID
- The lowest MAC address

Since Standby election candidates must have startup configurations that have been synchronized with the Active Controller, if the Active Controller does not have a startup-config.txt file, there will not be a Standby Controller. Once a write memory is performed on the Active Controller, the startup-config.txt file is written and synchronized to all stack members, and a Standby Controller can be elected.

Hitless stacking

Hitless stacking is supported on FCX and ICX units in an IronStack. It is a high-availability feature set that ensures sub-second or no loss of data traffic during the following events:

- Active Controller failure or role change
- Software failure
- Addition or removal of units in a stack
- Removal or disconnection of the stacking cable between the Active and Standby Controllers

During such events, the Standby Controller takes over the active role and the system continues to forward traffic seamlessly, as if no failure or topology change has occurred. In software releases that do not support hitless stacking, events such as these could cause most of the units in a stack to reset, resulting in an impact to data traffic.

The following hitless stacking features are supported:

Hitless stacking switchover – A manually-controlled (CLI-driven) or automatic switchover of the Active and Standby Controllers without reloading the stack and without any packet loss to the services and protocols that are supported by hitless stacking. A switchover is activated by the CLI command **stack switch-over**. A switchover might also be activated by the CLI command **priority**, depending on the configured priority value.

Hitless stacking failover – An automatic, forced switchover of the Active and Standby Controllers because of a failure or abnormal termination of the Active Controller. In the event of a failover, the Active Controller abruptly leaves the stack and the Standby Controller immediately assumes the active role. Like a switchover, a failover occurs without reloading the stack. Unlike a switchover, a failover generally happens without warning and will likely have sub-second packet loss (packets traversing the stacking link may be lost) for a brief period of time.

The services and protocols supported by hitless stacking are listed in [Table 56](#) on page 330.

Hitless stacking is disabled by default. To enable it, refer to [“Enabling hitless stacking”](#) on page 341.

Supported hitless stacking events

The following events are supported by hitless stacking:

- Failover
- Switchover
- Priority change
- Role change

Non-supported hitless stacking events

The following events are not supported by hitless stacking. These events require a software reload, resulting in an impact to data traffic.

- Unit ID change – When a stack is formed or when a unit is renumbered using secure-setup.
- Stack merge – When the old Active Controller comes back up, it reboots. If it has fewer number of members than the Active Controller, it loses the election, regardless of its priority. If it has a higher priority, it becomes the Standby Controller after the reboot and is synchronized with the Active Controller. Next, a switchover occurs and it becomes the new Active Controller.

Supported hitless stacking protocols and services

Table 56 lists the services and protocols that are supported by hitless stacking. Table 56 also highlights the impact of a hitless switchover or failover to the system's major functions.

NOTE

Services and protocols that are not listed in Table 56 will encounter disruptions, but will resume normal operation once the new Active Controller is back up and running.

TABLE 56 Hitless-supported services and protocols

Traffic type	Supported protocols and services	Impact
Layer 2 switched traffic, including unicast and multicast	<ul style="list-style-type: none"> • 802.1p and 802.1Q • 802.3ad – LACP • 802.3af – PoE • 802.3at – PoE+ 	Layer 2 switched traffic is not impacted during a hitless stacking event. All existing switched traffic flows continue uninterrupted.
+		
System-level	<ul style="list-style-type: none"> • DSCP honoring and Diffserv 	New switched flows are not learned by the switch during the switchover process and are flooded to the VLAN members in hardware. After the new Active Controller becomes operational, new switched flows are learned and forwarded accordingly. The Layer 2 control protocol states are not interrupted during the switchover process.
+	<ul style="list-style-type: none"> • Dual-mode VLAN 	
Layer 4	<ul style="list-style-type: none"> • IGMP v1, v2, and v3 snooping • IPv4 ACLs • Layer 2 switching (VLAN and 802.1Q-in-Q) • MAC-based VLANs • MLD v1 and v2 snooping • MRP • Multiple spanning tree (MSTP) • Physical port/link state • PIM SM snooping • Port mirroring and monitoring • Port trunking • Rapid spanning tree (RSTP) • Spanning tree (STP) • ToS-based QoS • Policy Based Routing • Traffic policies • UDLD • VSRP 	

TABLE 56 Hitless-supported services and protocols

Traffic type	Supported protocols and services	Impact
Layer 3 IPv4 routed traffic (unicast)	<ul style="list-style-type: none"> • IPv4 unicast forwarding • Static routes • OSPF v2 • OSPF v2 with ECMP • VRRP • VRRP-E 	<p>Layer 3 routed traffic for supported protocols is not impacted during a hitless stacking event.</p> <p>All existing Layer 3 IPv4 multicast flows and receivers may be interrupted. Traffic will converge to normalcy after the new active module becomes operational.</p> <p>Other Layer 3 protocols that are not supported will be interrupted during the switchover or failover.</p> <p>If BGP4 graceful restart or OSPF graceful restart is enabled, it will be gracefully restarted and traffic will converge to normalcy after the new active module becomes operational. For details about OSPF graceful restart, refer to “OSPF graceful restart” on page 1234. For details about BGP4 graceful restart, refer to “BGP4 graceful restart” on page 1347.</p>
Management traffic	N/A	<p>All existing management sessions (SNMP, TELNET, HTTP, HTTPS, FTP, TFTP, SSH etc.), are interrupted during the switchover process. All such sessions are terminated and can be re-established after the new Active Controller takes over.</p>

TABLE 56 Hitless-supported services and protocols

Traffic type	Supported protocols and services	Impact
Security	<ul style="list-style-type: none"> • 802.1X, including use with VLANs • EAP with RADIUS • IPv4 ACLs • DHCP snooping • Dynamic ARP inspection • IP source guard • Multi-device port authentication (MDPA), including use with dynamic VLANs • MAC port security 	<p>Supported security protocols and services are not impacted during a switchover or failover, with the following exceptions:</p> <ul style="list-style-type: none"> • 802.1X is impacted if re-authentication does not occur in a specific time window. • MDPA is impacted if re-authentication does not occur in a variable-length time window. • In some cases, a few IP source guard packets may be permitted or dropped. • If 802.1X and MDPA are enabled together on the same port, both will be impacted during a switchover or failover. Hitless support for these features applies to ports with 802.1X <i>only</i> or multi-device port authentication <i>only</i>. • For MAC port security, secure MACs are synchronized between the Active and Standby Controllers, so they are hitless. However, denied MACs are lost during a switchover or failover but may be relearned if traffic is present. <p>Configured ACLs will operate in a hitless manner, meaning the system will continue to permit and deny traffic during the switchover or failover process. However, dynamic ACLs are not supported for hitless switchover and failover.</p> <p>After a switchover or failover, the new Active Controller will re-authenticate 802.1X or MDPA sessions that were being forwarded in hardware. The hardware continues to forward them (even with dynamic VLAN) while re-authentication occurs. After trying to re-authenticate for a certain amount of time (depending on the number of sessions to re-authorize), sessions that did not re-authenticate are removed.</p>
Other services to Management	<ul style="list-style-type: none"> • AAA • DHCP • sFlow • SNMP v1, v2, and v3 • SNMP traps • SNTP • Traceroute 	<p>Supported protocols and services are not impacted during a switchover or failover.</p> <p>DNS lookups will continue after a switchover or failover. This information is not synchronized.</p> <p>Ping traffic will be minimally impacted.</p> <p>NOTE: If the FCX stack is rebooted, sFlow is disabled on standby and member units until the configuration is synchronized between the Active and Standby Controllers.</p>

Hitless stacking configuration notes and feature limitations

- For hitless stacking on the FCX, Brocade recommends that you configure the IronStack MAC address using the **stack mac** command. Without this configuration, the MAC address of the stack will change to the new base MAC address of the Active Controller. This could cause a spanning tree root change. Even without a spanning tree change, a client (for example, a personal computer) pinging the stack might encounter a long delay depending on the client MAC aging time. The client won't work until it ages out the old MAC address and sends ARP requests to relearn the new stack MAC address. Refer to [“Manually allocating the IronStack MAC address”](#) on page 275.
- Layer 3 multicast traffic is not supported by hitless stacking.
- After a switchover or failover, the Syslog may contain invalid (non-existent) port numbers in messages such as “Interface<portnum> state up”. This is because some messages from the old Active Controller will remain in the Syslog after a switchover or failover.
- Failover for devices connected to the management port is not supported. For example, if during a failover, an end station is connected to the stack through the management port of the Active Controller, the connection will be shut down. After the failover, the management port on the new Active Controller will work.
- The following describes hitless stacking limitations with software-based licensing for BGP:
 - To enable BGP on a stack unit, you should have an appropriate BGP license installed on all the stack units.
 - If the Active Controller has a BGP license but any other unit in the stack does not have, you cannot enable BGP on the stack unit.
 - If the Active Controller is not running BGP, a stack unit is operational regardless of whether the Active Controller or stack units have a BGP license or not.
 - If the Active Controller is running BGP, and a unit without a BGP license joins the stack, the unit is put into a non-operational state. But, If a user copies the BGP license to a non-operational unit, it must take effect immediately and becomes operational. Or, if the user disable BGP, Active Controller will again put all the non-operational units in operational mode.

What happens during a hitless stacking switchover or failover

This section describes the internal events that enable a controlled or forced switchover to take place in a hitless manner, as well as the events that occur during the switchover.

Real-time synchronization among all FCX units in a stack

Hitless stacking requires that the Active Controller, Standby Controller, and stack members are fully synchronized at any given point in time. This is accomplished by *baseline* and *dynamic synchronization* of all units in a stack.

When an FCX stack is first booted and becomes operational, baseline synchronization occurs across all of the units in the stack. The Active Controller copies the current state of its CPU to all units of the stack, including the Standby Controller. The information received from the Active Controller is programmed locally in hardware on all units. The information includes:

- Start-up and run-time configuration (CLI) – These files are copied to the Standby Controller only.
- Layer 2 protocols – Layer 2 protocols such as STP, RSTP, MRP, and VSRP run concurrently on both the Active and Standby Controller.
- Hardware Abstraction Layer (HAL) – This includes the prefix-based routing table, next hop information for outgoing interfaces, and tunnel information.
- Layer 3 IP forwarding information – This includes the routing table, IP cache table, and ARP table, as well as static and connected routes.
- Layer 3 routing protocols are not copied to any of the units in the stack, but remain in init state on the Standby Controller until a switchover occurs. Peer adjacency will be restored after a switchover. If BGP4 or OSPF graceful restart are enabled during a switchover, the Standby Controller (new Active Controller) will initiate a graceful restart and a new set of routes will be relearned. The new set of routes will be the same as the old routes, except in the case of a network change.

When control protocols are synchronized and protocol synchronization timers have expired, the Standby Controller will be in *hot-standby* mode, meaning the Standby Controller will be ready to take over as the Active Controller. In the event of a switchover, the Standby Controller will pick up where the active module left off, without interrupting data traffic.

After baseline synchronization, any new events that occur on the Active Controller will be dynamically synchronized on the Standby Controller. Examples of such events include:

- CLI/HTTP/SNMP configurations
- CPU receive packets
- Link events
- Interrupts
- Layer 2 and Layer 3 forwarding table updates
- Dynamic user authentication updates such as 802.1X or multi-device port authentication

After a switchover, the new Active Controller receives updates from the stack members and sends verification information to the stack members to ensure that they are synchronized.

NOTE

The events described previously occur internally and do not create or affect the external network topology.

NOTE

If there is no Active Controller after a reload, the bootup standby assumes the active role in approximately 60 seconds without a reload. A bootup standby is the device that was the Standby Controller before the reload. It may not be the current Standby Controller.

How a Hitless switchover or failover impacts system functions

For a description of the feature's impact on major system functions, refer to [Table 56](#) on page 330.

Standby Controller role in hitless stacking

In software releases that do not support hitless stacking, the Standby Controller functions as a *dummy* device, meaning it provides limited access to the CLI, such as **show**, **stack**, and a few **debug** commands. The Active Controller can access the full range of the CLI. The Standby Controller synchronizes its configuration with the Active Controller at each reset.

With the introduction of hitless stacking, the Standby Controller shadows the Active Controller. The role or behavior of the Standby Controller with hitless stacking is as follows:

- The local console on the Standby Controller still accepts only **show**, **stack**, and a few **debug** commands.
- The runtime configuration on the Standby Controller is synchronized with the Active Controller whenever there is a configuration change.
- Protocols are configured in the runtime configuration, but no protocol packets are sent out on the Standby.
- The state of every unit is known, including the state of the Active Controller. The **show** commands will display current information, such as STP or port states.
- When a failover occurs, the Standby Controller will use its current runtime configuration. The configuration could be different from the Active Controller if the last configuration transmission was lost.
- After a failover, the new Active Controller (old standby) programs all other units in hardware, based on its runtime configuration.

Standby Controller election

Candidates for Standby Controller must meet the following criteria:

- The unit is operational and the image and module configuration match that of the Active Controller.
- The runtime configuration matches that of the Active Controller.

If more than one unit in the stack meets this criteria, the Standby Controller is chosen according to the following criteria, in the order shown:

- Priority – The unit with the highest priority value.
- Current standby – The unit that is currently the Standby Controller.
- Bootup master – The unit that was the Active Controller before the stack was reloaded.
- Bootup standby – The unit that was the Standby Controller before the stack was reloaded.

Once the Standby Controller is identified, the following internal events take place.

1. The Standby Controller is assigned by the Active Controller 30 to 60 seconds after election (60 seconds if the Active Controller boots up in less than 120 seconds).
2. The Standby Controller receives and processes the runtime configuration sent by the Active Controller.
3. The Standby Controller learns the protocols within 70 seconds.

When the Standby Controller is fully synchronized, the system will be ready for a switchover or failover.

Runtime configuration mismatch

In some cases, such as a runtime configuration mismatch between the Active Controller and candidate Standby Controller, the Standby Controller cannot be assigned by the Active Controller unless the candidate Standby Controller is reloaded.

As shown in the following example, the **show stack** command output will indicate whether there is a runtime configuration mismatch.

```
Brocade#show stack
alone: standalone, D: dynamic config, S: static config
ID   Type      Role      Mac Address    Pri State   Comment
1   S FCX624S    active    00e0.5201.0000  30 local   Ready
2   d FCX648SPOE member  00e0.5202.0000  20 remote  ready, standby if reload

              active
      +----+      +----+
-2/1| 2 |2/2--2/1| 1 |2/2-
      +----+      +----+
```

Note: There is no standby. Reason: u2: diff run-time config
 Current stack management MAC is 00e0.5201.0000
 Note: no "stack mac" config. My MAC will change after failover.

Support during stack formation, stack merge, and stack split

This section illustrates hitless stacking support during stack formation, stack merge, and stack split.

Figure 22 illustrates hitless stacking support during stack formation. Operational stages 1 and 2 are also shown in this illustration.

FIGURE 22 Hitless stacking support during stack formation

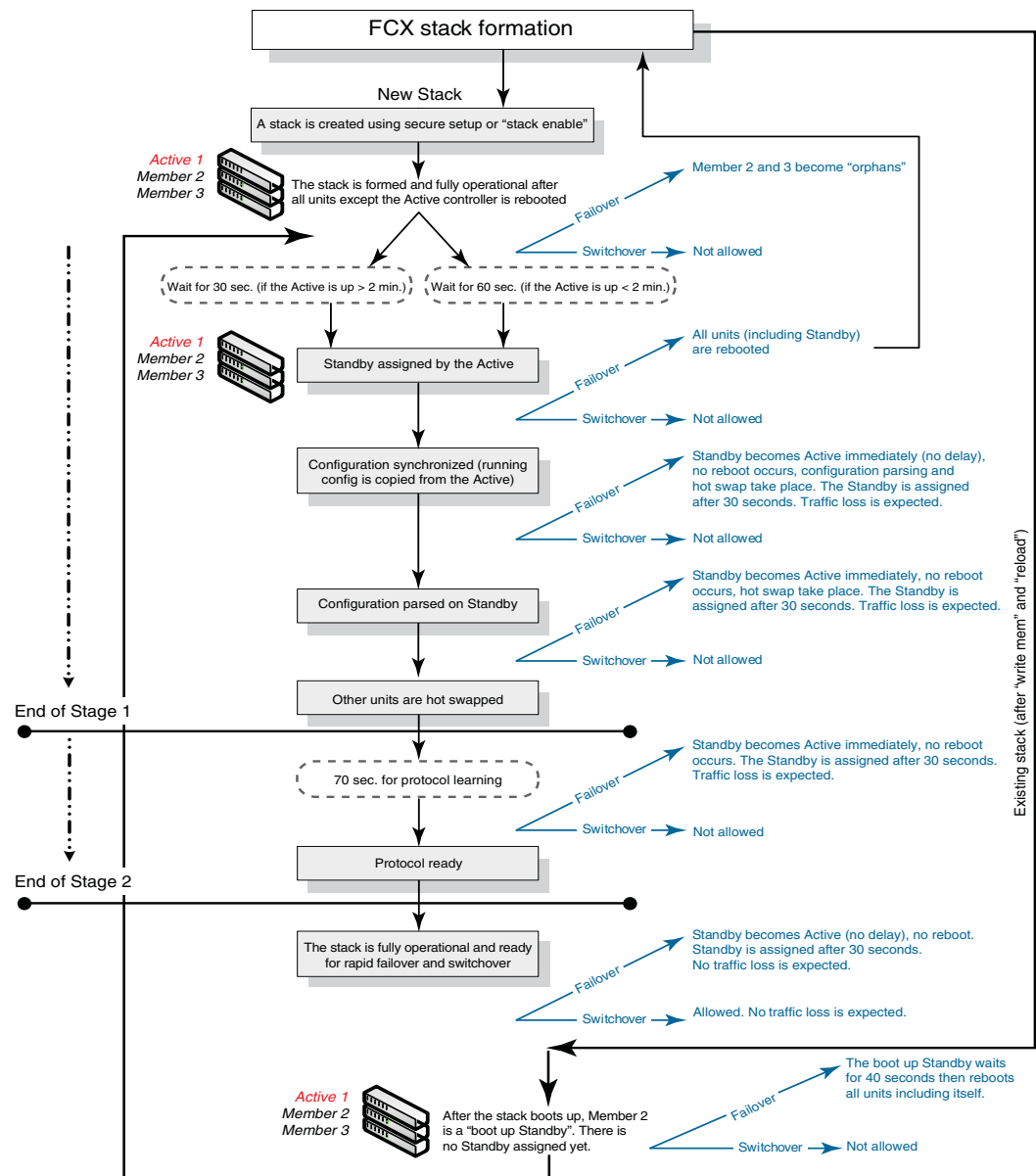


Figure 23 illustrates hitless stacking support during a stack merge.

FIGURE 23 Hitless stacking support during a stack merge

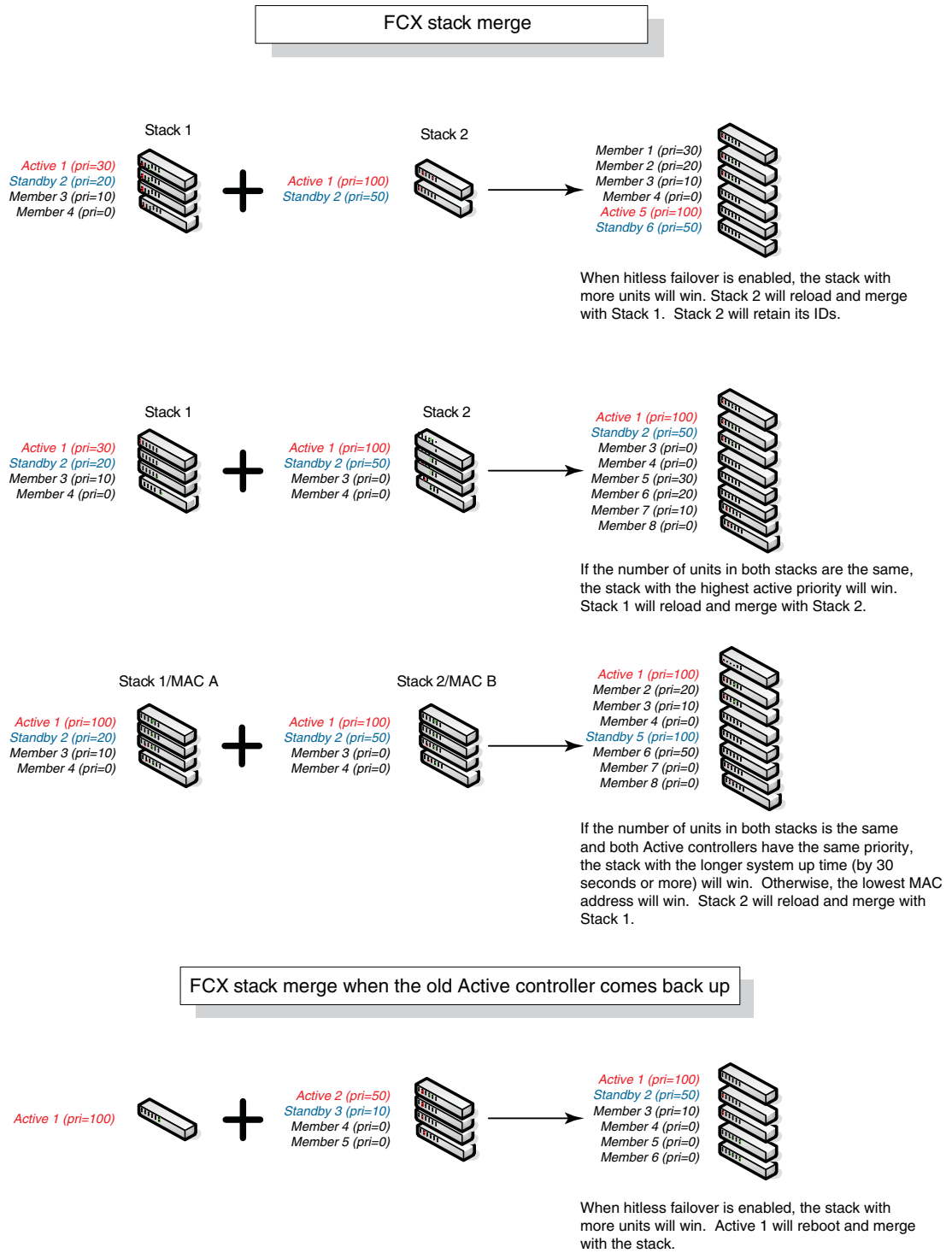
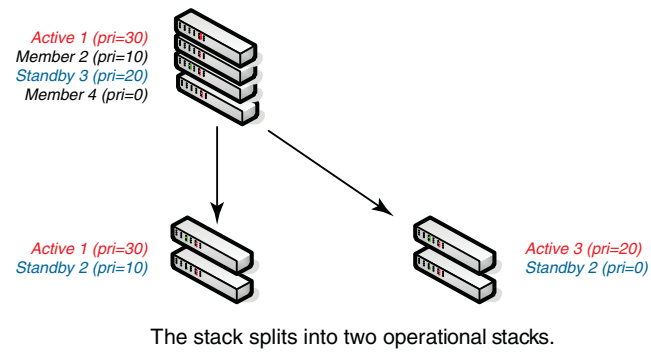
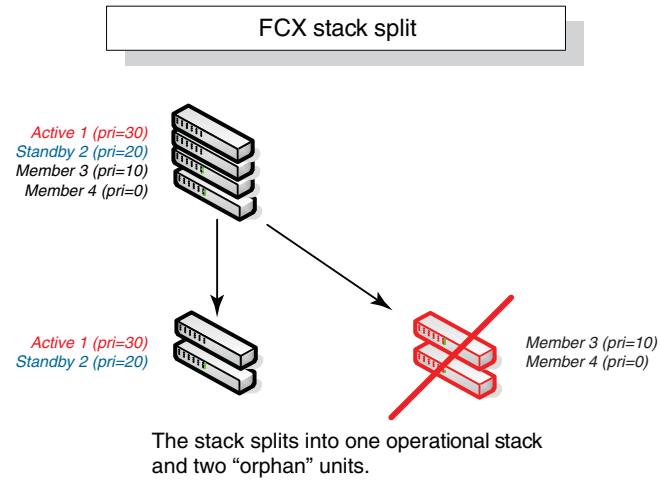


Figure 24 illustrates hitless stacking support in a stack split.

FIGURE 24 Hitless stacking support in a stack split



Hitless stacking default behavior

Hitless stacking is disabled by default. When disabled, the following limitations are in effect:

- If a failover occurs, every unit in the stack will reload
- Manual switchover is not allowed. If the CLI command **stack switch-over** is entered, the following message will appear on the console:
`Switch-over is not allowed. Reason: hitless-failover not configured.`
- Internal switchover resulting from a priority change is blocked until the entire stack is reloaded or hitless stacking is enabled. A priority change will trigger an election, but the newly elected winner will not immediately assume its role. For more information, refer to [“Displaying pending device roles”](#) on page 341.
- If there is no Active Controller after a reload, the bootup standby will assume the active role after reloading every unit in the stack, including itself.
- During a stack merge, the Active Controller with the highest priority will win the election and reload every unit of the losing stack.

NOTE

Synchronization between the Active Controller, Standby Controller, and stack members will occur whether or not hitless stacking is enabled.

When hitless stacking is enabled, the following behavior takes effect immediately:

- If a failover occurs, the stack will not reload.
- Manual switchover (CLI command **stack switch-over**) is allowed.
- If a priority change occurred while hitless stacking was disabled, and the configured priority value requires a switchover, the system will start a 60-second timer before performing a switchover. After the switchover, the highest priority standby will become the Active Controller.
- If there is no Active Controller after a reload, the bootup standby will assume the active role in approximately 90 seconds without a reload.
- During a stack merge, the Active Controller with the larger number of units will win the election and reload every unit of the losing stack. If two stacks have the same number of units, then the priority, system up time, ID, then MAC address is compared. If two stacks have the same number of units and the same priority, then the stack with the longest system up-time (by 30 seconds or more) will win the election. Otherwise, the smallest ID is compared next, followed by MAC address. If the losing Active Controller has the highest priority, it will become a standby after reloading and relearning the protocols. Finally, it will become the Active Controller after an internal switchover.

NOTE

If the Active Controllers of two merging stacks have different hitless stacking settings (i.e., hitless stacking is enabled in one stack and disabled in the other), the default behavior (hitless stacking disabled) will be used in the stack merge. After the merge, the winner will retain its hitless stacking setting and runtime configuration for the merged stack.

You can use the **show stack** command to view whether or not hitless stacking is enabled. Refer to [“Displaying hitless stacking status”](#) on page 341.

Enabling hitless stacking

Hitless stacking is disabled by default. To enable it, enable hitless failover as described in [“Enabling hitless stacking failover”](#) on page 342.

Displaying hitless stacking status

You can use the **show stack** command to view whether or not hitless stacking is enabled. The following example shows that hitless stacking is disabled.

```
Brocade#show stack
alone: standalone, D: dynamic config, S: static config
ID   Type      Role    Mac Address    Pri State  Comment
2   S FCX648S   member  0000.0000.0000    0 reserve
3   S FCX624   member  0024.3876.2640    0 remote  Ready
5   S FCX624   standby 00e0.5200.0400 100 remote  Ready
8   S FCX648   active  0024.3877.7980 128 local   Ready
```

```

      active                      standby
      +---+                      +---+
-2/1| 8 |2/2--2/2| 3 |2/1--2/1| 5 |2/2-
|   +---+                      +---+
|
|-----|

```

Standby u5 - No hitless failover. Reason: hitless-failover not configured

Syntax: show stack

Displaying pending device roles

When hitless stacking is disabled, a priority change will trigger an election, but the newly-elected winner will not assume its role until the entire stack is reloaded or hitless stacking is enabled.

You can use the **show stack** command to view pending device roles. The “Role” column displays the current role for each unit. The “Comment” column displays the role that will take effect after a reload or when hitless stacking is enabled.

Hitless stacking

```
Brocade#show stack
alone: standalone, D: dynamic config, S: static config
ID   Type      Role      Mac Address      Pri State      Comment
2   S FCX648S   member    0000.0000.0000    0  reserve
3   S FCX624    standby   0024.3876.2640    200 remote  Ready, active if reloaded
5   S FCX624    member    00e0.5200.0400    128 remote  Ready, standby if reloaded
8   S FCX648    active    0024.3877.7980    128 local   Ready, member if reloaded
```

```

      active      standby
      +---+      +---+      +---+
-2/1| 8 |2/2--2/2| 3 |2/1--2/1| 5 |2/2-
|      +---+      +---+      +---+
|-----|

```

Standby u3 - No hitless failover. Reason: hitless-failover not configured

Syntax: show stack

Hitless stacking failover

Hitless stacking failover provides automatic failover from the Active Controller to the Standby Controller without resetting any of the units in the stack and with sub-second or no packet loss to hitless stacking-supported services and protocols.

For a description of the events that occur during a hitless failover, refer to [“What happens during a hitless stacking switchover or failover”](#) on page 333.

For a description of this feature’s impact on major system functions, refer to [Table 56](#) on page 330.

For an example of hitless failover operation, refer to [“Hitless stacking failover example”](#) on page 343.

For feature limitations and configuration notes, refer to [“Hitless stacking configuration notes and feature limitations”](#) on page 333.

Enabling hitless stacking failover

To enable hitless stacking failover, enter the following command at the global CONFIG level of the CLI:

```
Brocade(config)#hitless-failover enable
```

The command takes effect immediately. Hitless switchover is allowed, and in the event of a failover, the standby controller will take over the active role without reloading the stack.

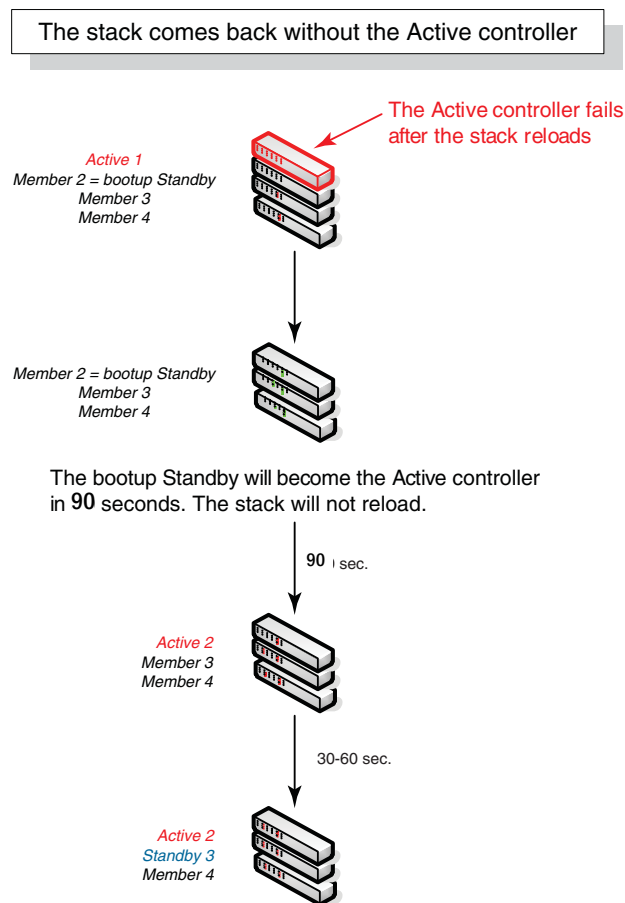
Syntax: [no] hitless-failover enable

Use the **no** form of the command to disable hitless stacking once it has been enabled.

Hitless stacking failover example

Figure 25 illustrates hitless stacking failover operation when the Active Controller fails.

FIGURE 25 Hitless stacking failover when the Active Controller fails



Hitless stacking switchover

Hitless stacking switchover is a manually-controlled (CLI-driven) or automatic switchover of the Active and Standby Controllers without reloading the stack and without any packet loss to the services and protocols that are supported by hitless stacking. A switchover is activated by the CLI command **stack switch-over**. A switchover might also be activated by the CLI command **priority**, depending on the configured priority value.

By default, hitless switchover is not allowed. The default behavior is described in [“Hitless stacking default behavior”](#) on page 340.

Hitless switchover can be used by a system administrator, for example, to perform maintenance on a controller that has been functioning as the Active Controller.

For a description of the events that occur during a hitless stacking switchover, refer to [“What happens during a hitless stacking switchover or failover”](#) on page 333.

For a description of this feature's impact on major system functions, refer to [Table 56](#) on page 330.

For examples of hitless stacking switchover operation, refer to [“Hitless stacking switchover examples”](#) on page 345.

Executing a hitless stacking switchover

The following must be in effect before a hitless switchover (switch over to the Standby Controller) is allowed:

- Hitless stacking is enabled
- The stack has a Standby Controller
- The Standby Controller has learned the protocols
- The Standby Controller has the same priority as the Active Controller
- More than 120 seconds have passed since the previous switchover or failover

You can use the **show stack** command to view whether or not these properties are in effect. For more information, refer to [“Displaying information about hitless stacking”](#) on page 350.

To perform a switchover, enter the following command:

```
Brocade# stack switch-over
Standby unit 8 will become Active Controller, and unit 1 will become standby
Are you sure? (enter 'y' or 'n'): y
Unit 1 is no longer the Active Controller
```

Syntax: **stack switch-over**

Hitless stacking switchover examples

This section illustrates hitless stacking failover and switchover operation during a CLI-driven switchover or priority change.

Figure 26 illustrates a hitless stacking switchover triggered by the **stack switch-over** command.

FIGURE 26 Manual switchover

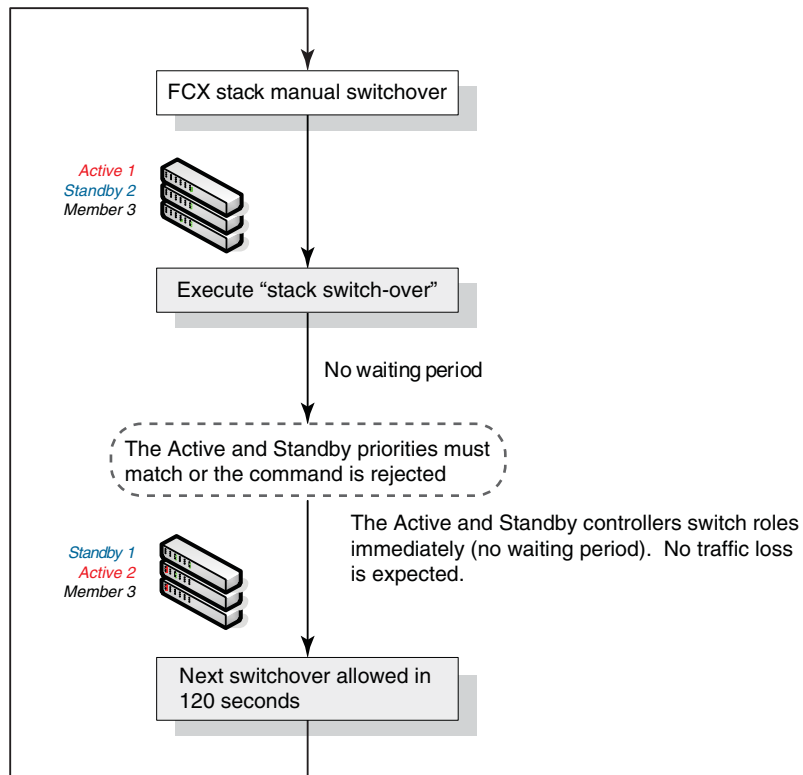


Figure 27 illustrates a hitless stacking switchover when the Active Controller goes down then comes back up. The stack in this example has user-configured priorities.

FIGURE 27 Hitless stacking switchover when the Active Controller comes back up

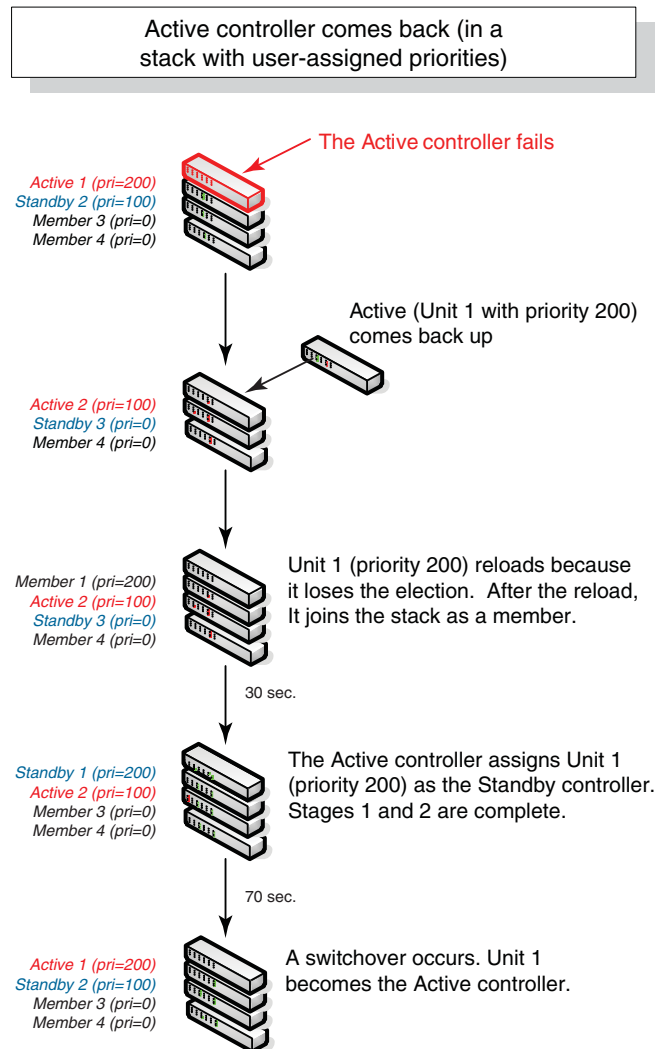


Figure 28 illustrates a hitless stacking switchover after the network administrator increases the priority value of the Standby Controller.

FIGURE 28 Scenario 1 – Hitless stacking switchover after a priority change

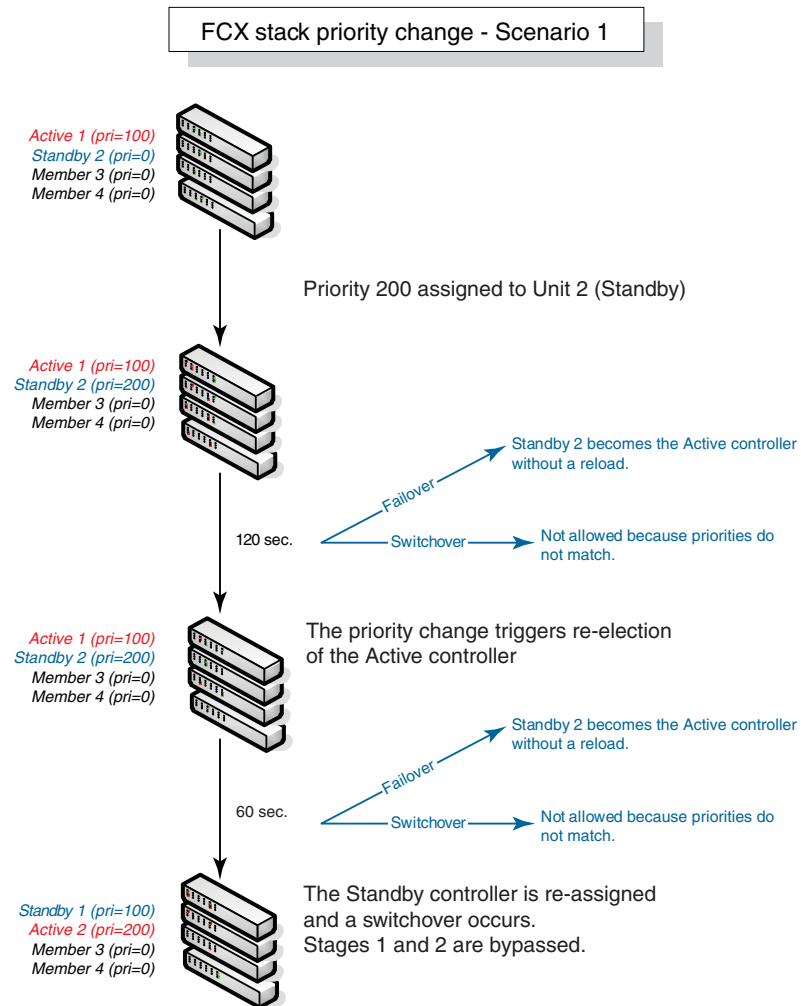


Figure 29 illustrates a hitless stacking switchover after the network administrator increases the priority value of one of the stack members.

FIGURE 29 Scenario 2 – Hitless stacking switchover after a priority change

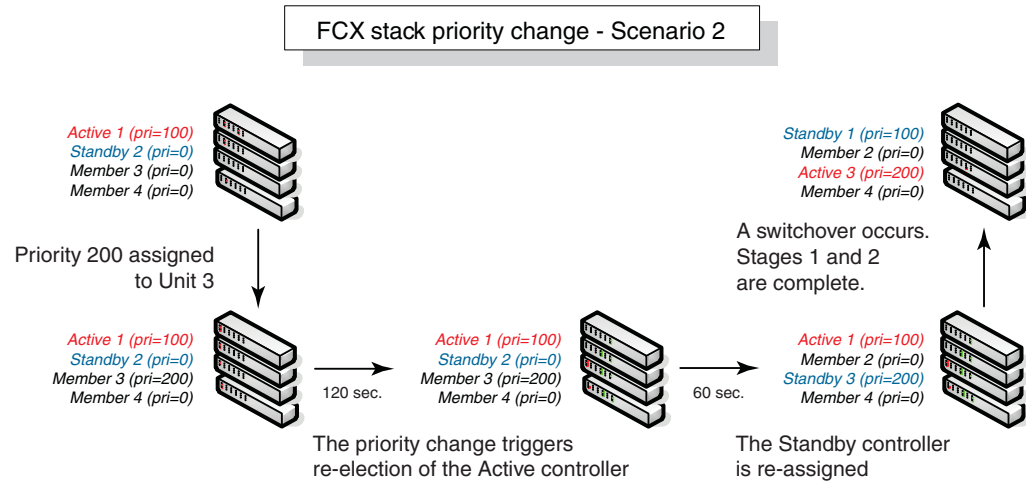
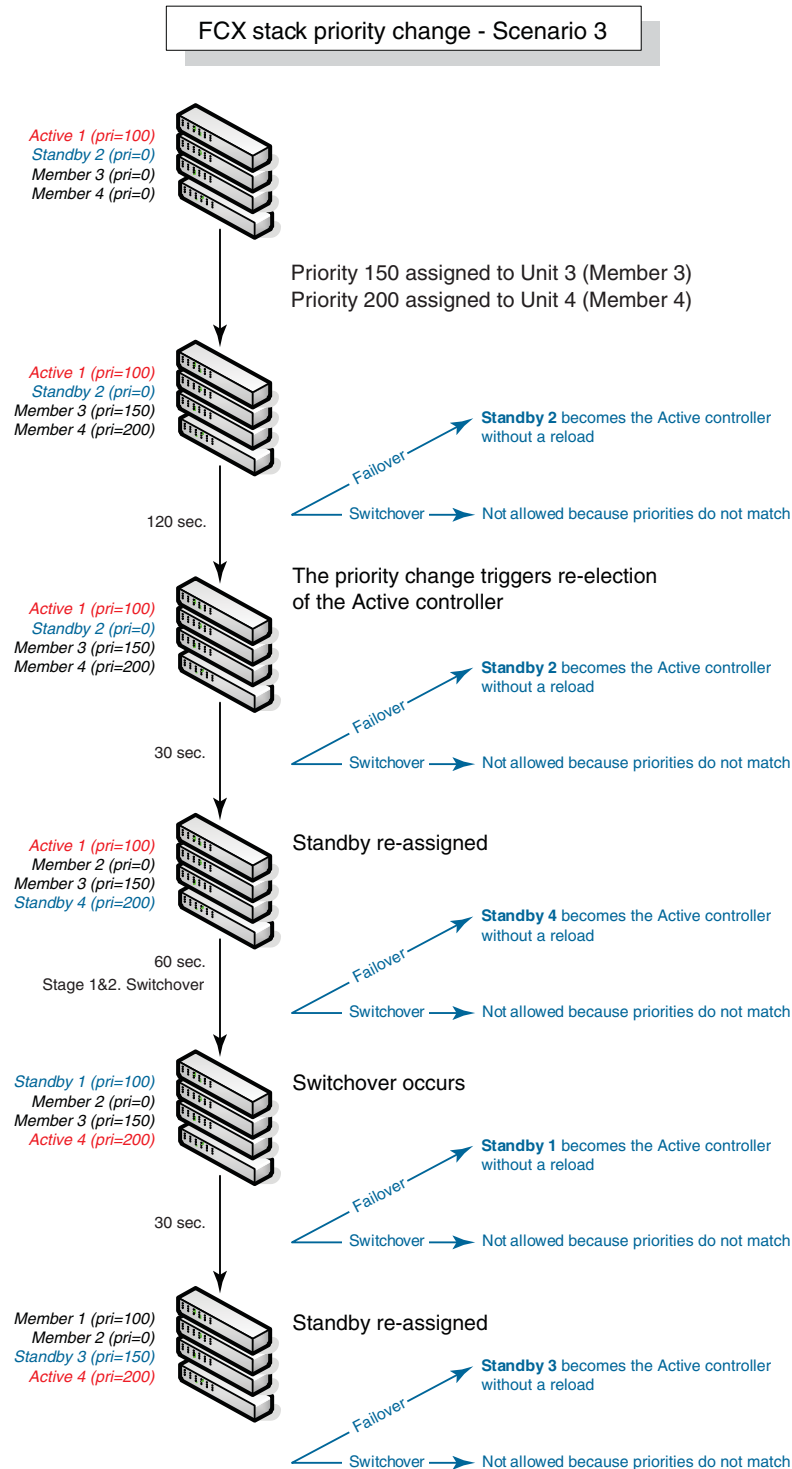


Figure 30 illustrates a hitless stacking switchover after the network administrator increases the priority value for two of the stack members.

FIGURE 30 Scenario 3 – Hitless stacking switchover after a priority change



Displaying information about hitless stacking

Use the **show stack** command to view information pertinent to a hitless stacking switchover or failover. The command output illustrates the Active and Standby Controllers, as well as the readiness of the Standby Controller to take over the role of Active Controller, if needed.

```
Brocade#show stack
alone: standalone, D: dynamic config, S: static config
ID   Type      Role      Mac Address    Pri State  Comment
1   S FCX624S   active    00e0.5200.2900 128 local  Ready
2   S FCX624S   standby   00e0.5200.0100 128 remote Ready
3   S FCX624SPOE member    00e0.5288.8888 50 remote Ready
4   S FCX624S   member    0000.0000.0000 128 reserve
```

```

      active      standby
      +---+      +---+      +---+
-1/3| 1 |1/5--1/5| 2 |1/3--1/5| 3 |1/3-
|   +---+      +---+      +---+ |
|                                   |
|-----|
```

Standby unit 2: protocols ready, can failover or manually switch over
 Current stack management MAC is 0000.5200.1100

NOTE

The text in bold highlights the information added for hitless stacking failover and switchover. For a description of the fields in this output, refer to [“Field descriptions for the show stack command”](#) on page 295.

Displaying information about stack failover

Use the **show stack failover** command to view information about rapid failover. The following command output is from a Brocade ICX 6610 device.

```
Brocade#show stack failover
Current standby is unit 5. state=ready
Standby u5 - protocols ready, can failover or manually switch over
```

Displaying information about link synchronization status

Use the **show stack link-sync status** command to view the status of the link synchronization. The following command output is from a Brocade ICX 6610 device.

```
Brocade#show stack link-sync status
STACKING_LINK_GLOBAL_CTRL messages sent: 0, received: 0
STACKING_LINK_INDIVIDUAL_CTRL messages sent: 359, received: 0
STACKING_LINK_STATUS messages sent: 22300, received: 128883
STACKING_POE_SCTRL messages sent: 0, received: 0
STACKING_POE_STATUS messages sent: 0, received: 0
```

```
global_ctrl_dest: ffffffff
individual_ctrl_dest: ee
status_dest: 30
```

Syslog messages for hitless stacking failover and switchover

Syslog messages are generated for the following events:

- Switchover
- Failover
- Standby Controller assignment

Table 57 lists the supported syslog messages.

TABLE 57 Syslog messages

Message level	Message	Explanation
Informational	Stack: Stack unit <unit_number> has been assigned as STANDBY unit of the stack system	Indicates that the unit has been assigned as the Standby Controller.
Informational	Stack: Stack is operational due to SWITCH-OVER	Indicates that a switchover has occurred.
Informational	Stack: Stack is operational due to FAIL-OVER	Indicates that a failover has occurred.

To view the System log or the traps logged on an SNMP trap receiver, enter the **show log** command at any level of the CLI. The following example output shows what the log might look like after a switchover or assignment of the Standby Controller.

```
Brocade# show log
Syslog logging: enabled (0 messages dropped, 1 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 8 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
               I=informational N=notification W=warningDynamic Log Buffer (50 lines):
0d00h05m34s:I:System: Interface ethernet mgmt1, state up
0d00h05m33s:I:Stack: Stack unit 8 has been assigned as STANDBY unit of the stack system
0d00h05m33s:I:Stack: Stack is operational due to SWITCH-OVER
0d00h05m32s:I:Stack: Stack unit 1 has been elected as ACTIVE unit of the stack system
0d00h05m29s:W:System:Stack unit 2 Fan speed changed automatically to 2
0d00h05m25s:W:System:Stack unit 5 Fan speed changed automatically to 2
0d00h05m00s:I:System: Interface ethernet mgmt1, state down
0d00h05m00s:I:Security: Telnet server enabled by from session
```

The following example output shows what the log might look like after a failover of the Active Controller.

NOTE

In FCX devices after failover, the output of the **show log** command displays the status of the previous management port interface.

Hitless stacking

```
Brocade# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 12 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
0d00h04m41s:I:Stack: Stack unit 3 has been assigned as STANDBY unit of the stack system
0d00h04m12s:I:System: Interface ethernet mgmt1, state up
0d00h04m10s:I:System: Interface ethernet mgmt1, state down
0d00h04m10s:I:System: Interface ethernet mgmt1, state up
0d00h04m09s:I:STP: VLAN 1 Bridge is RootBridge: 800000e052010000 (MgmtPriChg)
0d00h04m09s:I:System: Management MAC address changed to 00e0.5201.0000
0d00h04m09s:I:Stack: Stack is operational due to FAIL-OVER
0d00h04m08s:I:Stack: Stack unit 1 has been elected as ACTIVE unit of the stack system
0d00h04m08s:I:STP: VLAN 1 Port 8/1/1 STP State -> DISABLED (PortDown)
0d00h04m08s:I:STP: VLAN 1 Port 8/1/1 STP State -> FORWARDING (PortDown)
0d00h04m08s:I:System: Interface ethernet 1/2/2, state down
0d00h04m06s:I:System: Interface ethernet 8/2/2, state down
```

IPv6 Configuration on FastIron X Series, FCX, and ICX Series Switches

[Table 58](#) lists the individual Brocade FastIron switches and the IPv6 features they support. These features are supported in the Layer 2, base Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 58 Supported IPv6 features on FastIron X Series, FCX, and ICX devices

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Global IPv6 address	Yes	This chapter describes IPv6 features on the FCX, FESX, ICX 6610, FSX 800 and FSX 1600. For information about IPv6 management features on other FastIron WS devices, refer to Chapter 9, “FWS Series Switch IPv6 management” .	Yes	Yes	Yes
IPv6 access list ¹	Yes		Yes	Yes	Yes
IPv6 access-list (management ACLs)	Yes		Yes	Yes	Yes
Site-local IPv6 address	Yes		Yes	Yes	Yes
Link-local IPv6 address	Yes		Yes	Yes	Yes
IPv4 and IPv6 host stacks	Yes		Yes	Yes	Yes
IPv6 copy ¹	Yes		Yes	Yes	Yes
IPv6 ncopy ¹	Yes		Yes	Yes	Yes
IPv6 debug	Yes		Yes	Yes	Yes
IPv6 ping	Yes		Yes	Yes	Yes
IPv6 traceroute	Yes		Yes	Yes	Yes
DNS server name resolution	Yes		Yes	Yes	Yes
HTTP/HTTPS	Yes		Yes	Yes	Yes
Logging (Syslog)	Yes		Yes	Yes	Yes
RADIUS ¹	Yes		Yes	Yes	Yes
SCP	Yes		Yes	Yes	Yes
SSH	Yes		Yes	Yes	Yes
SNMP	Yes		Yes	Yes	Yes
SNMP traps	Yes		Yes	Yes	Yes

TABLE 58 Supported IPv6 features on FastIron X Series, FCX, and ICX devices (Continued)

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
SNTP	Yes		Yes	Yes	Yes
Telnet	Yes		Yes	Yes	Yes
TFTP ¹	Yes		Yes	Yes	Yes
Router advertisement and solicitation	Yes		Yes	Yes	ICX 6450 only
IPv6 static routes	Yes		Yes	Yes	ICX 6450 only
IPv6 over IPv4 tunnels	Yes		Yes	No	No
ECMP load sharing	Yes		Yes	Yes	ICX 6450 only
IPv6 ICMP	Yes		Yes	Yes	Yes
IPv6 routing protocols ¹	Yes		Yes	Yes	No
ICMP redirect messages	Yes		Yes	Yes	ICX 6450 only
IPv6 neighbor discovery	Yes		Yes	Yes	Yes
IPv6 Layer 3 forwarding	Yes		Yes	Yes	ICX 6450 only
IPv6 redistribution	Yes		Yes	Yes	No
IPv6 MTU	Yes		Yes	Yes	ICX 6450 only
Static neighbor entries	Yes		Yes	Yes	ICX 6450 only
Hop limit for IPv6 packets	Yes		Yes	Yes	ICX 6450 only
Clear IPv6 global information	Yes		Yes	Yes	ICX 6450 only
IPv6 source routing security enhancements	No		Yes	Yes	No

¹The following IPv6 features, listed in Table 58, are documented in other chapters of this guide:

- IPv6 access list – [“IPv6 ACLs”](#) on page 1755
- IPv6 copy – [“Using the IPv6 copy command”](#) on page 87
- IPv6 ncopy – [“IPv6 ncopy command”](#) on page 89
- RADIUS – [“Setting RADIUS over IPv6”](#) on page 167
- TFTP – [“Loading and saving configuration files with IPv6”](#) on page 87
- IPV6 routing protocols – Various chapters

Full Layer 3 IPv6 feature support

The following IPv6 Layer 3 features are supported only with the IPv6 Layer 3 PROM, IPv6-series hardware, and the full Layer 3 image:

- IPv6 unicast routing (multicast routing is not supported)
- OSPF V3
- RIPng
- IPv6 ICMP redirect messages
- IPv6 route redistribution
- IPv6 static routes
- IPv6 over IPv4 tunnels in hardware
- IPv6 Layer 3 forwarding

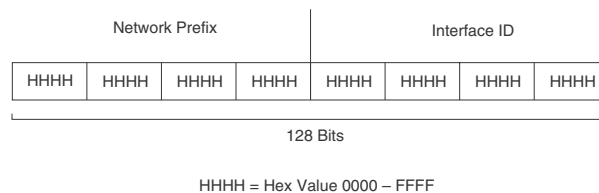
IPv6 addressing overview

IPv6 was designed to replace IPv4, the Internet protocol that is most commonly used currently throughout the world. IPv6 increases the number of network address bits from 32 (IPv4) to 128 bits, which provides more than enough unique IP addresses to support all of the network devices on the planet into the future. IPv6 is expected to quickly become the network standard.

An IPv6 address is composed of 8 fields of 16-bit hexadecimal values separated by colons (:).

[Figure 31](#) shows the IPv6 address format.

FIGURE 31 IPv6 address format



As shown in [Figure 31](#), HHHH is a 16-bit hexadecimal value, while H is a 4-bit hexadecimal value. The following is an example of an IPv6 address.

2001:0000:0000:0200:002D:D0FF:FE48:4672

Note that this IPv6 address includes hexadecimal fields of zeros. To make the address less cumbersome, you can do the following:

- Omit the leading zeros; for example, 2001:0:0:200:2D:D0FF:FE48:4672.
- Compress the successive groups of zeros at the beginning, middle, or end of an IPv6 address to two colons (::) once per address; for example, 2001::200:2D:D0FF:FE48:4672.

When specifying an IPv6 address in a command syntax, keep the following in mind:

- You can use the two colons (::) only once in the address to represent the longest successive hexadecimal fields of zeros
- The hexadecimal letters in IPv6 addresses are not case-sensitive

As shown in [Figure 31](#), the IPv6 network prefix is composed of the left-most bits of the address. As with an IPv4 address, you can specify the IPv6 prefix using the `<prefix>/<prefix-length>` format, where the following applies.

The `<prefix>` parameter is specified as 16-bit hexadecimal values separated by a colon.

The `<prefix-length>` parameter is specified as a decimal value that indicates the left-most bits of the IPv6 address.

The following is an example of an IPv6 prefix.

2001:FF08:49EA:D088::/64

IPv6 address types

As with IPv4 addresses, you can assign multiple IPv6 addresses to a switch interface. [Table 59](#) presents the three major types of IPv6 addresses that you can assign to a switch interface.

A major difference between IPv4 and IPv6 addresses is that IPv6 addresses support **scope**, which describes the topology in which the address may be used as a unique identifier for an interface or set of interfaces.

Unicast and multicast addresses support scoping as follows:

- Unicast addresses support two types of scope: global scope and local scope. In turn, local scope supports site-local addresses and link-local addresses. [Table 59](#) describes global, site-local, and link-local addresses and the topologies in which they are used.
- Multicast addresses support a scope field, which [Table 59](#) describes.

TABLE 59 IPv6 address types

Address type	Description	Address structure
Unicast	An address for a single interface. A packet sent to a unicast address is delivered to the interface identified by the address.	<p>Depends on the type of the unicast address:</p> <ul style="list-style-type: none"> • Aggregatable global address—An address equivalent to a global or public IPv4 address. The address structure is as follows: a fixed prefix of 2000::/3 (001), a 45-bit global routing prefix, a 16-bit subnet ID, and a 64-bit interface ID. • Site-local address—An address used within a site or intranet. (This address is similar to a private IPv4 address.) A site consists of multiple network links. The address structure is as follows: a fixed prefix of FEC0::/10 (1111 1110 11), a 16-bit subnet ID, and a 64-bit interface ID. • Link-local address—An address used between directly connected nodes on a single network link. The address structure is as follows: a fixed prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. • IPv4-compatible address—An address used in IPv6 transition mechanisms that tunnel IPv6 packets dynamically over IPv4 infrastructures. The address embeds an IPv4 address in the low-order 32 bits and the high-order 96 bits are zeros. The address structure is as follows: 0:0:0:0:0:A.B.C.D. • Loopback address—An address (0:0:0:0:0:0:1 or ::1) that a switch can use to send an IPv6 packet to itself. You cannot assign a loopback address to a physical interface. • Unspecified address—An address (0:0:0:0:0:0:0 or ::) that a node can use until you configure an IPv6 address for it.
Multicast	An address for a set of interfaces belonging to different nodes. Sending a packet to a multicast address results in the delivery of the packet to all interfaces in the set.	A multicast address has a fixed prefix of FF00::/8 (1111 1111). The next 4 bits define the address as a permanent or temporary address. The next 4 bits define the scope of the address (node, link, site, organization, global).
Anycast	An address for a set of interfaces belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface identified by the address.	<p>An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign a unicast address to multiple interfaces, it is an anycast address. An interface assigned an anycast address must be configured to recognize the address as an anycast address.</p> <p>An anycast address can be assigned to a switch only.</p> <p>An anycast address must not be used as the source address of an IPv6 packet.</p>

A switch automatically configures a link-local unicast address for an interface by using the prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link. If desired, you can override this automatically configured address by explicitly configuring an address.

NOTE

Brocade FastIron devices support RFC 2526, which requires that within each subnet, the highest 128 interface identifier values reserved for assignment as subnet anycast addresses. Thus, if you assign individual IPv6 addresses within a subnet, the second highest IPv6 address in the subnet does not work.

IPv6 stateless auto-configuration

Brocade routers use the IPv6 stateless autoconfiguration feature to enable a host on a local link to automatically configure its interfaces with new and globally unique IPv6 addresses associated with its location. The automatic configuration of a host interface is performed without the use of a server, such as a Dynamic Host Configuration Protocol (DHCP) server, or manual configuration.

The automatic configuration of a host interface works in the following way: a switch on a local link periodically sends switch advertisement messages containing network-type information, such as the 64-bit prefix of the local link and the default route, to all nodes on the link. When a host on the link receives the message, it takes the local link prefix from the message and appends a 64-bit interface ID, thereby automatically configuring its interface. (The 64-bit interface ID is derived from the MAC address of the host's NIC.) The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link.

The duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless auto configuration feature. Duplicate address detection uses neighbor solicitation messages to verify that a unicast IPv6 address is unique.

NOTE

For the stateless auto configuration feature to work properly, the advertised prefix length in switch advertisement messages must always be 64 bits.

The IPv6 stateless autoconfiguration feature can also automatically reconfigure a host's interfaces if you change the ISP for the host's network. (The host's interfaces must be renumbered with the IPv6 prefix of the new ISP.)

The renumbering occurs in the following way: a switch on a local link periodically sends advertisements updated with the prefix of the new ISP to all nodes on the link. (The advertisements still contain the prefix of the old ISP.) A host can use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. When you are ready for the host to use the new addresses only, you can configure the lifetime parameters appropriately using the **ipv6 nd prefix-advertisement** command. During this transition, the old prefix is removed from the switch advertisements. At this point, only addresses that contain the new prefix are used on the link.

IPv6 CLI command support

Table 60 lists the IPv6 CLI commands supported.

TABLE 60 IPv6 CLI command support

IPv6 command	Description	Switch code	Router code
clear ipv6 cache	Deletes all entries in the dynamic host cache.		X
clear ipv6 mld-snooping	Deletes MLD-snooping-related counters or cache entries.	X	X
clear ipv6 neighbor	Deletes all dynamic entries in the IPv6 neighbor table.	X	X
clear ipv6 ospf	Clears OSPF-related entries.		X
clear ipv6 rip	Clears RIP-related entries.		X
clear ipv6 route	Deletes all dynamic entries in the IPv6 route table.		X
clear ipv6 traffic	Resets all IPv6 packet counters.	X	X

TABLE 60 IPv6 CLI command support (Continued)

IPv6 command	Description	Switch code	Router code
clear ipv6 tunnel	Clears statistics for IPv6 tunnels		X
copy tftp	Downloads a copy of a Brocade software image from a TFTP server into the system flash using IPv6.	X	X
debug ipv6	Displays IPv6 debug information.	X	X
ipv6 access-class	Configures access control for IPv6 management traffic.	X	X
ipv6 access-list	Configures an IPv6 access control list for IPv6 access control.	X	X
ipv6 address	Configures an IPv6 address on an interface (router) or globally (switch)	X	X
ipv6 debug	Enables IPv6 debugging.	X	X
ipv6 dns domain-name	Configures an IPv6 domain name.	X	X
ipv6 dns server-address	Configures an IPv6 DNS server address.	X	X
ipv6 enable	Enables IPv6 on an interface.	X	X
ipv6 hop-limit	Sets the IPv6 hop limit.		X
ipv6 icmp	Configures IPv6 ICMP parameters		X
ipv6 load-sharing	Enables IPv6 load sharing		X
ipv6 mld-snooping	Configures MLD snooping	X	X
ipv6 mtu	Configures the maximum length of an IPv6 packet that can be transmitted on a particular interface.		X
ipv6 nd	Configures neighbor discovery.		X
ipv6 neighbor	Maps a static IPv6 address to a MAC address in the IPv6 neighbor table.		X
ipv6 ospf	Configures OSPF V3 parameters on an interface.		X
ipv6 prefix-list	Builds an IPv6 prefix list.		X
ipv6 redirects	Enables the sending of ICMP redirect messages on an interface.		X
ipv6 rip	Configures RIPng parameters on an interface		X
ipv6 route	Configures an IPv6 static route.		X
ipv6 router	Enables an IPv6 routing protocol.		X
ipv6 traffic-filter	Applies an IPv6 ACL to an interface.	X	X
ipv6 unicast-routing	Enables IPv6 unicast routing.		X
log host ipv6	Configures the IPv6 Syslog server.	X	X
ping ipv6	Performs an ICMP for IPv6 echo test.	X	X
show ipv6	Displays some global IPv6 parameters, such as IPv6 DNS server address.	X	X
show ipv6 access-list	Displays configured IPv6 access control lists.	X	X
show ipv6 cache	Displays the IPv6 host cache.		X

TABLE 60 IPv6 CLI command support (Continued)

IPv6 command	Description	Switch code	Router code
show ipv6 interface	Displays IPv6 information for an interface.		X
show ipv6 mld-snooping	Displays information about MLD snooping.	X	X
show ipv6 neighbor	Displays the IPv6 neighbor table.	X	X
show ipv6 ospf	Displays information about OSPF V3.		X
show ipv6 prefix-lists	Displays the configured IPv6 prefix lists.		X
show ipv6 rip	Displays information about RIPng.		X
show ipv6 route	Displays IPv6 routes.		X
show ipv6 router	Displays IPv6 local routers.		X
show ipv6 tcp	Displays information about IPv6 TCP sessions.	X	X
show ipv6 traffic	Displays IPv6 packet counters.	X	X
show ipv6 tunnel	Displays information about IPv6 tunnels.	X	X
snmp-client ipv6	Restricts SNMP access to a certain IPv6 node.	X	X
snmp-server host ipv6	Specifies the recipient of SNMP notifications.	X	X
sntp server ipv6	Enables the Brocade device to send SNTP packets over IPv6.	X	X
telnet	Enables a Telnet connection from the Brocade device to a remote IPv6 host using the console.	X	X
traceroute ipv6	Traces a path from the Brocade device to an IPv6 host.	X	X
web access-group ipv6	Restricts Web management access to certain IPv6 hosts as determined by IPv6 ACLs.	X	X
web client ipv6	Restricts Web management access to certain IPv6 hosts.	X	X

IPv6 host address on a Layer 2 switch

In a Layer 3 (router) configuration, each port can be configured separately with an IPv6 address. This is accomplished using the interface configuration process that is described in [“IPv6 configuration on each router interface”](#) on page 362.

In a Layer 2 (switch) configuration, individual ports cannot be configured with an IP address (IPv4 or IPv6). In this situation, the switch has one IP address for the management port and one IP address for the system. This has previously been supported for IPv4 but not for IPv6.

There is support for configuring an IPv6 address on the management port as described in [“Configuring the management port for an IPv6 automatic address configuration”](#) on page 362, and for configuring a system-wide IPv6 address on a Layer 2 switch. Configuration of the system-wide IPv6 address is exactly like configuration of an IPv6 address in router mode, except that the IPv6 configuration is at the Global CONFIG level instead of at the Interface level.

The process for defining the system-wide interface for IPv6 is described in the following sections:

- [“Configuring a global or site-local IPv6 address with a manually configured interface ID”](#) on page 361

- [“Configuring a link-local IPv6 address as a system-wide address for a switch”](#) on page 361

NOTE

When configuring an IPv6 host address on a Layer 2 switch that has multiple VLANs, make sure the configuration includes a designated management VLAN that identifies the VLAN to which the global IP address belongs. Refer to [“Designated VLAN for Telnet management sessions to a Layer 2 Switch”](#) on page 120.

Configuring a global or site-local IPv6 address with a manually configured interface ID

To configure a global or site-local IPv6 address with a manually-configured interface ID, such as a system-wide address for a switch, enter a command similar to the following at the Global CONFIG level.

```
Brocade(config)#ipv6 address 2001:200:12D:1300:240:D0FF:FE48:4000:1/64
```

Syntax: `ipv6 address <ipv6-prefix>/<prefix-length>`

You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the `<prefix-length>` parameter in decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

Configuring a link-local IPv6 address as a system-wide address for a switch

To enable IPv6 and automatically configure a global interface enter commands such as the following.

```
Brocade(config)#ipv6 enable
```

This command enables IPv6 on the switch and specifies that the interface is assigned an automatically computed link-local address.

Syntax: `[no] ipv6 enable`

To override a link-local address that is automatically computed for the global interface with a manually configured address, enter a command such as the following.

```
Brocade(config)#ipv6 address FE80::240:D0FF:FE48:4672 link-local
```

This command explicitly configures the link-local address FE80::240:D0FF:FE48:4672 for the global interface.

Syntax: `ipv6 address <ipv6-address> link-local`

You must specify the `<ipv6-address>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **link-local** keyword indicates that the router interface should use the manually configured link-local address instead of the automatically computed link-local address.

Configuring the management port for an IPv6 automatic address configuration

You can have the management port configured to automatically obtain an IPv6 address. This process is the same for any other port and is described in detail in the section [“Configuring a global IPv6 address with an automatically computed EUI-64 interface ID”](#) on page 364

Configuring basic IPv6 connectivity on a Layer 3 switch

To configure basic IPv6 connectivity on a Brocade Layer 3 Switch, you must do the following:

- Enable IPv6 routing globally on the switch
- Configure an IPv6 address or explicitly enable IPv6 on each router interface over which you plan to forward IPv6 traffic
- Configure IPv4 and IPv6 protocol stacks. (This step is mandatory only if you want a router interface to send and receive both IPv4 and IPv6 traffic.)

All other configuration tasks in this chapter are optional.

Enabling IPv6 routing

By default, IPv6 routing is disabled. To enable the forwarding of IPv6 traffic globally on the Layer 3 switch, enter the following command.

```
Brocade(config)#ipv6 unicast-routing
```

Syntax: [no] ipv6 unicast-routing

To disable the forwarding of IPv6 traffic globally on the Brocade device, enter the **no** form of this command.

IPv6 configuration on each router interface

To forward IPv6 traffic on a router interface, the interface must have an IPv6 address, or IPv6 must be explicitly enabled. By default, an IPv6 address is not configured on a router interface.

If you choose to configure a global or site-local IPv6 address for an interface, IPv6 is also enabled on the interface. Further, when you configure a global or site-local IPv6 address, you must decide on one of the following in the low-order 64 bits:

- A manually configured interface ID.
- An automatically computed EUI-64 interface ID.

If you prefer to assign a link-local IPv6 address to the interface, you must explicitly enable IPv6 on the interface, which causes a link-local address to be automatically computed for the interface. If preferred, you can override the automatically configured link-local address with an address that you manually configure.

This section provides the following information:

- Configuring a global or site-local address with a manually configured or automatically computed interface ID for an interface.
- Automatically or manually configuring a link-local address for an interface.
- Configuring IPv6 anycast addresses

Configuring a global or site-local IPv6 address on an interface

Configuring a global or site-local IPv6 address on an interface does the following:

- Automatically configures an interface ID (a link-local address), if specified.
- Enables IPv6 on that interface.

Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast address assigned to the interface.
- Solicited-node for subnet anycast address for each unicast assigned address
- Solicited-node for anycast address FF02:0:0:0:0:1:FF00::0000
- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2

The neighbor discovery feature sends messages to these multicast groups. For more information, refer to [“IPv6 neighbor discovery configuration”](#) on page 384.

Configuring a global or site-local IPv6 address with a manually configured interface ID

To configure a global or site-local IPv6 address, including a manually configured interface ID, for an interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 address 2001:200:12D:1300:240:D0FF:
FE48:4672:/64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and the interface ID ::240:D0FF:FE48:4672, and enable IPv6 on Ethernet interface 3/1.

Syntax: `ipv6 address <ipv6-prefix>/<prefix-length>`

You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

To configure a /122 address on a VE enter commands similar to the following.

```
Brocade(config-vlan-11)#int ve11
Brocade(config-vif-11)#ipv6 add 2020::1/122
Brocade(config-vif-11)#sh ipv6 int
Routing Protocols : R - RIP O - OSPF
Interface      Status      Routing  Global Unicast Address
VE 11          up/up      2020::1/122
Brocade(config-vif-11)#sh ipv6 route
```

Configuring basic IPv6 connectivity on a Layer 3 switch

IPv6 Routing Table - 1 entries:

Type Codes: C - Connected, S - Static, R - RIP, O - OSPF, B - BGP

OSPF Sub Type Codes: O - Intra, Oi - Inter, Ol - Type1 external, O2 - Type2 external

Type	IPv6 Prefix	Next Hop Router	Interface	Dis/Metric
C	2020::/122	::	ve 11	0/0

Configuring a global IPv6 address with an automatically computed EUI-64 interface ID

To configure a global IPv6 address with an automatically computed EUI-64 interface ID in the low-order 64-bits, enter commands such as the following.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 address 2001:200:12D:1300::/64 eui-64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and an interface ID, and enable IPv6 on Ethernet interface 3/1.

Syntax: `ipv6 address <ipv6-prefix>/<prefix-length> eui-64`

You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

The **eui-64** keyword configures the global address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Configuring a link-local IPv6 address on an interface

To explicitly enable IPv6 on a router interface without configuring a global or site-local address for the interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 enable
```

These commands enable IPv6 on Ethernet interface 3/1 and specify that the interface is assigned an automatically computed link-local address.

Syntax: `[no] ipv6 enable`

NOTE

When configuring VLANs that share a common tagged interface with a physical or Virtual Ethernet (VE) interface, Brocade recommends that you override the automatically computed link-local address with a manually configured unique address for the interface. If the interface uses the automatically computed address, which in the case of physical and VE interfaces is derived from a global MAC address, all physical and VE interfaces will have the same MAC address.

To override a link-local address that is automatically computed for an interface with a manually configured address, enter commands such as the following.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 address FE80::240:D0FF:FE48:4672 link-local
```

These commands explicitly configure the link-local address FE80::240:DOFF:FE48:4672 for Ethernet interface 3/1.

Syntax: `ipv6 address <ipv6-address> link-local`

You must specify the `<ipv6-address>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **link-local** keyword indicates that the router interface should use the manually configured link-local address instead of the automatically computed link-local address.

Configuring an IPv6 anycast address on an interface

In IPv6, an **anycast** address is an address for a set of interfaces belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface configured with the anycast address.

An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign an IPv6 unicast address to multiple interfaces, it is an anycast address. On the Brocade device, you configure an interface assigned an anycast address to recognize the address as an anycast address.

For example, the following commands configure an anycast address on interface 2/1.

```
Brocade(config)#int e 2/1
Brocade(config-if-e1000-2/1)#ipv6 address 2002::/64 anycast
```

Syntax: `ipv6 address <ipv6-prefix>/<prefix-length> [anycast]`

IPv6 anycast addresses are described in detail in RFC 1884. See RFC 2461 for a description of how the IPv6 Neighbor Discovery mechanism handles anycast addresses.

Configuring IPv4 and IPv6 protocol stacks

One situation in which you must configure a router to run both IPv4 and IPv6 protocol stacks is if it is deployed as an endpoint for an IPv6 over IPv4 tunnel.

Each router interface that will send and receive both IPv4 and IPv6 traffic must be configured with an IPv4 address and an IPv6 address. (An alternative to configuring a router interface with an IPv6 address is to explicitly enable IPv6 using the **ipv6 enable** command. For more information about using this command, refer to [“Configuring a link-local IPv6 address on an interface”](#) on page 364.)

To configure a router interface to support both the IPv4 and IPv6 protocol stacks, use commands such as the following.

```
Brocade(config)#ipv6 unicast-routing
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ip address 192.168.1.1 255.255.255.0
Brocade(config-if-e1000-3/1)#ipv6 address 2001:200:12d:1300::/64 eui-64
```

These commands globally enable IPv6 routing and configure an IPv4 address and an IPv6 address for Ethernet interface 3/1.

Syntax: `[no] ipv6 unicast-routing`

To disable IPv6 traffic globally on the router, enter the **no** form of this command.

Syntax: `ip address <ip-address> <sub-net-mask> [secondary]`

You must specify the `<ip-address>` parameter using 8-bit values in dotted decimal notation.

You can specify the `<sub-net-mask>` parameter in either dotted decimal notation or as a decimal value preceded by a slash mark (/).

The **secondary** keyword specifies that the configured address is a secondary IPv4 address.

To remove the IPv4 address from the interface, enter the **no** form of this command.

Syntax: `ipv6 address <ipv6-prefix>/<prefix-length> [eui-64]`

This syntax specifies a global or site-local IPv6 address. For information about configuring a link-local IPv6 address, refer to [“Configuring a link-local IPv6 address on an interface”](#) on page 364.

You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

The **eui-64** keyword configures the global address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface’s MAC address. If you do not specify the **eui-64** keyword, you must manually configure the 64-bit interface ID as well as the 64-bit network prefix. For more information about manually configuring an interface ID, refer to [“Configuring a global or site-local IPv6 address on an interface”](#) on page 363.

IPv6 management (IPv6 host support)

You can configure a FastIron X Series switch to serve as an IPv6 host in an IPv6 network. An **IPv6 host** has IPv6 addresses on its interfaces, but does not have full IPv6 routing enabled on it.

This section describes the IPv6 host features supported on FastIron X Series devices.

Configuring IPv6 management ACLs

When you enter the **ipv6 access-list** command, the Brocade device enters the IPv6 Access List configuration level, where you can access several commands for configuring IPv6 ACL entries. After configuring the ACL entries, you can apply them to network management access features such as Telnet, SSH, Web, and SNMP.

NOTE

Unlike IPv4, there is no distinction between standard and extended ACLs in IPv6.

Example

```
FastIron(config)#ipv6 access-list netw
FastIron(config-ipv6-access-list-netw)#
```

Syntax: `[no] ipv6 access-list <ACL name>`

The `<ACL name>` variable specifies a name for the IPv6 ACL. An IPv6 ACL name cannot start with a numeral, for example, **1access**. Also, an IPv4 ACL and an IPv6 ACL cannot share the same name.

Restricting SNMP access to an IPv6 node

You can restrict SNMP access to the device to the IPv6 host whose IP address you specify. To do so, enter a command such as the following.

```
Brocade(config)#snmp-client ipv6 2001:efff:89::23
```

Syntax: `snmp-client ipv6 <ipv6-address>`

The `<ipv6-address>` you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specifying an IPv6 SNMP trap receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter a command such as the following.

```
Brocade(config)#snmp-server host ipv6 2001:efff:89::13
```

Syntax: `snmp-server host ipv6 <ipv6-address>`

The `<ipv6-address>` you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Configuring SNMP V3 over IPv6

Brocade FastIron X Series devices support IPv6 for SNMP version 3. For more information about how to configure SNMP, refer to [Chapter 10, “SNMP Access”](#).

Configuring SNTP over IPv6

To enable the Brocade device to send SNTP packets over IPv6, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#sntp server ipv6 3000::400
```

Syntax: `sntp server ipv6 <ipv6-address>`

The `<ipv6-address>` is the IPv6 address of the SNTP server. When you enter the IPv6 address, you do not need to specify the prefix length. A prefix length of 128 is implied.

Secure Shell, SCP, and IPv6

Secure Shell (SSH) is a mechanism that allows secure remote access to management functions on the Brocade device. SSH provides a function similar to Telnet. You can log in to and configure the Brocade device using a publicly or commercially available SSH client program, just as you can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the Brocade device.

To open an SSH session between an IPv6 host running an SSH client program and the Brocade device, open the SSH client program and specify the IPv6 address of the device. For more information about configuring SSH on the Brocade device, refer to [“SSH2 and SCP”](#) on page 179.

IPv6 Telnet

Telnet sessions can be established between a Brocade device to a remote IPv6 host, and from a remote IPv6 host to the Brocade device using IPv6 addresses.

The **telnet** command establishes a Telnet connection from a Brocade device to a remote IPv6 host using the console. Up to five **read-access** Telnet sessions are supported on the router at one time. **Write-access** through Telnet is limited to one session, and only one outgoing Telnet session is supported on the router at one time. To see the number of open Telnet sessions at any time, enter the **show telnet** command.

Example

To establish a Telnet connection to a remote host with the IPv6 address of 3001:2837:3de2:c37::6, enter the following command.

```
Brocade#telnet 3001:2837:3de2:c37::6
```

Syntax: **telnet** <ipv6-address> [<port-number> | **outgoing-interface ethernet** <port> | **ve** <number>]

The <ipv6-address> parameter specifies the address of a remote host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <port-number> parameter specifies the port number on which the Brocade device establishes the Telnet connection. You can specify a value between 1 - 65535. If you do not specify a port number, the Brocade device establishes the Telnet connection on port 23.

If the IPv6 address you specify is a link-local address, you must specify the **outgoing-interface ethernet <port> | ve <number>** parameter. This parameter identifies the interface that must be used to reach the remote host. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

Establishing a Telnet session from an IPv6 host

To establish a Telnet session from an IPv6 host to the Brocade device, open your Telnet application and specify the IPv6 address of the Layer 3 Switch.

IPv6 traceroute

NOTE

This section describes the **IPv6 traceroute** command. For details about **IPv4 traceroute**, refer to [“Tracing an IPv4 route”](#) on page 96.

The **traceroute** command allows you to trace a path from the Brocade device to an IPv6 host.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a minimum TTL of 1 second and a maximum TTL of 30 seconds. In addition, if there are multiple equal-cost routes to the destination, the Brocade device displays up to three responses.

For example, to trace the path from the Brocade device to a host with an IPv6 address of 3301:23dd:349e:a384::34, enter the following command:

```
Brocade#traceroute ipv6 3301:23dd:349e:a384::34
```

Syntax: **traceroute ipv6** <ipv6-address>

The <ipv6-address> parameter specifies the address of a host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

IPv6 Web management using HTTP and HTTPS

When you have an IPv6 management station connected to a switch with an IPv6 address applied to the management port, you can manage the switch from a Web browser by entering one of the following in the browser address field.

http://[<ipv6 address>]

or

https://[<ipv6 address>]

NOTE

You must enclose the IPv6 address with square brackets [] in order for the Web browser to work.

Restricting Web management access

You can restrict Web management access to include only management functions on a Brocade device that is acting as an IPv6 host, or restrict access so that the Brocade host can be reached by a specified IPv6 device.

Restricting Web management access by specifying an IPv6 ACL

You can specify an IPv6 ACL that restricts Web management access to management functions on the device that is acting as the IPv6 host.

Example

```
Brocade(config)#access-list 12 deny host 2000:2383:e0bb::2/128 log
Brocade(config)#access-list 12 deny 30ff:3782::ff89/128 log
Brocade(config)#access-list 12 deny 3000:4828::fe19/128 log
Brocade(config)#access-list 12 permit any
Brocade(config)#web access-group ipv6 12
```

Syntax: `web access-group ipv6 <ipv6 ACL name>`

where <ipv6 ACL name> is a valid IPv6 ACL.

Restricting Web management access to an IPv6 host

You can restrict Web management access to the device to the IPv6 host whose IP address you specify. No other device except the one with the specified IPv6 address can access the Web Management Interface.

Example

```
Brocade(config)#web client ipv6 3000:2383:e0bb::2/128
```

Syntax: `web client ipv6 <ipv6-address>`

The <ipv6-address> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Configuring name-to-IPv6 address resolution using IPv6 DNS resolver

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet and ping commands. You can also define a DNS domain on a Brocade device and thereby recognize all hosts within that domain. After you define a domain name, the Brocade device automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain “newyork.com” is defined on a Brocade device, and you want to initiate a ping to host “NYC01” on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping.

```
Brocade#ping ipv6 nyc01
Brocade#ping ipv6 nyc01.newyork.com
```

Defining an IPv6 DNS entry

IPv6 defines new DNS record types to resolve queries for domain names to IPv6 addresses, as well as IPv6 addresses to domain names. Brocade devices running IPv6 software support AAAA DNS records, which are defined in RFC 1886.

AAAA DNS records are analogous to the A DNS records used with IPv4. They store a complete IPv6 address in each record. AAAA records have a type value of 28.

To establish an IPv6 DNS entry for the device, enter the following command.

```
Brocade(config)#ipv6 dns domain-name companynet.com
```

Syntax: [no] **ipv6 dns domain-name** <domain name>

To define an IPv6 DNS server address, enter the following command.

```
Brocade(config)#ipv6 dns server-address 200::1
```

Syntax: [no] **ipv6 dns server-address** <ipv6-addr> [<ipv6-addr>] [<ipv6-addr>] [<ipv6-addr>]

As an example, in a configuration where ftp6.companynet.com is a server with an IPv6 protocol stack, when a user pings ftp6.companynet.com, the Brocade device attempts to resolve the AAAA DNS record. In addition, if the DNS server does not have an IPv6 address, as long as it is able to resolve AAAA records, it can still respond to DNS queries.

Pinging an IPv6 address

NOTE

This section describes the *IPv6 ping* command. For details about *IPv4 ping*, refer to [“Pinging an IPv4 address”](#) on page 95.

The **ping** command allows you to verify the connectivity from a Brocade device to an IPv6 device by performing an ICMP for IPv6 echo test.

For example, to ping a device with the IPv6 address of 2001:3424:847f:a385:34dd::45 from the Brocade device, enter the following command.

```
Brocade#ping ipv6 2001:3424:847f:a385:34dd::45
```


Syntax: `ping ipv6 <ipv6-address> [outgoing-interface [<port> | ve <number>]] [source <ipv6-address>] [count <number>] [timeout <milliseconds>] [ttl <number>] [size <bytes>] [quiet] [numeric] [no-fragment] [verify] [data <1-to-4 byte hex>] [brief]`

- The `<ipv6-address>` parameter specifies the address of the router. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.
- The **outgoing-interface** keyword specifies a physical interface over which you can verify connectivity. If you specify a physical interface, such as an Ethernet interface, you must also specify the port number of the interface. If you specify a virtual interface, such as a VE, you must specify the number associated with the VE.
- The **source** `<ipv6-address>` parameter specifies an IPv6 address to be used as the origin of the ping packets.
- The **count** `<number>` parameter specifies how many ping packets the router sends. You can specify from 1 - 4294967296. The default is 1.
- The **timeout** `<milliseconds>` parameter specifies how many milliseconds the router waits for a reply from the pinged device. You can specify a timeout from 1 - 4294967296 milliseconds. The default is 5000 (5 seconds).
- The **ttl** `<number>` parameter specifies the maximum number of hops. You can specify a TTL from 1 - 255. The default is 64.
- The **size** `<bytes>` parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 - 4000. The default is 16.
- The **no-fragment** keyword turns on the "do not fragment" bit in the IPv6 header of the ping packet. This option is disabled by default.
- The **quiet** keyword hides informational messages such as a summary of the ping parameters sent to the device, and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.
- The **verify** keyword verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.
- The **data** `<1 - 4 byte hex>` parameter lets you specify a specific data pattern for the payload instead of the default data pattern, "abcd", in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE

For parameters that require a numeric value, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

- The **brief** keyword causes ping test characters to be displayed. The following ping test characters are supported.

! Indicates that a reply was received.

. Indicates that the network server timed out while waiting for a reply.

U Indicates that a destination unreachable error PDU was received.

I Indicates that the user interrupted ping.

Configuring an IPv6 Syslog server

To enable IPv6 logging, specify an IPv6 Syslog server. Enter a command such as the following.

```
Brocade(config)#log host ipv6 2000:2383:e0bb::4/128
```

Syntax: `log host ipv6 <ipv6-address> [<udp-port-num>]`

The `<ipv6-address>` must be in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<udp-port-num>` optional parameter specifies the UDP application port used for the Syslog facility.

Viewing IPv6 SNMP server addresses

Some of the **show** commands display IPv6 addresses for IPv6 SNMP servers. The following shows an example output for the **show snmp server** command.

```
Brocade#show snmp server
```

```
    Contact:
    Location:
Community(ro): .....
```

```
Traps
```

```
    Warm/Cold start: Enable
    Link up: Enable
    Link down: Enable
    Authentication: Enable
    Locked address violation: Enable
    Power supply failure: Enable
    Fan failure: Enable
    Temperature warning: Enable
    STP new root: Enable
    STP topology change: Enable
    vsrp: Enable
```

```
Total Trap-Receiver Entries: 4
```

Trap-Receiver	IP-Address	Port-Number	Community
1	192.147.201.100	162
2	4000::200	162
3	192.147.202.100	162
4	3000::200	162

Disabling router advertisement and solicitation messages

Router advertisement and solicitation messages enable a node on a link to discover the routers on the same link. By default, router advertisement and solicitation messages are permitted on the device. To disable these messages, configure an IPv6 access control list that denies them. The following shows an example configuration.

Example

```
Brocade(config)#ipv6 access-list rtradvert
Brocade(config)#deny icmp any any router-advertisement
Brocade(config)#deny icmp any any router-solicitation
Brocade(config)#permit ipv6 any any
```

Disabling IPv6 on a Layer 2 switch

IPv6 is enabled by default in the Layer 2 switch code. If desired, you can disable IPv6 on a global basis on a device running the switch code. To do so, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)#no ipv6 enable
```

Syntax: no ipv6 enable

To re-enable IPv6 after it has been disabled, enter **ipv6 enable**.

NOTE

IPv6 is disabled by default in the router code and must be configured on each interface that will support IPv6.

Static IPv6 route configuration

NOTE

Static IPv6 route configuration is supported only with the IPv6 Layer 3 PROM and the full Layer 3 image.

You can configure a static IPv6 route to be redistributed into a routing protocol, but you cannot redistribute routes learned by a routing protocol into the static IPv6 routing table.

NOTE

The maximum IPv6 static routes supported on an ICX 6450 device is 1070.

Before configuring a static IPv6 route, you must enable the forwarding of IPv6 traffic on the Layer 3 switch using the **ipv6 unicast-routing** command and enable IPv6 on at least one interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface. For more information on performing these configuration tasks, refer to [“Configuring IPv4 and IPv6 protocol stacks”](#) on page 365.

Configuring a static IPv6 route

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32, a next-hop gateway with the global address 4fee:2343:0:ee44::1, and an administrative distance of 110, enter the following command.

Static IPv6 route configuration

```
Brocade(config)#ipv6 route 8eff::0/32 4fee:2343:0:ee44::1 distance 110
```

Syntax: `ipv6 route <dest-ipv6-prefix>/<prefix-length> <next-hop-ipv6-address> [<metric>] [distance <number>]`

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32 and a next-hop gateway with the link-local address fe80::1 that the Layer 3 switch can access through Ethernet interface 3/1, enter the following command.

```
Brocade(config)#ipv6 route 8eff::0/32 ethernet 1 fe80::1
```

Syntax: `ipv6 route <dest-ipv6-prefix>/<prefix-length> [ethernet <slot/port> | ve <num>] <next-hop-ipv6-address> [<metric>] [distance <number>]`

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32 and a next-hop gateway that the Layer 3 switch can access through tunnel 1, enter the following command.

```
Brocade(config)#ipv6 route 8eff::0/32 tunnel 1
```

Syntax: `ipv6 route <dest-ipv6-prefix>/<prefix-length> <interface> <port> [<metric>] [distance <number>]`

[Table 61](#) describes the parameters associated with this command and indicates the status of each parameter.

TABLE 61 Static IPv6 route parameters

Parameter	Configuration details	Status
The IPv6 prefix and prefix length of the route's destination network.	<p>You must specify the <code><dest-ipv6-prefix></code> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.</p> <p>You must specify the <code><prefix-length></code> parameter as a decimal value. A slash mark (/) must follow the <code><ipv6-prefix></code> parameter and precede the <code><prefix-length></code> parameter.</p>	Mandatory for all static IPv6 routes.
<p>The route's next-hop gateway, which can be one of the following:</p> <ul style="list-style-type: none"> • The IPv6 address of a next-hop gateway. • A tunnel interface. 	<p>You can specify the next-hop gateway as one of the following types of IPv6 addresses:</p> <ul style="list-style-type: none"> • A global address. • A link-local address. <p>If you specify a global address, you do not need to specify any additional parameters for the next-hop gateway. If you specify a link-local address, you must also specify the interface through which to access the address. You can specify one of the following interfaces:</p> <ul style="list-style-type: none"> • An Ethernet interface. • A tunnel interface. • A virtual interface (VE). <p>If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number. You can also specify the next-hop gateway as a tunnel interface. If you specify a tunnel interface, also specify the tunnel number.</p>	Mandatory for all static IPv6 routes.
The route's metric.	You can specify a value from 1 – 16.	Optional for all static IPv6 routes. (The default metric is 1.)
The route's administrative distance.	You must specify the distance keyword and any numerical value.	Optional for all static IPv6 routes. (The default administrative distance is 1.)

A metric is a value that the Layer 3 switch uses when comparing this route to other static routes in the IPv6 static route table that have the same destination. The metric applies only to routes that the Layer 3 switch has already placed in the IPv6 static route table.

The administrative distance is a value that the Layer 3 switch uses to compare this route with routes from other route sources that have the same destination. (The Layer 3 switch performs this comparison before placing a route in the IPv6 route table.) This parameter does not apply to routes that are already in the IPv6 route table. In general, a low administrative distance indicates a preferred route. By default, static routes take precedence over routes learned by routing protocols. If you want a dynamic route to be chosen over a static route, you can configure the static route with a higher administrative distance than the dynamic route.

IPv6 over IPv4 tunnels

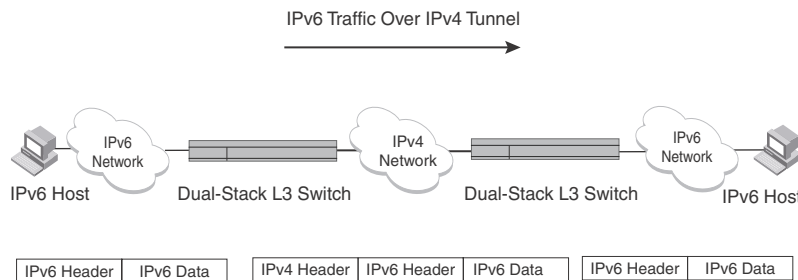
NOTE

This feature is supported only with the IPv6 Layer 3 PROM and the full Layer 3 image.

To enable communication between isolated IPv6 domains using the IPv4 infrastructure, you can manually configure IPv6 over IPv4 tunnels that provide static point-point connectivity.

As shown in [Figure 32](#), these tunnels encapsulate an IPv6 packet within an IPv4 packet.

FIGURE 32 IPv6 over an IPv4 tunnel



In general, a manually configured tunnel establishes a permanent link between switches in IPv6 domains. A manually configured tunnel has explicitly configured IPv4 addresses for the tunnel source and destination.

This tunneling mechanism requires that the Layer 3 switch at each end of the tunnel run both IPv4 and IPv6 protocol stacks. The Layer 3 switches running both protocol stacks, or dual-stack routers, can interoperate directly with both IPv4 and IPv6 end systems and routers. Refer to [“Configuring IPv4 and IPv6 protocol stacks”](#) on page 365.

IPv6 over IPv4 tunnel configuration notes

- This feature is not supported in a mixed hardware configuration with 48-port 10/100/1000 Mbps Ethernet POE (SX-FI48GPP) interface modules, together with IPv4 or IPv6 interface modules, or management modules with user ports.

NOTE

If the 48-port 10/100/1000 Mbps Ethernet POE (SX-FI48GPP) interface module is inadvertently inserted in a system that has IPv4 or IPv6 interface modules, or a management module with user ports, existing tunnels will be taken down immediately. To recover, physically remove the module that caused the mix-and-match condition and then disable and re-enable the tunnel interfaces.

- The local tunnel configuration must include both source and destination addresses.
- The remote side of the tunnel must have the opposite source/destination pair.
- A tunnel interface supports static and dynamic IPv6 configuration settings and routing protocols.
- Duplicate Address Detection (DAD) is not currently supported with IPv6 tunnels. Make sure tunnel endpoints do not have duplicate IP addresses.
- Neighbor Discovery (ND) is not supported with IPv6 tunnels.

- If a tunnel source port is a multi-homed IPv4 source, the tunnel will use the first IPv4 address only. For proper tunnel operation, use the **ip address** option.
- Hitless management is not supported with IPv6-over-IPv4 tunnels or GRE tunnels. When either of these tunnel types are enabled on non supported devices, the behavior is as follows:
 - The CLI commands that execute a hitless switchover (**switch-over-active-role** and **hitless reload**) are disabled.
 - In the event of a failover, the following message is displayed on the console and in the Syslog, and the entire system is rebooted:

```
WARNING: All the System will now reload since GRE or IPv6 Tunnel was
configured
```

Configuring a manual IPv6 tunnel

You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnelling mechanism if you need a permanent and stable connection.

To configure a manual IPv6 tunnel, enter commands such as the following on a Layer 3 Switch running both IPv4 and IPv6 protocol stacks on each end of the tunnel.

```
Brocade(config)#interface tunnel 1
Brocade(config-tnif-1)#tunnel source ethernet 3/1
Brocade(config-tnif-1)#tunnel destination 198.162.100.1
Brocade(config-tnif-1)#tunnel mode ipv6ip
Brocade(config-tnif-1)#ipv6 enable
```

This example creates tunnel interface 1 and assigns a link local IPv6 address with an automatically computed EUI-64 interface ID to it. The IPv4 address assigned to Ethernet interface 3/1 is used as the tunnel source, while the IPv4 address 192.168.100.1 is configured as the tunnel destination. The tunnel mode is specified as a manual IPv6 tunnel. Finally, the tunnel is enabled. Note that instead of entering **ipv6 enable**, you could specify an IPv6 address, for example, **ipv6 address 2001:b78:384d:34::/64 eui-64**, which would also enable the tunnel.

Syntax: [no] interface tunnel <number>

For the <number> parameter, specify a value between 1 – 8.

Syntax: [no] tunnel source <ipv4-address> | ethernet <port> | loopback <number> | ve <number>

The tunnel source can be an IP address or an interface.

For <ipv4-address>, use 8-bit values in dotted decimal notation.

The **ethernet | loopback | ve** parameter specifies an interface as the tunnel source. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, VE, or interface, also specify the loopback, VE, or number, respectively.

Syntax: [no] tunnel destination <ipv4-address>

Specify the <ipv4-address> parameter using 8-bit values in dotted decimal notation.

Syntax: [no] tunnel mode ipv6ip

ipv6ip indicates that this is an IPv6 manual tunnel.

Syntax: ipv6 enable

The **ipv6 enable** command enables the tunnel. Alternatively, you could specify an IPv6 address, which would also enable the tunnel.

Syntax: **ipv6 address** *<ipv6-prefix>/<prefix-length>* **[eui-64]**

The **ipv6 address** command enables the tunnel. Alternatively, you could enter **ipv6 enable**, which would also enable the tunnel.

Specify the *<ipv6-prefix>* parameter in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specify the *<prefix-length>* parameter as a decimal value. A slash mark (/) must follow the *<ipv6-prefix>* parameter and precede the *<prefix-length>* parameter. The **eui-64** keyword configures the global address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Clearing IPv6 tunnel statistics

You can clear statistics (reset all fields to zero) for all IPv6 tunnels or for a specific tunnel interface.

For example, to clear statistics for tunnel 1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
Brocade#clear ipv6 tunnel 1
```

To clear statistics for all IPv6 tunnels, enter the following command.

```
Brocade#clear ipv6 tunnel
```

Syntax: **clear ipv6 tunnel** [*<number>*]

The *<number>* parameter specifies the tunnel number.

Displaying IPv6 tunnel information

Use the commands in this section to display the configuration, status, and counters associated with IPv6 tunnels.

Displaying a summary of tunnel information

To display a summary of tunnel information, enter the following command at any level of the CLI.

```
Brocade#show ipv6 tunnel
IP6 Tunnels
  Tunnel  Mode      Packet Received  Packet Sent
  1       configured    0                0
  2       configured    0              22419
```

Syntax: **show ipv6 tunnel**

This display shows the following information.

TABLE 62 IPv6 tunnel summary information

Field	Description
Tunnel	The tunnel interface number.
Mode	The tunnel mode. Possible modes include the following: <ul style="list-style-type: none"> configured – Indicates a manually configured tunnel.
Packet Received	The number of packets received by a tunnel interface. Note that this is the number of packets received by the CPU. It does not include the number of packets processed in hardware.
Packet Sent	The number of packets sent by a tunnel interface. Note that this is the number of packets sent by the CPU. It does not include the number of packets processed in hardware.

Displaying tunnel interface information

To display status and configuration information for tunnel interface 1, enter the following command at any level of the CLI.

```
Brocade#show interfaces tunnel 1
Tunnell is up, line protocol is up
  Hardware is Tunnel
  Tunnel source ve 30
  Tunnel destination is 2.2.2.10
  Tunnel mode ipv6ip
  No port name
  MTU 1480 bytes, encapsulation IPV4
```

Syntax: `show interfaces tunnel <number>`

The `<number>` parameter indicates the tunnel interface number for which you want to display information.

This display shows the following information.

TABLE 63 IPv6 tunnel interface information

Field	Description
Tunnel interface status	The status of the tunnel interface can be one of the following: <ul style="list-style-type: none"> up – The tunnel mode is set and the tunnel interface is enabled. down – The tunnel mode is not set. administratively down – The tunnel interface was disabled with the <code>disable</code> command.
Line protocol status	The status of the line protocol can be one of the following: <ul style="list-style-type: none"> up – IPv4 connectivity is established. down – The line protocol is not functioning and is down.
Hardware is tunnel	The interface is a tunnel interface.
Tunnel source	The tunnel source can be one of the following: <ul style="list-style-type: none"> An IPv4 address The IPv4 address associated with an interface/port.
Tunnel destination	The tunnel destination can be an IPv4 address.
Tunnel mode	The tunnel mode can be the following: <ul style="list-style-type: none"> ipv6ip – indicates a manually configured tunnel

TABLE 63 IPv6 tunnel interface information (Continued)

Field	Description
Port name	The port name configured for the tunnel interface.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying interface level IPv6 settings

To display Interface level IPv6 settings for tunnel interface 1, enter the following command at any level of the CLI.

```
Brocade#show ipv6 inter tunnel 1
Interface Tunnel 1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::3:4:2 [Preferred]
  Global unicast address(es):
    1001::1 [Preferred], subnet is 1001::/64
    1011::1 [Preferred], subnet is 1011::/64
  Joined group address(es):
    ff02::1:ff04:2
    ff02::5
    ff02::1:ff00:1
    ff02::2
    ff02::1
  MTU is 1480 bytes
  ICMP redirects are enabled
  No Inbound Access List Set
  No Outbound Access List Set
  OSPF enabled
```

The display command above reflects the following configuration.

```
Brocade#show running-config interface tunnel 1
!
interface tunnel 1
  port-name ManualTunnell
  tunnel mode ipv6ip
  tunnel source loopback 1
  tunnel destination 2.1.1.1
  ipv6 address 1011::1/64
  ipv6 address 1001::1/64
  ipv6 ospf area 0
```

This display shows the following information.

TABLE 64 Interface level IPv6 tunnel information

Field	Description
Interface Tunnel status	<p>The status of the tunnel interface can be one of the following:</p> <ul style="list-style-type: none"> • up – IPv4 connectivity is established. • down – The tunnel mode is not set. • administratively down – The tunnel interface was disabled with the disable command.
Line protocol status	<p>The status of the line protocol can be one of the following:</p> <ul style="list-style-type: none"> • up – IPv6 is enabled through the ipv6 enable or ipv6 address command. • down – The line protocol is not functioning and is down.

ECMP load sharing for IPv6

The IPv6 route table selects the best route to a given destination from among the routes in the tables maintained by the configured routing protocols (BGP4, OSPF, static, and so on). The IPv6 route table can contain more than one path to a given destination. When this occurs, the Brocade device selects the path with the lowest cost for insertion into the routing table. If more than one path with the lowest cost exists, all of these paths are inserted into the routing table, subject to the configured maximum number of load sharing paths (by default 4). The device uses **Equal-Cost Multi-Path (ECMP) load sharing** to select a path to a destination.

When a route is installed by routing protocols or configured static route for the first time, and the IPv6 route table contains multiple, equal-cost paths to that route, the device checks the IPv6 neighbor for each next hop. Every next hop where the link layer address has been resolved will be stored in hardware. The device will initiate neighbor discovery for the next hops whose link layer addresses are not resolved. The hardware will hash the packet and choose one of the paths. The number of paths would be updated in hardware as the link layer gets resolved for a next hop.

If the path selected by the device becomes unavailable, the IPv6 neighbor should change state and trigger the update of the destination in the hardware.

Brocade devices support network-based ECMP load-sharing methods for IPv6 traffic. The Brocade device distributes traffic across equal-cost paths based on a XOR of some bits from the MAC source address, MAC destination address, IPv6 source address, IPv6 destination address, IPv6 flow label, IPv6 next header. The software selects a path based on a calculation involving the maximum number of load-sharing paths allowed and the actual number of paths to the destination network. This is the default ECMP load-sharing method for IPv6.

You can manually disable or enable ECMP load sharing for IPv6 and specify the number of equal-cost paths the device can distribute traffic across. In addition, you can display information about the status of ECMP load-sharing on the device.

Disabling or re-enabling ECMP load sharing for IPv6

ECMP load sharing for IPv6 is enabled by default. To disable the feature, enter the following command.

```
Brocade(config)#no ipv6 load-sharing
```

If you want to re-enable the feature after disabling it, you must specify the number of load-sharing paths. The maximum number of paths the device supports is a value from 2–8. By entering a command such as the following, IPv6 load-sharing will be re-enabled.

```
Brocade(config)#ipv6 load-sharing 4
```

Syntax: `[no] ipv6 load-sharing <num>`

The `<num>` parameter specifies the number of paths and can be from 2–8. The default is 4.

Changing the maximum load sharing paths for IPv6

By default, IPv6 ECMP load sharing allows traffic to be balanced across up to four equal paths. You can change the maximum number of paths the device supports to a value from 2–8.

To change the number of ECMP load sharing paths for IPv6, enter a command such as the following.

```
Brocade(config)#ipv6 load-sharing 6
```

Syntax: `[no] ipv6 load-sharing [<num>]`

The `<num>` parameter specifies the number of paths and can be from 2–8. The default is 4.

Enabling support for network-based ECMP load sharing for IPv6

Network-based ECMP load sharing is supported. In this configuration, traffic is distributed across equal-cost paths based on the destination network address. Routes to each network are stored in CAM and accessed when a path to a network is required. Because multiple hosts are likely to reside on a network, this method uses fewer CAM entries.

Displaying ECMP load-sharing information for IPv6

To display the status of ECMP load sharing for IPv6, enter the following command.

```
Brocade#show ipv6
Global Settings
  unicast-routing enabled, hop-limit 64
  No Inbound Access List Set
  No Outbound Access List Set
  Prefix-based IPv6 Load-sharing is Enabled, Number of load share paths: 4
```

Syntax: `show ipv6`

IPv6 ICMP feature configuration

As with the Internet Control Message Protocol (ICMP) for IPv4, ICMP for IPv6 provides error and informational messages. Implementation of the stateless auto configuration, neighbor discovery, and path MTU discovery features use ICMP messages.

This section explains how to configure following IPv6 ICMP features:

- ICMP rate limiting
- ICMP redirects

Configuring ICMP rate limiting

You can limit the rate at which IPv6 ICMP error messages are sent out on a network. IPv6 ICMP implements a token bucket algorithm.

To illustrate how this algorithm works, imagine a virtual bucket that contains a number of tokens. Each token represents the ability to send one ICMP error message. Tokens are placed in the bucket at a specified interval until the maximum number of tokens allowed in the bucket is reached. For each error message that ICMP sends, a token is removed from the bucket. If ICMP generates a series of error messages, messages can be sent until the bucket is empty. If the bucket is empty of tokens, error messages cannot be sent until a new token is placed in the bucket.

You can adjust the following elements related to the token bucket algorithm:

- The interval at which tokens are added to the bucket. The default is 100 milliseconds.
- The maximum number of tokens in the bucket. The default is 10 tokens.

For example, to adjust the interval to 1000 milliseconds and the number of tokens to 100 tokens, enter the following command.

```
Brocade(config)# ipv6 icmp error-interval 1000 100
```

Syntax: `ipv6 icmp error-interval <interval> [<number-of-tokens>]`

The interval in milliseconds at which tokens are placed in the bucket can range from 0 – 2147483647. The maximum number of tokens stored in the bucket can range from 1 – 200.

NOTE

If you retain the default interval value or explicitly set the value to 100 milliseconds, output from the **show run** command does not include the setting of the **ipv6 icmp error-interval** command because the setting is the default.

Also, if you configure the interval value to a number that does not evenly divide into 100000 (100 milliseconds), the system rounds up the value to a next higher value that does divide evenly into 100000. For example, if you specify an interval value of 150, the system rounds up the value to 200.

ICMP rate limiting is enabled by default. To disable ICMP rate limiting, set the interval to zero.

Enabling IPv6 ICMP redirect messages

NOTE

This feature is supported only with the IPv6 Layer 3 PROM and the full Layer 3 image.

You can enable a Layer 3 switch to send an IPv6 ICMP redirect message to a neighboring host to inform it of a better first-hop router on a path to a destination. By default, the sending of IPv6 ICMP redirect messages by a Layer 3 switch is disabled. (For more information about how ICMP redirect messages are implemented for IPv6, refer to “[IPv6 neighbor discovery configuration](#)” on page 384.)

NOTE

This feature is supported on Virtual Ethernet (VE) interfaces only.

For example, to enable the sending of IPv6 ICMP redirect messages on VE 2, enter the following commands.

```
Brocade(config)#interface ve2
Brocade(config-vif-2)#ipv6 redirects
```

To disable the sending of IPv6 ICMP redirect messages after it has been enabled on VE 2, enter the following commands.

```
Brocade(config)#interface ve2
Brocade(config-vif-2)#no ipv6 redirects
```

Syntax: [no] ipv6 redirects

Use the **show ipv6 interface** command to verify that the sending of IPv6 ICMP redirect messages is enabled on a particular interface.

IPv6 neighbor discovery configuration

The neighbor discovery feature for IPv6 uses IPv6 ICMP messages to do the following tasks:

- Determine the link-layer address of a neighbor on the same link.
- Verify that a neighbor is reachable.
- Track neighbor routers.

An IPv6 host is required to listen for and recognize the following addresses that identify itself:

- Link-local address.
- Assigned unicast address.
- Loopback address.
- All-nodes multicast address.
- Solicited-node multicast address.
- Multicast address to all other groups to which it belongs.

You can adjust the following IPv6 neighbor discovery features:

- Neighbor solicitation messages for duplicate address detection.
- Router advertisement messages:

- Interval between router advertisement messages.
- Value that indicates a router is advertised as a default router (for use by all nodes on a given link).
- Prefixes advertised in router advertisement messages.
- Flags for host stateful autoconfiguration.
- Amount of time during which an IPv6 node considers a remote node reachable (for use by all nodes on a given link).

IPv6 neighbor discovery configuration notes

NOTE

For all solicitation and advertisement messages, Brocade uses seconds as the unit of measure instead of milliseconds.

- If you add a port to a port-based VLAN, and the port has IPv6 neighbor discovery configuration, the system will clean up the neighbor discovery configuration from the port and display the following message on the console.
`ND6 port config on the new member ports removed`
- Neighbor discovery is not supported on tunnel interfaces.

Neighbor solicitation and advertisement messages

Neighbor solicitation and advertisement messages enable a node to determine the link-layer address of another node (neighbor) on the same link. (This function is similar to the function provided by the Address Resolution Protocol [ARP] in IPv4.) For example, node 1 on a link wants to determine the link-layer address of node 2 on the same link. To do so, node 1, the source node, multicasts a neighbor solicitation message. The neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, contains the following information:

- **Source address:** IPv6 address of node 1 interface that sends the message.
- **Destination address:** solicited-node multicast address (FF02:0:0:0:0:1:FF00::/104) that corresponds the IPv6 address of node 2.
- Link-layer address of node 1.
- A query for the link-layer address of node 2.

After receiving the neighbor solicitation message from node 1, node 2 replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header. The neighbor solicitation message contains the following information:

- **Source address:** IPv6 address of the node 2 interface that sends the message.
- **Destination address:** IPv6 address of node 1.
- Link-layer address of node 2.

After node 1 receives the neighbor advertisement message from node 2, nodes 1 and 2 can now exchange packets on the link.

After the link-layer address of node 2 is determined, node 1 can send neighbor solicitation messages to node 2 to verify that it is reachable. Also, nodes 1, 2, or any other node on the same link can send a neighbor advertisement message to the all-nodes multicast address (FF02::1) if there is a change in their link-layer address.

Router advertisement and solicitation messages

Router advertisement and solicitation messages enable a node on a link to discover the routers on the same link.

Each configured router interface on a link sends out a router advertisement message, which has a value of 134 in the Type field of the ICMP packet header, periodically to the all-nodes link-local multicast address (FF02::1).

A configured router interface can also send a router advertisement message in response to a router solicitation message from a node on the same link. This message is sent to the unicast IPv6 address of the node that sent the router solicitation message.

At system startup, a host on a link sends a router solicitation message to the all-routers multicast address (FF01). Sending a router solicitation message, which has a value of 133 in the Type field of the ICMP packet header, enables the host to automatically configure its IPv6 address immediately instead of awaiting the next periodic router advertisement message.

Because a host at system startup typically does not have a unicast IPv6 address, the source address in the router solicitation message is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a unicast IPv6 address, the source address is the unicast IPv6 address of the host interface sending the router solicitation message.

Entering the **ipv6 unicast-routing** command automatically enables the sending of router advertisement messages on all configured router Ethernet interfaces. You can configure several router advertisement message parameters. For information about disabling the sending of router advertisement messages and the router advertisement parameters that you can configure, refer to [“Enabling and disabling IPv6 router advertisements”](#) on page 390 and [“Setting IPv6 router advertisement parameters”](#) on page 387.

Neighbor redirect messages

After forwarding a packet, by default, a router can send a neighbor redirect message to a host to inform it of a better first-hop router. The host receiving the neighbor redirect message will then readdress the packet to the better router.

A router sends a neighbor redirect message only for unicast packets, only to the originating node, and to be processed by the node.

A neighbor redirect message has a value of 137 in the Type field of the ICMP packet header.

Setting neighbor solicitation parameters for duplicate address detection

Although the stateless auto configuration feature assigns the 64-bit interface ID portion of an IPv6 address using the MAC address of the host's NIC, duplicate MAC addresses can occur. Therefore, the duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless auto configuration feature. Duplicate address detection verifies that a unicast IPv6 address is unique.

If duplicate address detection identifies a duplicate unicast IPv6 address, the address is not used. If the duplicate address is the link-local address of the host interface, the interface stops processing IPv6 packets.

NOTE

Duplicate Address Detection (DAD) is not currently supported with IPv6 tunnels. Make sure tunnel endpoints do not have duplicate IP addresses.

You can configure the following neighbor solicitation message parameters that affect duplicate address detection while it verifies that a tentative unicast IPv6 address is unique:

- The number of consecutive neighbor solicitation messages that duplicate address detection sends on an interface. By default, duplicate address detection sends three neighbor solicitation messages without any follow-up messages.
- The interval in seconds at which duplicate address detection sends a neighbor solicitation message on an interface. By default, duplicate address detection sends a neighbor solicitation message every 1000 milliseconds.

For example, to change the number of neighbor solicitation messages sent on Ethernet interface 3/1 to two and the interval between the transmission of the two messages to 9 seconds, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 nd dad attempt 2
Brocade(config-if-e1000-3/1)#ipv6 nd ns-interval 9000
```

Syntax: `[no] ipv6 nd dad attempt <number>`

Syntax: `[no] ipv6 nd ns-interval <number>`

For the number of neighbor solicitation messages, specify a number from 0 – 255. The default is 3. Configuring a value of 0 disables duplicate address detection processing on the specified interface. To restore the number of messages to the default value, use the **no** form of this command.

For the interval between neighbor solicitation messages and the value for the retrans timer in router advertisements, specify a number from 0 – 4294967295 milliseconds. The default value for the interval between neighbor solicitation messages is 1000 milliseconds. The default value for the retrans timer is 0. Brocade does not recommend very short intervals in normal IPv6 operation. When a non-default value is configured, the configured time is both advertised and used by the router itself. To restore the default interval, use the **no** form of this command.

Setting IPv6 router advertisement parameters

You can adjust the following parameters for router advertisement messages:

- The interval (in seconds) at which an interface sends router advertisement messages. By default, an interface sends a router advertisement message every 200 seconds.
- The "router lifetime" value, which is included in router advertisements sent from a particular interface. The value (in seconds) indicates if the router is advertised as a default router on this interface. If you set the value of this parameter to 0, the router is not advertised as a default router on an interface. If you set this parameter to a value that is not 0, the router is advertised as a default router on this interface. By default, the router lifetime value included in router advertisement messages sent from an interface is 1800 seconds.
- The hop limit to be advertised in the router advertisement.

When adjusting these parameter settings, Brocade recommends that the interval between router advertisement transmission be less than or equal to the router lifetime value if the router is advertised as a default router. For example, to adjust the interval of router advertisements to 300 seconds and the router lifetime value to 1900 seconds on Ethernet interface 3/1, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 nd ra-interval 300
Brocade(config-if-e1000-3/1)#ipv6 nd ra-lifetime 1900
Brocade(config-if-e1000-3/1)#ipv6 nd ra-hop-limit 1
```

Here is another example with a specified range.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 nd ra-interval range 33 55
Brocade(config-if-e1000-3/1)#ipv6 nd ra-lifetime 1900
Brocade(config-if-e1000-3/1)#ipv6 nd ra-hop-limit 1
```

Syntax: [no] **ipv6 nd ra-interval** <number> | <min range value> <max range value>

Syntax: [no] **ipv6 nd ra-lifetime** <number>

Syntax: **ipv6 nd ra-hop-limit** <number>

<number> is a value from 0 – 255. The default is 64.

The **ipv6 nd ra-interval** <number> can be a value between 3 – 1800 seconds. The default is 200 seconds. The actual RA interval will be from .5 to 1.5 times the configured or default value. For example, in the above configuration, for **ipv6 nd ra-interval 300**, the range would be 150 – 450. To restore the default interval of 200 seconds, use the **no** form of the command.

The **ipv6 nd ra-interval range** <min range value> <max range value> command lets you specify a range of values instead of a single value.

The <min range value> specifies the minimum number of seconds allowed between sending unsolicited multicast router advertisements from the interface. The default is 0.33 times the <max range value> if the <max range value> is greater than or equal to 9 seconds. Otherwise, the default is the value specified by the <max range value>. The <min range value> can be a number between -3 – (.75 x <max range value>).

The <max range value> parameter specifies the maximum number of seconds allowed between sending unsolicited multicast router advertisements from the interface. This number can be between 4 – 1800 seconds and must be greater than the <min range value> x 1.33. The default is 600 seconds.

The **ipv6 nd ra-lifetime** <number> is a value between 0 – 9000 seconds. To restore the router lifetime value of 1800 seconds, use the **no** form of the command.

The **ipv6 nd ra-hop-limit** <number> is a value from 0 – 255. The default is 64.

NOTE

By default, router advertisements will always have the MTU option. To suppress the MTU option, use the following command at the Interface level of the CLI: **ipv6 nd suppress-mtu-option**.

Prefixes advertised in IPv6 router advertisement messages

By default, router advertisement messages include prefixes configured as addresses on router interfaces using the **ipv6 address** command. You can use the **ipv6 nd prefix-advertisement** command to control exactly which prefixes are included in router advertisement messages. Along with which prefixes the router advertisement messages contain, you can also specify the following parameters:

- **Valid lifetime**—(Mandatory) The time interval (in seconds) in which the specified prefix is advertised as valid. The default is 2592000 seconds (30 days). When the timer expires, the prefix is no longer considered to be valid.
- **Preferred lifetime**—(Mandatory) The time interval (in seconds) in which the specified prefix is advertised as preferred. The default is 604800 seconds (7 days). When the timer expires, the prefix is no longer considered to be preferred.
- **Onlink flag**—(Optional) If this flag is set, the specified prefix is assigned to the link upon which it is advertised. Nodes sending traffic to addresses that contain the specified prefix consider the destination to be reachable on the local link.
- **Autoconfiguration flag**—(Optional) If this flag is set, the stateless auto configuration feature can use the specified prefix in the automatic configuration of 128-bit IPv6 addresses for hosts on the local link, provided the specified prefix is aggregatable, as specified in RFC 2374.

For example, to advertise the prefix 2001:e077:a487:7365::/64 in router advertisement messages sent out on Ethernet interface 3/1 with a valid lifetime of 1000 seconds, a preferred lifetime of 800 seconds, and the Onlink and Autoconfig flags set, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 nd prefix-advertisement
2001:e077:a487:7365::/64 1000 800 onlink autoconfig
```

Syntax: **[no] ipv6 nd prefix-advertisement** <ipv6-prefix>/<prefix-length> <valid-lifetime> <preferred-lifetime> **[autoconfig] [onlink]**

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The valid lifetime and preferred lifetime is a numerical value between 0 – 4294967295 seconds. The default valid lifetime is 2592000 seconds (30 days), while the default preferred lifetime is 604800 seconds (7 days).

To remove a prefix from the router advertisement messages sent from a particular interface, use the **no** form of this command.

Setting flags in IPv6 router advertisement messages

An IPv6 router advertisement message can include the following flags:

- **Managed Address Configuration**—This flag indicates to hosts on a local link if they should use the stateful autoconfiguration feature to get IPv6 addresses for their interfaces. If the flag is set, the hosts use stateful autoconfiguration to get addresses as well as non-IPv6-address information. If the flag is not set, the hosts do not use stateful autoconfiguration to get addresses and if the hosts can get non-IPv6-address information from stateful autoconfiguration is determined by the setting of the Other Stateful Configuration flag.
- **Other Stateful Configuration**—This flag indicates to hosts on a local link if they can get non-IPv6 address autoconfiguration information. If the flag is set, the hosts can use stateful autoconfiguration to get non-IPv6-address information.

NOTE

When determining if hosts can use stateful autoconfiguration to get non-IPv6-address information, a set Managed Address Configuration flag overrides an unset Other Stateful Configuration flag. In this situation, the hosts can obtain nonaddress information. However, if the Managed Address Configuration flag is not set and the Other Stateful Configuration flag is set, then the setting of the Other Stateful Configuration flag is used.

By default, the Managed Address Configuration and Other Stateful Configuration flags are not set in router advertisement messages. For example, to set these flags in router advertisement messages sent from Ethernet interface 3/1, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 nd managed-config-flag
Brocade(config-if-e1000-3/1)#ipv6 nd other-config-flag
```

Syntax: [no] ipv6 nd managed-config-flag

Syntax: [no] ipv6 nd other-config-flag

To remove either flag from router advertisement messages sent on an interface, use the **no** form of the respective command.

Enabling and disabling IPv6 router advertisements

If IPv6 unicast routing is enabled on an Ethernet interface, by default, this interface sends IPv6 router advertisement messages. However, by default, non-LAN interface types, for example, tunnel interfaces, do not send router advertisement messages.

To disable the sending of router advertisement messages on an Ethernet interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 nd suppress-ra
```

To enable the sending of router advertisement messages on a tunnel interface, enter commands such as the following.

```
Brocade(config)#interface tunnel 1
Brocade(config-tnif-1)#no ipv6 nd suppress-ra
```

Syntax: [no] ipv6 nd suppress-ra

Configuring reachable time for remote IPv6 nodes

You can configure the duration (in seconds) that a router considers a remote IPv6 node reachable. By default, a router interface uses the value of 30 seconds.

The router advertisement messages sent by a router interface include the amount of time specified by the **ipv6 nd reachable-time** command so that nodes on a link use the same reachable time duration. By default, the messages include a default value of 0.

Brocade does not recommend configuring a short reachable time duration, because a short duration causes the IPv6 network devices to process the information at a greater frequency.

For example, to configure the reachable time of 40 seconds for Ethernet interface 3/1, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 nd reachable-time 40
```

Syntax: [no] **ipv6 nd reachable-time** <seconds>

For the <seconds> parameter, specify a number from 0 – 3600 seconds. To restore the default time, use the **no** form of this command.

NOTE

The actual reachable time will be from .5 to 1.5 times the configured or default value.

IPv6 MTU

The IPv6 maximum transmission unit (MTU) is the maximum length of an IPv6 packet that can be transmitted on a particular interface. If an IPv6 packet is longer than an MTU, the host that originated the packet fragments the packet and transmits its contents in multiple packets that are shorter than the configured MTU.

By default, in *non-jumbo mode*, the maximum Ethernet MTU size is 1500 bytes. When **jumbo** is enabled, the default maximum Ethernet MTU size is 10218.

Configuration notes and feature limitations for IPv6 MTU

- The IPv6 MTU command is applicable to VEs and physical IP interfaces. It applies to traffic routed between networks.
- You cannot use this command to set Layer 2 maximum frame sizes per interface. The global **jumbo** command causes all interfaces to accept Layer 2 frames.
- For non-jumbo mode, you can configure an IPv6 MTU greater than 1500 bytes, although the default remains at 1500 bytes. The value of the MTU you can define depends on the following:
 - For a physical port, the maximum value of the MTU is the equal to the maximum frame size of the port minus 18 (Layer 2 MAC header + CRC).
 - If the size of a jumbo packet received on a port is equal to the maximum frame size – 18 (Layer 2 MAC header + CRC) and if this value is greater than the outgoing port's IPv4/IPv6 MTU, then it will be forwarded in the CPU.

- For a virtual routing interface, the maximum value of the MTU is the maximum frame size configured for the VLAN to which it is associated, minus 18 (Layer 2 MAC header + CRC). If a maximum frame size for a VLAN is not configured, then configure the MTU based on the smallest maximum frame size of all the ports of the VLAN that corresponds to the virtual routing interface, minus 18 (Layer 2 MAC header + CRC).

Changing the IPv6 MTU

To define IPv6 maximum transmission unit (MTU) globally, enter the **ipv6 mtu** command at the Global CONFIG level of the CLI:

```
Brocade(config)#ipv6 mtu 1300
```

You can configure the IPv6 MTU on individual interfaces. For example, to configure the MTU on Ethernet interface 3/1 as 1280 bytes, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#ipv6 mtu 1280
```

Syntax: [no] **ipv6 mtu** <bytes>

For <bytes>, specify a value between 1280 – 1500, or 1280 – 10218 if **jumbo** mode is enabled. If a nondefault value is configured for an interface, router advertisements include an MTU option.

Static neighbor entries configuration

In some special cases, a neighbor cannot be reached using the neighbor discovery feature. In this situation, you can add a static entry to the IPv6 neighbor discovery cache, which causes a neighbor to be reachable at all times without using neighbor discovery. (A static entry in the IPv6 neighbor discovery cache functions like a static ARP entry in IPv4.)

NOTE

A port that has a statically assigned IPv6 entry cannot be added to a VLAN.

NOTE

Static neighbor configurations will be cleared on secondary ports when a trunk is formed.

For example, to add a static entry for a neighbor with the IPv6 address 3001::ffe0:2678:47b and link-layer address 0004.6a2b.8641 that is reachable through Ethernet interface 3/1, enter the **ipv6 neighbor** command.

```
Brocade(config)#ipv6 neighbor 3001::ffe0:2678:47b ethernet 3/1 0004.6a2b.8641
```

Syntax: [no] **ipv6 neighbor** <ipv6-address> **ethernet** <port> | **ve** <ve-number> [**ethernet** <port>] <link-layer-address>

The <ipv6-address> parameter specifies the address of the neighbor.

The **ethernet** | **ve** parameter specifies the interface through which to reach a neighbor. If you specify an Ethernet interface, specify the port number of the Ethernet interface. If you specify a VE, specify the VE number and then the Ethernet port numbers associated with the VE. The link-layer address is a 48-bit hardware address of the neighbor.

If you attempt to add an entry that already exists in the neighbor discovery cache, the software changes the already existing entry to a static entry.

To remove a static IPv6 entry from the IPv6 neighbor discovery cache, use the **no** form of this command.

Limiting the number of hops an IPv6 packet can traverse

By default, the maximum number of hops an IPv6 packet can traverse is 64. You can change this value to between 0 – 255 hops. For example, to change the maximum number of hops to 70, enter the following command.

```
Brocade(config)#ipv6 hop-limit 70
```

Syntax: [no] **ipv6 hop-limit** <number>

Use the **no** form of the command to restore the default value.

hop-limit 0 will transmit packets with default (64) hop limit.

<number> can be from 0 – 255.

IPv6 source routing security enhancements

The IPv6 specification (RFC 2460) specifies support for IPv6 source-routed packets using a type 0 Routing extension header, requiring device and host to process the type 0 routing extension header. However, this requirement may leave a network open to a DoS attack.

A security enhancement disables sending IPv6 source-routed packets to IPv6 devices. (This enhancement conforms to RFC 5095.)

By default, when the router drops a source-routed packet, it sends an ICMP Parameter Problem (type 4), Header Error (code 0) message to the packet's source address, pointing to the unrecognized routing type. To disable these ICMP error messages, enter the following command:

```
Brocade(config)# no ipv6 icmp source-route
```

Syntax: [no] **ipv6 icmp source-route**

Use the **ipv6 icmp source-route** form of the command to enable the ICMP error messages.

TCAM space on FCX device configuration

FCX devices store routing information for IPv4 and IPv6 and GRE tunnel information in the same TCAM table. You can configure the amount of TCAM space to allocate for IPv4 routing information and GRE tunnels. The remaining space is allocated automatically for IPv6 routing information.

FCX devices have TCAM space to store 16,000 IPv4 route entries. Each IPv6 route entry and GRE tunnel use as much storage space as four IPv4 route entries. The default, maximum, and minimum allocation values for each type of data are shown in [Table 65](#).

TABLE 65 TCAM space allocation on FCX devices

	Default	Maximum	Minimum
IPv4 route entries	12000	15168	4096
IPv6 route entries	908	2884	68
GRE tunnels	16	64	16

Allocating TCAM space for IPv4 routing information

For example, to allocate 13,512 IPv4 route entries, enter the following command:

```
Brocade(config)# system-max ip-route 13512
```

Syntax: `system-max ip-route <routes>`

The `<routes>` parameter specifies how many IPv4 route entries get allocated. The command output shows the new space allocations for IPv4 and IPv6. You must save the running configuration to the startup configuration and reload the device for the changes to take effect.

After the device reloads, the space allocated for IPv4 and IPv6 routing information appears in the device running configuration in this format:

```
system-max ip-route 13512
system-max ip6-route 514
```

NOTE

If you disable IPv6 routing, the TCAM space allocations do not change. If you want to allocate the maximum possible space for IPv4 routing information, you must configure the TCAM space manually.

Allocating TCAM space for GRE tunnel information

For example, to allocate space for 64 GRE tunnels, enter the following command at the Privileged EXEC level:

```
Brocade#system-max gre-tunnels 64
```

Syntax: `system-max gre-tunnels <tunnels>`

The `<tunnels>` parameter specifies the number of GRE tunnels to allocate.

Clearing global IPv6 information

You can clear the following global IPv6 information:

- Entries from the IPv6 cache.
- Entries from the IPv6 neighbor table.
- IPv6 routes from the IPv6 route table.
- IPv6 traffic statistics.

Clearing the IPv6 cache

You can remove all entries from the IPv6 cache or specify an entry based on the following:

- IPv6 prefix.
- IPv6 address.
- Interface type.

For example, to remove entries for IPv6 address 2000:e0ff::1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
Brocade#clear ipv6 cache 2000:e0ff::1
```

Syntax: `clear ipv6 cache [<ipv6-prefix>/<prefix-length> | <ipv6-address> | ethernet <port> | tunnel <number> | ve <number>]`

You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

You must specify the `<ipv6-address>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet | tunnel | ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number, respectively.

Clearing IPv6 neighbor information

You can remove all entries from the IPv6 neighbor table or specify an entry based on the following:

- IPv6 prefix
- IPv6 address
- Interface type

For example, to remove entries for Ethernet interface 3/1, enter the following command at the Privileged EXEC level or any of the CONFIG levels of the CLI.

```
Brocade#clear ipv6 neighbor ethernet 3/1
```

Syntax: `clear ipv6 neighbor [<ipv6-prefix>/<prefix-length> | <ipv6-address> | ethernet <port> | ve <number>]`

You must specify the *<ipv6-prefix>* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *<prefix-length>* parameter as a decimal value. A slash mark (/) must follow the *<ipv6-prefix>* parameter and precede the *<prefix-length>* parameter.

You must specify the *<ipv6-address>* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet | ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE, also specify the VE number.

Clearing IPv6 routes from the IPv6 route table

You can clear all IPv6 routes or only those routes associated with a particular IPv6 prefix from the IPv6 route table and reset the routes.

For example, to clear IPv6 routes associated with the prefix 2000:7838::/32, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
Brocade#clear ipv6 route 2000:7838::/32
```

Syntax: `clear ipv6 route [<ipv6-prefix>/<prefix-length>]`

The *<ipv6-prefix>/<prefix-length>* parameter clears routes associated with a particular IPv6 prefix. You must specify the *<ipv6-prefix>* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *<prefix-length>* parameter as a decimal value. A slash mark (/) must follow the *<ipv6-prefix>* parameter and precede the *<prefix-length>* parameter.

Clearing IPv6 traffic statistics

To clear all IPv6 traffic statistics (reset all fields to zero), enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
Brocade(config)#clear ipv6 traffic
```

Syntax: `clear ipv6 traffic`

Displaying global IPv6 information

You can display output for the following global IPv6 parameters:

- IPv6 cache
- IPv6 interfaces
- IPv6 neighbors
- IPv6 route table
- Local IPv6 routers
- IPv6 TCP connections and the status of individual connections
- IPv6 traffic statistics

Displaying IPv6 cache information

The IPv6 cache contains an IPv6 host table that has indices to the next hop gateway and the router interface on which the route was learned.

To display IPv6 cache information, enter the following command at any CLI level.

```
Brocade#show ipv6 cache
Total number of cache entries: 10
  IPv6 Address      Next Hop      Port
1  5000:2::2        LOCAL        tunnel 2
2  2000:4::106      LOCAL        ethe 3/2
3  2000:4::110      DIRECT       ethe 3/2
4  2002:c0a8:46a::1 LOCAL        ethe 3/2
5  5005::2e0:52ff:fe99:9737 LOCAL        ethe 3/2
6  5005::ffff:ffff:feff:ffff LOCAL        loopback 2
7  5005::c0a8:46a    LOCAL        tunnel 2
8  5005::c0a8:46a    LOCAL        tunnel 6
9  2999::1          LOCAL        loopback 2
10 5005::2e0:52ff:fe99:9700 LOCAL        ethe 3/1
```

Syntax: `show ipv6 cache [<index-number> | <ipv6-prefix>/<prefix-length> | <ipv6-address> | ethernet <port> | ve <number> | tunnel <number>]`

The `<index-number>` parameter restricts the display to the entry for the specified index number and subsequent entries.

The `<ipv6-prefix>/<prefix-length>` parameter restricts the display to the entries for the specified IPv6 prefix. You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

The **ethernet | ve | tunnel** parameter restricts the display to the entries for the specified interface. The `<ipv6-address>` parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number. If you specify a tunnel interface, also specify the tunnel number.

This display shows the following information.

TABLE 66 IPv6 cache information fields

Field	Description
Total number of cache entries	The number of entries in the cache table.
IPv6 Address	The host IPv6 address.
Next Hop	The next hop, which can be one of the following: <ul style="list-style-type: none"> Direct – The next hop is directly connected to the router. Local – The next hop is originated on this router. <code><ipv6 address></code> – The IPv6 address of the next hop.
Port	The port on which the entry was learned.

Displaying IPv6 interface information

To display IPv6 interface information, enter the following command at any CLI level.

```
Brocade#show ipv6 interface
Routing Protocols : R - RIP  O - OSPF
Interface          Status      Routing  Global Unicast Address
Ethernet 3/3        down/down  R
Ethernet 3/5        down/down
Ethernet 3/17       up/up      2017::c017:101/64
Ethernet 3/19       up/up      2019::c019:101/64
VE 4                down/down
VE 14               up/up      2024::c060:101/64
Loopback 1          up/up      ::1/128
Loopback 2          up/up      2005::303:303/128
Loopback 3          up/up
```

Syntax: `show ipv6 interface [<interface> [<port-number> | <number>]]`

The `<interface>` parameter displays detailed information for a specified interface. For the interface, you can specify the **Ethernet**, **loopback**, **tunnel**, or **VE** keywords. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the interface.

This display shows the following information.

TABLE 67 General IPv6 interface information fields

Field	Description
Routing protocols	A one-letter code that represents a routing protocol that can be enabled on an interface.
Interface	The interface type, and the port number or number of the interface.
Status	The status of the interface. The entry in the Status field will be either “up/up” or “down/down”.
Routing	The routing protocols enabled on the interface.
Global Unicast Address	The global unicast address of the interface.

To display detailed information for a specific interface, enter a command such as the following at any CLI level.

```

Brocade#show ipv6 interface ethernet 3/1
Interface Ethernet 3/1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::2e0:52ff:fe99:97
  Global unicast address(es):
  Joined group address(es):
    ff02::9
    ff02::1:ff99:9700
    ff02::2
    ff02::1
  MTU is 1500 bytes
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30 seconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1 seconds
  ND advertised retransmit interval is 0 seconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  No Inbound Access List Set
  No Outbound Access List Set
  RIP enabled

```

This display shows the following information.

TABLE 68 Detailed IPv6 interface information fields

Field	Description
Interface/line protocol status	The status of interface and line protocol. If you have disabled the interface with the disable command, the status will be “administratively down”. Otherwise, the status is either “up” or “down”.
IPv6 status/link-local address	The status of IPv6. The status is either “enabled” or “disabled”. Displays the link-local address, if one is configured for the interface.
Global unicast address(es)	Displays the global unicast address(es), if one or more are configured for the interface.
Joined group address(es)	The multicast address(es) that a router interface listens for and recognizes.
MTU	The setting of the maximum transmission unit (MTU) configured for the IPv6 interface. The MTU is the maximum length an IPv6 packet can have to be transmitted on the interface. If an IPv6 packet is longer than an MTU, the host that originated the packet fragments the packet and transmits its contents in multiple packets that are shorter than the configured MTU.
ICMP	The setting of the ICMP redirect parameter for the interface.
ND	The setting of the various neighbor discovery parameters for the interface.
Access List	The inbound and outbound access control lists applied to the interface.
Routing protocols	The routing protocols enabled on the interface.

Displaying IPv6 neighbor information

You can display the IPv6 neighbor table, which contains an entry for each IPv6 neighbor with which the router exchanges IPv6 packets.

To display the IPv6 neighbor table, enter the following command at any CLI level.

```
Brocade(config)#show ipv6 neighbor
Total number of Neighbor entries: 3
  IPv6 Address                               LinkLayer-Addr State Age Port  vlan
IsR 5555::55                                0002.0002.0002 *REACH0 e 3/11 - 0
2000:4::110                                  00e0.5291.bb37 REACH 20 e 3/1 5 1
fe80::2e0:52ff:fe91:bb37                     00e0.5291.bb37 DELAY 1 e 3/2 4 1
fe80::2e0:52ff:fe91:bb40                     00e0.5291.bb40 STALE 5930e 3/3 5 1
```

Syntax: `show ipv6 neighbor [<ipv6-prefix>/<prefix-length> | <ipv6-address> | <interface> [<port> | <number>]]`

The `<ipv6-prefix>/<prefix-length>` parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

The `<ipv6-address>` parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<interface>` parameter restricts the display to the entries for the specified router interface. For this parameter, you can specify the **Ethernet** or **VE** keywords. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

This display shows the following information.

TABLE 69 IPv6 neighbor information fields

Field	Description
Total number of neighbor entries	The total number of entries in the IPv6 neighbor table.
IPv6 Address	The 128-bit IPv6 address of the neighbor.
Link-Layer Address	The 48-bit interface ID of the neighbor.
State	<p>The current state of the neighbor. Possible states are as follows:</p> <ul style="list-style-type: none"> • INCOMPLETE – Address resolution of the entry is being performed. • *REACH – The static forward path to the neighbor is functioning properly. • REACH – The forward path to the neighbor is functioning properly. • STALE – This entry has remained unused for the maximum interval. While stale, no action takes place until a packet is sent. • DELAY – This entry has remained unused for the maximum interval, and a packet was sent before another interval elapsed. • PROBE – Neighbor solicitation are transmitted until a reachability confirmation is received.

TABLE 69 IPv6 neighbor information fields (Continued)

Field	Description
Age	The number of seconds the entry has remained unused. If this value remains unused for the number of seconds specified by the ipv6 nd reachable-time command (the default is 30 seconds), the entry is removed from the table.
Port	The physical port on which the entry was learned.
vlan	The VLAN on which the entry was learned.
IsR	Determines if the neighbor is a router or host: 0 – Indicates that the neighbor is a host. 1 – Indicates that the neighbor is a router.

Displaying the IPv6 route table

To display the IPv6 route table, enter the following command at any CLI level.

```
Brocade#show ipv6 route
IPv6 Routing Table - 7 entries:
Type Codes: C - Connected, S - Static, R - RIP, O - OSPF, B - BGP
OSPF Sub Type Codes: O - Intra, Oi - Inter, Ol - Type1 external, O2 - Type2
external
Type IPv6 Prefix                Next Hop Router                Interface  Dis/Metric
C  2000:4::/64                  ::                             ethe 3/2  0/0
S  2002::/16                    ::                             tunnel 6   1/1
S  2002:1234::/32               ::                             tunnel 6   1/1
C  2002:c0a8:46a::/64          ::                             ethe 3/2  0/0
C  2999::1/128                  ::                             loopback 2 0/0
O  2999::2/128                  fe80::2e0:52ff:fe91:bb37      ethe 3/2  110/1
C  5000:2::/64                  ::                             tunnel 2   0/0
```

Syntax: `show ipv6 route [<ipv6-address> | <ipv6-prefix>/<prefix-length> | bgp | connect | ospf | rip | static | summary]`

The `<ipv6-address>` parameter restricts the display to the entries for the specified IPv6 address. You must specify the `<ipv6-address>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<ipv6-prefix>/<prefix-length>` parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

The **bgp** keyword restricts the display to entries for BGP4 routes.

The **connect** keyword restricts the display to entries for directly connected interface IPv6 routes.

The **ospf** keyword restricts the display to entries for OSPFv3 routes.

The **rip** keyword restricts the display to entries for RIPng routes.

The **static** keyword restricts the display to entries for static IPv6 routes.

The **summary** keyword displays a summary of the prefixes and different route types.

The following table lists the information displayed by the **show ipv6 route** command.

TABLE 70 IPv6 route table fields

Field	Description
Number of entries	The number of entries in the IPv6 route table.
Type	The route type, which can be one of the following: <ul style="list-style-type: none"> • C – The destination is directly connected to the router. • S – The route is a static route. • R – The route is learned from RIPng. • O – The route is learned from OSPFv3. • B – The route is learned from BGP4.
IPv6 Prefix	The destination network of the route.
Next-Hop Router	The next-hop router.
Interface	The interface through which this router sends packets to reach the route's destination.
Dis/Metric	The route's administrative distance and metric value.

To display a summary of the IPv6 route table, enter the following command at any CLI level.

```
Brocade#show ipv6 route summary
IPv6 Routing Table - 7 entries:
  4 connected, 2 static, 0 RIP, 1 OSPF, 0 BGP
Number of prefixes:
 /16: 1 /32: 1 /64: 3 /128: 2
```

The following table lists the information displayed by the **show ipv6 route summary** command.

TABLE 71 IPv6 route table summary fields

Field	Description
Number of entries	The number of entries in the IPv6 route table.
Number of route types	The number of entries for each route type.
Number of prefixes	A summary of prefixes in the IPv6 route table, sorted by prefix length.

Displaying local IPv6 routers

The Brocade device can function as an IPv6 host, instead of an IPv6 router, if you configure IPv6 addresses on its interfaces but do not enable IPv6 routing using the **ipv6 unicast-routing** command.

From the IPv6 host, you can display information about IPv6 routers to which the host is connected. The host learns about the routers through their router advertisement messages. To display information about the IPv6 routers connected to an IPv6 host, enter the following command at any CLI level.

```
Brocade#show ipv6 router
Router fe80::2e0:80ff:fe46:3431 on Ethernet 50, last update 0 min
Hops 64, Lifetime 1800 sec
Reachable time 0 msec, Retransmit time 0 msec
```


Syntax: show ipv6 router

If you configure your Brocade device to function as an IPv6 router (you configure IPv6 addresses on its interfaces and enable IPv6 routing using the **ipv6 unicast-routing** command) and you enter the **show ipv6 router** command, you will receive the following output.

```
No IPv6 router in table
```

Meaningful output for this command is generated for Brocade devices configured to function as IPv6 hosts only.

This display shows the following information.

TABLE 72 IPv6 local router information fields

Field	Description
Router <ipv6 address> on <interface> <port>	The IPv6 address for a particular router interface.
Last update	The amount of elapsed time (in minutes) between the current and previous updates received from a router.
Hops	The default value that should be included in the Hop Count field of the IPv6 header for outgoing IPv6 packets. The hops value applies to the router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.
Lifetime	The amount of time (in seconds) that the router is useful as the default router.
Reachable time	The amount of time (in milliseconds) that a router assumes a neighbor is reachable after receiving a reachability confirmation. The reachable time value applies to the router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.
Retransmit time	The amount of time (in milliseconds) between retransmissions of neighbor solicitation messages. The retransmit time value applies to the router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.

Displaying IPv6 TCP information

You can display the following IPv6 TCP information:

- General information about each TCP connection on the router, including the percentage of free memory for each of the internal TCP buffers.
- Detailed information about a specified TCP connection.

To display general information about each TCP connection on the router, enter the following command at any CLI level.

Displaying global IPv6 information

```
Brocade#show ipv6 tcp connections
Local IP address:port <-> Remote IP address:port TCP state
192.168.182.110:23 <-> 192.168.8.186:4933 ESTABLISHED
192.168.182.110:8218 <-> 192.168.182.106:179 ESTABLISHED
192.168.182.110:8039 <-> 192.168.2.119:179 SYN-SENT
192.168.182.110:8159 <-> 192.168.2.102:179 SYN-SENT
2000:4::110:179 <-> 2000:4::106:8222 ESTABLISHED (1440)
Total 5 TCP connections

TCP MEMORY USAGE PERCENTAGE
FREE TCP = 98 percent
FREE TCP QUEUE BUFFER = 99 percent
FREE TCP SEND BUFFER = 97 percent
FREE TCP RECEIVE BUFFER = 100 percent
FREE TCP OUT OF SEQUENCE BUFFER = 100 percent
```

Syntax: show ipv6 tcp connections

This display shows the following information.

TABLE 73 General IPv6 TCP connection fields

Field	Description
Local IP address:port	The IPv4 or IPv6 address and port number of the local router interface over which the TCP connection occurs.
Remote IP address:port	The IPv4 or IPv6 address and port number of the remote router interface over which the TCP connection occurs.
TCP state	<p>The state of the TCP connection. Possible states include the following:</p> <ul style="list-style-type: none">• LISTEN – Waiting for a connection request.• SYN-SENT – Waiting for a matching connection request after having sent a connection request.• SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request.• ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection.• FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.• FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP.• CLOSE-WAIT – Waiting for a connection termination request from the local user.• CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP.• LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).• TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.• CLOSED – There is no connection state.
FREE TCP = <percentage>	The percentage of free TCP control block (TCP) space.

TABLE 73 General IPv6 TCP connection fields (Continued)

Field	Description
FREE TCP QUEUE BUFFER = <percentage>	The percentage of free TCP queue buffer space.
FREE TCP SEND BUFFER = <percentage>	The percentage of free TCP send buffer space.
FREE TCP RECEIVE BUFFER = <percentage>	The percentage of free TCP receive buffer space.
FREE TCP OUT OF SEQUENCE BUFFER = <percentage>	The percentage of free TCP out of sequence buffer space.

To display detailed information about a specified TCP connection, enter a command such as the following at any CLI level.

```
Brocade#show ipv6 tcp status 2000:4::110 179 2000:4::106 8222
TCP: TCP = 0x217fc300
TCP: 2000:4::110:179 <-> 2000:4::106:8222: state: ESTABLISHED Port: 1
  Send: initial sequence number = 242365900
  Send: first unacknowledged sequence number = 242434080
  Send: current send pointer = 242434080
  Send: next sequence number to send = 242434080
  Send: remote received window = 16384
  Send: total unacknowledged sequence number = 0
  Send: total used buffers 0
  Receive: initial incoming sequence number = 740437769
  Receive: expected incoming sequence number = 740507227
  Receive: received window = 16384
  Receive: bytes in receive queue = 0
  Receive: congestion window = 1459
```

Syntax: `show ipv6 tcp status <local-ip-address> <local-port-number> <remote-ip-address>
<remote-port-number>`

The <local-ip-address> parameter can be the IPv4 or IPv6 address of the local interface over which the TCP connection is taking place.

The <local-port-number> parameter is the local port number over which a TCP connection is taking place.

The <remote-ip-address> parameter can be the IPv4 or IPv6 address of the remote interface over which the TCP connection is taking place.

The <remote-port-number> parameter is the local port number over which a TCP connection is taking place.

This display shows the following information.

TABLE 74 Specific IPv6 TCP connection fields

Field	Description
TCP = <i><location></i>	The location of the TCP.
<i><local-ip-address> <local-port-number> <remote-ip-address> <remote-port-number> <state> <port></i>	This field provides a general summary of the following: <ul style="list-style-type: none"> • The local IPv4 or IPv6 address and port number. • The remote IPv4 or IPv6 address and port number. • The state of the TCP connection. For information on possible states, refer to Table 73 on page 404. • The port numbers of the local interface.
Send: initial sequence number = <i><number></i>	The initial sequence number sent by the local router.
Send: first unacknowledged sequence number = <i><number></i>	The first unacknowledged sequence number sent by the local router.
Send: current send pointer = <i><number></i>	The current send pointer.
Send: next sequence number to send = <i><number></i>	The next sequence number sent by the local router.
Send: remote received window = <i><number></i>	The size of the remote received window.
Send: total unacknowledged sequence number = <i><number></i>	The total number of unacknowledged sequence numbers sent by the local router.
Send: total used buffers <i><number></i>	The total number of buffers used by the local router in setting up the TCP connection.
Receive: initial incoming sequence number = <i><number></i>	The initial incoming sequence number received by the local router.
Receive: expected incoming sequence number = <i><number></i>	The incoming sequence number expected by the local router.
Receive: received window = <i><number></i>	The size of the local router's receive window.
Receive: bytes in receive queue = <i><number></i>	The number of bytes in the local router's receive queue.
Receive: congestion window = <i><number></i>	The size of the local router's receive congestion window.

Displaying IPv6 traffic statistics

To display IPv6 traffic statistics, enter the following command at any CLI level.

```
Brocade#show ipv6 traffic
IP6 Statistics
    36947 received, 66818 sent, 0 forwarded, 36867 delivered, 0 rawout
    0 bad vers, 23 bad scope, 0 bad options, 0 too many hdr
    0 no route, 0 can not forward, 0 redirect sent
    0 frag recv, 0 frag dropped, 0 frag timeout, 0 frag overflow
    0 reassembled, 0 fragmented, 0 ofragments, 0 can not frag
    0 too short, 0 too small, 11 not member
    0 no buffer, 66819 allocated, 21769 freed
    0 forward cache hit, 46 forward cache miss

ICMP6 Statistics
Received:
    0 dest unreachable, 0 pkt too big, 0 time exceeded, 0 param prob
    2 echo req, 1 echo reply, 0 mem query, 0 mem report, 0 mem red
    0 router soli, 2393 router adv, 106 nei soli, 3700 nei adv, 0 redirect
    0 bad code, 0 too short, 0 bad checksum, 0 bad len
    0 reflect, 0 nd toomany opt, 0 badhopcount
Sent:
    0 dest unreachable, 0 pkt too big, 0 time exceeded, 0 param prob
    1 echo req, 2 echo reply, 0 mem query, 0 mem report, 0 mem red
    0 router soli, 2423 router adv, 3754 nei soli, 102 nei adv, 0 redirect
    0 error, 0 can not send error, 0 too freq
Sent Errors:
    0 unreachable no route, 0 admin, 0 beyond scope, 0 address, 0 no port
    0 pkt too big, 0 time exceed transit, 0 time exceed reassembly
    0 param problem header, 0 nexthheader, 0 option, 0 redirect, 0 unknown

UDP Statistics
    470 received, 7851 sent, 6 no port, 0 input errors

TCP Statistics
    57913 active opens, 0 passive opens, 57882 failed attempts
    159 active resets, 0 passive resets, 0 input errors
    565189 in segments, 618152 out segments, 171337 retransmission
```

Syntax: show ipv6 traffic

This **show ipv6 traffic** command displays the following information.

Field	Description
IPv6 statistics	
received	The total number of IPv6 packets received by the router.
sent	The total number of IPv6 packets originated and sent by the router.
forwarded	The total number of IPv6 packets received by the router and forwarded to other routers.
delivered	The total number of IPv6 packets delivered to the upper layer protocol.
rawout	This information is used by Brocade Technical Support.
bad vers	The number of IPv6 packets dropped by the router because the version number is not 6.

Field	Description (Continued)
bad scope	The number of IPv6 packets dropped by the router because of a bad address scope.
bad options	The number of IPv6 packets dropped by the router because of bad options.
too many hdr	The number of IPv6 packets dropped by the router because the packets had too many headers.
no route	The number of IPv6 packets dropped by the router because there was no route.
can not forward	The number of IPv6 packets the router could not forward to another router.
redirect sent	This information is used by Brocade Technical Support.
frag rcv	The number of fragments received by the router.
frag dropped	The number of fragments dropped by the router.
frag timeout	The number of fragment timeouts that occurred.
frag overflow	The number of fragment overflows that occurred.
reassembled	The number of fragmented IPv6 packets that the router reassembled.
fragmented	The number of IPv6 packets fragmented by the router to accommodate the MTU of this router or of another device.
ofragments	The number of output fragments generated by the router.
can not frag	The number of IPv6 packets the router could not fragment.
too short	The number of IPv6 packets dropped because they are too short.
too small	The number of IPv6 packets dropped because they do not have enough data.
not member	The number of IPv6 packets dropped because the recipient is not a member of a multicast group.
no buffer	The number of IPv6 packets dropped because there is no buffer available.
forward cache miss	The number of IPv6 packets received for which there is no corresponding cache entry.
ICMP6 statistics	
Some ICMP statistics apply to both Received and Sent, some apply to Received only, some apply to Sent only, and some apply to Sent Errors only.	
Applies to received and sent	
dest unreachable	The number of Destination Unreachable messages sent or received by the router.
pkt too big	The number of Packet Too Big messages sent or received by the router.
time exceeded	The number of Time Exceeded messages sent or received by the router.
param prob	The number of Parameter Problem messages sent or received by the router.
echo req	The number of Echo Request messages sent or received by the router.
echo reply	The number of Echo Reply messages sent or received by the router.
mem query	The number of Group Membership Query messages sent or received by the router.
mem report	The number of Membership Report messages sent or received by the router.
mem red	The number of Membership Reduction messages sent or received by the router.
router soli	The number of Router Solicitation messages sent or received by the router.

Field	Description (Continued)
router adv	The number of Router Advertisement messages sent or received by the router.
nei soli	The number of Neighbor Solicitation messages sent or received by the router.
nei adv	The number of Router Advertisement messages sent or received by the router.
redirect	The number of redirect messages sent or received by the router.
Applies to received only	
bad code	The number of Bad Code messages received by the router.
too short	The number of Too Short messages received by the router.
bad checksum	The number of Bad Checksum messages received by the router.
bad len	The number of Bad Length messages received by the router.
nd toomany opt	The number of Neighbor Discovery Too Many Options messages received by the router.
badhopcount	The number of Bad Hop Count messages received by the router.
Applies to sent only	
error	The number of Error messages sent by the router.
can not send error	The number of times the node encountered errors in ICMP error messages.
too freq	The number of times the node has exceeded the frequency of sending error messages.
Applies to sent errors only	
unreach no route	The number of Unreachable No Route errors sent by the router.
admin	The number of Admin errors sent by the router.
beyond scope	The number of Beyond Scope errors sent by the router.
address	The number of Address errors sent by the router.
no port	The number of No Port errors sent by the router.
pkt too big	The number of Packet Too Big errors sent by the router.
time exceed transit	The number of Time Exceed Transit errors sent by the router.
time exceed reassembly	The number of Time Exceed Reassembly errors sent by the router.
param problem header	The number of Parameter Problem Header errors sent by the router.
nextheader	The number of Next Header errors sent by the router.
option	The number of Option errors sent by the router.
redirect	The number of Redirect errors sent by the router.
unknown	The number of Unknown errors sent by the router.
UDP statistics	
received	The number of UDP packets received by the router.
sent	The number of UDP packets sent by the router.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Brocade Technical Support.
TCP statistics	

Field	Description (Continued)
active opens	The number of TCP connections opened by the router by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by the router in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Brocade Technical Support.
active resets	The number of TCP connections the router reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections the router reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Brocade Technical Support.
in segments	The number of TCP segments received by the router.
out segments	The number of TCP segments sent by the router.
retransmission	The number of segments that the router retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

FWS Series Switch IPv6 management

[Table 75](#) lists the individual Brocade FastIron switches and the IPv6 management features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 75 Supported IPv6 management features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX
Link-Local IPv6 address	For information about IPv6 management features on FastIron X Series switches, refer to Chapter 8, “IPv6 Configuration on FastIron X Series, FCX, and ICX Series Switches” .	Yes	For information about IPv6 management features on Brocade FCX Series switches, refer to Chapter 8, “IPv6 Configuration on FastIron X Series, FCX, and ICX Series Switches”	For information about IPv6 management features on ICX switches, refer to Chapter 8, “IPv6 Configuration on FastIron X Series, FCX, and ICX Series Switches”
IPv6 copy ¹		Yes		
IPv6 ncopy ¹		Yes		
IPv6 debug		Yes		
IPv6 access-list (management ACLs)		Yes		
IPv6 ping		Yes		
IPv6 traceroute		Yes		
DNS server name resolution		Yes		
HTTP/HTTPS		Yes		
Logging (Syslog)		Yes		
RADIUS ¹		Yes		
SCP		Yes		
SSH		Yes		
SNMP		Yes		
SNMP traps		Yes		
SNTP		Yes		
Telnet		Yes		
TFTP ¹		Yes		

¹The following IPv6 management features, listed in [Table 75](#), are documented in other chapters of this guide:

- IPv6 copy – [“Using the IPv6 copy command”](#) on page 87
- IPv6 ncopy – [“IPv6 ncopy command”](#) on page 89
- RADIUS – [“Setting RADIUS over IPv6”](#) on page 167
- TFTP – [“Loading and saving configuration files with IPv6”](#) on page 87

This chapter describes the IPv6 management features, including command syntax and management examples.

IPv6 management overview

IPv6 was designed to replace IPv4, the Internet protocol that is most commonly used currently throughout the world. IPv6 increases the number of network address bits from 32 (IPv4) to 128, which provides more than enough unique IP addresses to support all of the network devices on the planet into the future. IPv6 is expected to quickly become the network standard.

Brocade FastIron devices that support IPv6 may be used as management hosts. Interfaces on these devices are configured with IPv6 addresses, but do not have full IPv6 routing enabled. IPv6 is available on all FastIron devices that are running Layer 2, base Layer 3, or full Layer 3 software images.

NOTE

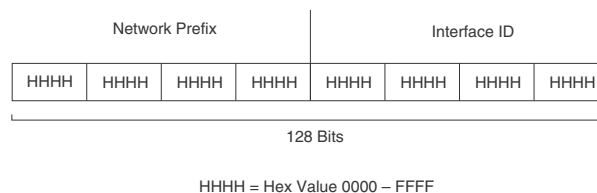
Brocade FastIron devices can serve as management hosts on an IPv6 network. However, **IPv6 routing** functionality is not supported for these devices.

IPv6 addressing

IPv4 is limited because of the 32-bit addressing format, which cannot satisfy potential increases in the number of users, geographical needs, and emerging applications. To address this limitation, IPv6 introduces a new 128-bit addressing format.

An IPv6 address is composed of 8 fields of 16-bit hexadecimal values separated by colons (:). [Figure 33](#) shows the IPv6 address format.

FIGURE 33 IPv6 address format



As shown in [Figure 33](#), HHHH is a 16-bit hexadecimal value, while H is a 4-bit hexadecimal value. The following is an example of an IPv6 address.

2001:0000:0000:0200:002D:D0FF:FE48:4672

Note that this IPv6 address includes hexadecimal fields of zeros. To make the address less cumbersome, you can do the following:

- Omit the leading zeros; for example, 2001:0:0:200:2D:D0FF:FE48:4672.
- Compress the successive groups of zeros at the beginning, middle, or end of an IPv6 address to two colons (::) once per address; for example, 2001::200:2D:D0FF:FE48:4672.

When specifying an IPv6 address in a command syntax, keep the following in mind:

- You can use the two colons (::) only once in the address to represent the longest successive hexadecimal fields of zeros
- The hexadecimal letters in IPv6 addresses are not case-sensitive

As shown in [Figure 33](#), the IPv6 network prefix is composed of the left-most bits of the address. As with an IPv4 address, you can specify the IPv6 prefix using the `<prefix>/<prefix-length>` format, where the following applies.

The `<prefix>` parameter is specified as 16-bit hexadecimal values separated by a colon.

The `<prefix-length>` parameter is specified as a decimal value that indicates the left-most bits of the IPv6 address.

The following is an example of an IPv6 prefix.

```
2001:FF08:49EA:D088::/64
```

Enabling and disabling IPv6

IPv6 is enabled by default for Brocade devices that support it. If desired, you can disable IPv6 on a global basis on an device by entering the following command at the Global CONFIG level of the CLI.

```
Brocade(config)#no ipv6 enable
```

Syntax: `no ipv6 enable`

To re-enable IPv6 after it has been disabled, enter the `ipv6 enable` command.

IPv6 management features

This section describes the CLI management commands that are available to FastIron devices that support IPv6.

IPv6 management ACLs

When you enter the `ipv6 access-list` command, the Brocade device enters the IPv6 Access List configuration level, where you can access several commands for configuring IPv6 ACL entries. After configuring the ACL entries, you can apply them to network management access features such as Telnet, SSH, Web, and SNMP.

NOTE

Unlike IPv4, there is no distinction between standard and extended ACLs in IPv6.

Example

```
Brocade(config)#ipv6 access-list netw
Brocade(config-ipv6-access-list-netw)#
```

Syntax: `[no] ipv6 access-list <ACL name>`

The `<ACL name>` variable specifies a name for the IPv6 ACL. An IPv6 ACL name cannot start with a numeral, for example, `1access`. Also, an IPv4 ACL and an IPv6 ACL cannot share the same name.

IPv6 Web management using HTTP and HTTPS

When you have an IPv6 management station connected to a switch with an IPv6 address applied to the management port, you can manage the switch from a Web browser by entering `http://[<ipv6 address>]` or

`https://[<ipv6 address>]` in the browser address field.

NOTE

You must enclose the IPv6 address with square brackets [] in order for the Web browser to work.

Restricting web access

You can restrict Web management access to include only management functions on a Brocade device that is acting as an IPv6 host, or restrict access so that the Brocade host can be reached by a specified IPv6 device.

Restricting Web management access by specifying an IPv6 ACL

You can specify an IPv6 ACL that restricts Web management access to management functions on the device that is acting as the IPv6 host.

Example

```
Brocade(config)#access-list 12 deny host 2000:2383:e0bb::2/128 log
Brocade(config)#access-list 12 deny 30ff:3782::ff89/128 log
Brocade(config)#access-list 12 deny 3000:4828::fe19/128 log
Brocade(config)#access-list 12 permit any
Brocade(config)#web access-group ipv6 12
```

Syntax: `web access-group ipv6 <ipv6 ACL name>`

where `<ipv6 ACL name>` is a valid IPv6 ACL.

Restricting Web management access to an IPv6 host

You can specify a single device with an IPv6 address to have Web management access to the host device. No other device except the one with the specified IPv6 address can access the Web Management Interface.

Example

```
Brocade(config)#web client ipv6 3000:2383:e0bb::2/128
```

Syntax: `web client ipv6 <ipv6-address>`

The `<ipv6-address>` you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

IPv6 logging

This feature allows you to specify an IPv6 server as the Syslog server.

Specifying an IPv6 Syslog server

To specify an IPv6 Syslog server, enter the log host ipv6 command as shown below.

Example

```
Brocade(config)#log host ipv6 2000:2383:e0bb::4/128
```

Syntax: `[no] log host ipv6 <ipv6-address> [<udp-port-num>]`

The `<ipv6-address>` must be in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The `<udp-port-num>` optional parameter specifies the UDP application port used for the Syslog facility.

Name-to-IPv6 address resolution using IPv6 DNS server

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a Brocade device and thereby recognize all hosts within that domain. After you define a domain name, the Brocade device automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain “newyork.com” is defined on a Brocade device, and you want to initiate a ping to host “NYC01” on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping.

```
Brocade#ping nyc01
Brocade#ping nyc01.newyork.com
```

Defining an IPv6 DNS entry

IPv6 defines new DNS record types to resolve queries for domain names to IPv6 addresses, as well as IPv6 addresses to domain names. Brocade devices running IPv6 software support AAAA DNS records, which are defined in RFC 1886.

AAAA DNS records are analogous to the A DNS records used with IPv4. They store a complete IPv6 address in each record. AAAA records have a type value of 28.

To establish an IPv6 DNS entry for the device, enter the following command.

```
Brocade(config)#ipv6 dns domain-name companynet.com
```

Syntax: `[no] ipv6 dns domain-name <domain name>`

To define an IPv6 DNS server address, enter the following command.

```
Brocade(config)#ipv6 dns server-address 200::1
```

Syntax: `[no] ipv6 dns server-address <ipv6-addr> [<ipv6-addr>] [<ipv6-addr>] [<ipv6-addr>]`

As an example, in a configuration where **ftp6.companynet.com** is a server with an IPv6 protocol stack, when a user pings ftp6.companynet.com, the Brocade device attempts to resolve the AAAA DNS record. In addition, if the DNS server does not have an IPv6 address, as long as it is able to resolve AAAA records, it can still respond to DNS queries.

Pinging IPv6

The **ping** command allows you to verify the connectivity from a Brocade device to an IPv6 device by performing an ICMP for IPv6 echo test.

For example, to ping a device with the IPv6 address of 2001:3424:847f:a385:34dd::45 from the Brocade device, enter the **ping ipv6** command:

```
Brocade#ping ipv6 2001:3424:847f:a385:34dd::45
```

Syntax: `ping ipv6 <ipv6-address> [outgoing-interface [<port> | ve <number>]] [source <ipv6-address>] [count <number>] [timeout <milliseconds>] [ttl <number>] [size`

<bytes>] [quiet] [numeric] [no-fragment] [verify]
[data <1-to-4 byte hex>] [brief]

- The **<ipv6-address>** parameter specifies the address of the router. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.
- The **outgoing-interface** keyword specifies a physical interface over which you can verify connectivity. If you specify a physical interface, such as an Ethernet interface, you must also specify the port number of the interface. If you specify a virtual interface, such as a VE, you must specify the number associated with the VE.
- The **source <ipv6-address>** parameter specifies an IPv6 address to be used as the origin of the ping packets.

NOTE

The **outgoing-interface** and **source** options are available only on Layer 3 code and not on Layer 2 code.

- The **count <number>** parameter specifies how many ping packets the router sends. You can specify from 1 - 4294967296. The default is 1.
- The **timeout <milliseconds>** parameter specifies how many milliseconds the router waits for a reply from the pinged device. You can specify a timeout from 1 - 4294967294 milliseconds. The default is 5000 (5 seconds).
- The **ttl <number>** parameter specifies the maximum number of hops. You can specify a TTL from 1 - 255. The default is 64.
- The **size <bytes>** parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 - 10173. The default is 16.
- The **no-fragment** keyword turns on the "do not fragment" bit in the IPv6 header of the ping packet. This option is disabled by default.
- The **quiet** keyword hides informational messages such as a summary of the ping parameters sent to the device, and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.
- The **verify** keyword verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.
- The **data <1 - 4 byte hex>** parameter lets you specify a specific data pattern for the payload instead of the default data pattern, "abcd", in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE

For parameters that require a numeric value, the CLI does not check whether the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

- The **brief** keyword causes ping test characters to be displayed. The following ping test characters are supported.

! Indicates that a reply was received.

. Indicates that the network server timed out while waiting for a reply.

U Indicates that a destination unreachable error PDU was received.

I Indicates that the user interrupted ping.

NOTE

On 48GC modules in non-jumbo mode, the maximum size of ping packets is 1486 bytes and the maximum frame size of tagged traffic is no larger than 1581 bytes.

SNTP over IPv6

To enable the Brocade device to send SNTP packets over IPv6, enter the **sntp server ipv6** command at the Global CONFIG level of the CLI.

```
Brocade(config)#sntp server ipv6 3000::400
```

Syntax: **sntp server ipv6** <ipv6-address>

The <ipv6-address> is the IPv6 address of the SNTP server. When you enter the IPv6 address, you do not need to specify the prefix length. A prefix length of 128 is implied.

SNMP3 over IPv6

Brocade FastIron devices support IPv6 for SNMP version 3. For more information about how to configure SNMP, refer to [Chapter 10, “SNMP Access”](#).

Specifying an IPv6 SNMP trap receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter a command such as the following.

```
Brocade(config)#snmp-server host ipv6 2001:efff:89::13
```

Syntax: **snmp-server host ipv6** <ipv6-address>

The <ipv6-address> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Secure Shell, SCP, and IPv6

Secure Shell (SSH) is a mechanism that allows secure remote access to management functions on the Brocade device. SSH provides a function similar to Telnet. You can log in to and configure the Brocade device using a publicly or commercially available SSH client program, just as you can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the Brocade device.

To open an SSH session between an IPv6 host running an SSH client program and the Brocade device, open the SSH client program and specify the IPv6 address of the device. For more information about configuring SSH on the Brocade device, refer to [“SSH2 and SCP”](#) on page 179.

IPv6 Telnet

Telnet sessions can be established between a Brocade device to a remote IPv6 host, and from a remote IPv6 host to the Brocade device using IPv6 addresses.

The **telnet** command establishes a Telnet connection from a Brocade device to a remote IPv6 host using the console. Up to five **read-access** Telnet sessions are supported on the router at one time. **Write-access** through Telnet is limited to one session, and only one outgoing Telnet session is supported on the router at one time. To see the number of open Telnet sessions at any time, enter the **show telnet** command.

Example

To establish a Telnet connection to a remote host with the IPv6 address of 3001:2837:3de2:c37::6, enter the following command.

```
Brocade#telnet 3001:2837:3de2:c37::6
```

Syntax: **telnet** <ipv6-address> [<port-number> | **outgoing-interface ethernet** <port> | **ve** <number>]

The <ipv6-address> parameter specifies the address of a remote host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <port-number> parameter specifies the port number on which the Brocade device establishes the Telnet connection. You can specify a value between 1 - 65535. If you do not specify a port number, the Brocade device establishes the Telnet connection on port 23.

If the IPv6 address you specify is a link-local address, you must specify the **outgoing-interface ethernet <port> | ve <number>** parameter. This parameter identifies the interface that must be used to reach the remote host. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

Establishing a Telnet session from an IPv6 host

To establish a Telnet session from an IPv6 host to the Brocade device, open your Telnet application and specify the IPv6 address of the Layer 3 Switch.

IPv6 traceroute

The **traceroute** command allows you to trace a path from the Brocade device to an IPv6 host.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a minimum TTL of 1 second and a maximum TTL of 30 seconds. In addition, if there are multiple equal-cost routes to the destination, the Brocade device displays up to three responses.

For example, to trace the path from the Brocade device to a host with an IPv6 address of 3301:23dd:349e:a384::34, enter the **traceroute ipv6** command.

```
Brocade#traceroute ipv6 3301:23dd:349e:a384::34
```

Syntax: **traceroute ipv6** <ipv6-address>

The <ipv6-address> parameter specifies the address of a host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

IPv6 management commands

The following management CLI commands are available in FastIron devices that support IPv6:

- show ipv6 traffic
- clear ipv6 traffic
- show ipv6 TCP
- show ipv6 access-list
- show ipv6 neighbor
- clear ipv6 neighbor

SNMP Access

[Table 76](#) lists individual Brocade switches and the SNMP access methods they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 76 Supported SNMP access features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
SNMP v1, v2, v3	Yes	Yes	Yes	Yes	Yes
Community strings	Yes	Yes	Yes	Yes	Yes
User-based security model for SNMP v3	Yes	Yes	Yes	Yes	Yes
SNMP v3 traps	Yes	Yes	Yes	Yes	Yes
Defining the UDP port for SNMP v3 traps	Yes	Yes	Yes	Yes	Yes
SNMP v3 over IPv6	Yes	Yes	Yes	Yes	Yes
AES encryption for SNMP v3	Yes	Yes	Yes	Yes	Yes

SNMP overview

SNMP is a set of protocols for managing complex networks. SNMP sends messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

[Chapter 4, “Security Access”](#) introduced a few methods used to secure SNMP access. They included the following:

- [“Using ACLs to restrict SNMP access”](#) on page 114
- [“Restricting SNMP access to a specific IP address”](#) on page 116
- [“Restricting SNMP access to a specific VLAN”](#) on page 119
- [“Disabling SNMP access”](#) on page 124

This chapter presents additional methods for securing SNMP access to Brocade devices. It contains the following sections:

- [“SNMP community strings”](#)
- [“User-based security model”](#)
- [“SNMP v3 configuration examples”](#)
- [“SNMP version 3 traps”](#)
- [“Displaying SNMP Information”](#)
- [“SNMP v3 configuration examples”](#)

Restricting SNMP access using ACL, VLAN, or a specific IP address constitute the first level of defense when the packet arrives at a Brocade device. The next level uses one of the following methods:

- Community string match In SNMP versions 1 and 2
- User-based model in SNMP version 3

SNMP views are incorporated in community strings and the user-based model.

SNMP community strings

SNMP versions 1 and 2 use community strings to restrict SNMP access. The default passwords for Web management access are the SNMP community strings configured on the device:

- The default read-only community string is “public”. To open a read-only Web management session, enter “get” and “public” for the user name and password.
- There is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web Management Interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure as the password.

You can configure as many additional read-only and read-write community strings as you need. The number of strings you can configure depends on the memory on the device. There is no practical limit.

The Web Management Interface supports only one read-write session at a time. When a read-write session is open on the Web Management Interface, subsequent sessions are read-only, even if the session login is “set” with a valid read-write password.

NOTE

If you delete the startup-config file, the device automatically re-adds the default “public” read-only community string the next time you load the software.

NOTE

As an alternative to the SNMP community strings, you can secure Web management access using local user accounts or ACLs. Refer to [“Local user accounts”](#) on page 129 or [“Using an ACL to restrict Web management access”](#) on page 114.

Encryption of SNMP community strings

The software automatically encrypts SNMP community strings. Users with read-only access or who do not have access to management functions in the CLI cannot display the strings. For users with read-write access, the strings are encrypted in the CLI but are shown in the clear in the Web Management Interface.

Encryption is enabled by default. You can disable encryption for individual strings or trap receivers if desired. Refer to the next section for information about encryption.

Adding an SNMP community string

The default SNMP community name (string) on a device is “public” with read only privilege.

You can assign other SNMP community strings, and indicate if the string is encrypted or clear. By default, the string is encrypted.

To add an encrypted community string, enter commands such as the following.

```
Brocade(config)#snmp-server community private rw
Brocade(config)#write memory
```

Syntax: `snmp-server community [0 | 1] <string>`
`ro | rw [view <viewname>] [<standard-ACL-name> | <standard-ACL-id>]`

The `<string>` parameter specifies the community string name. The string can be up to 32 characters long.

The `ro | rw` parameter specifies whether the string is **read-only (ro)** or **read-write (rw)**.

NOTE

If you issue a **no snmp-server community public ro** command and then enter a **write memory** command to save that configuration, the “public” community name is removed and will have no SNMP access. If for some reason the device is brought down and then brought up, the “no snmp-server community public ro” command is restored in the system and the “public” community string has no SNMP access.

The `0 | 1` parameter affects encryption for display of the string in the running-config and the startup-config file. Encryption is enabled by default. When encryption is enabled, the community string is encrypted in the CLI regardless of the access level you are using. In the Web Management Interface, the community string is encrypted at the read-only access level but is visible at the read-write access level.

The encryption option can be omitted (the default) or can be one of the following:

- **0** – Disables encryption for the community string you specify with the command. The community string is shown as clear text in the running-config and the startup-config file. Use this option if you do not want the display of the community string to be encrypted.
 - **1** – Assumes that the community string you enter is encrypted, and decrypts the value before using it.
-

NOTE

If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

NOTE

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the community string. In this case, the software decrypts the community string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the community string, authentication will fail because the value used by the software will not match the value you intended to use.

The command in the example above adds the read-write SNMP community string “private”. When you save the new community string to the startup-config file (using the **write memory** command), the software adds the following command to the file.

```
snmp-server community 1 <encrypted-string> rw
```

To add a non-encrypted community string, you must explicitly specify that you do not want the software to encrypt the string. Here is an example.

```
Brocade(config)#snmp-server community 0 private rw
Brocade(config)#write memory
```

The command in this example adds the string “private” in the clear, which means the string is displayed in the clear. When you save the new community string to the startup-config file, the software adds the following command to the file.

```
snmp-server community 0 private rw
```

The **view** <viewname> parameter is optional. It allows you to associate a view to the members of this community string. Enter up to 32 alphanumeric characters. If no view is specified, access to the full MIB is granted. The view that you want must exist before you can associate it to a community string. Here is an example of how to use the view parameter in the community string command.

```
Brocade(config)#snmp-s community myread ro view sysview
```

The command in this example associates the view “sysview” to the community string named “myread”. The community string has read-only access to “sysview”. For information on how to create views, refer to [“SNMP v3 configuration examples”](#) on page 436.

The <standard-ACL-name> | <standard-ACL-id> parameter is optional. It allows you to specify which ACL group will be used to filter incoming SNMP packets. You can enter either the ACL name or its ID. Here are some examples.

```
Brocade(config)#snmp-s community myread ro view sysview 2
Brocade(config)#snmp-s community myread ro view sysview myACL
```

The command in the first example indicates that ACL group 2 will filter incoming SNMP packets; whereas, the command in the second example uses the ACL group called “myACL” to filter incoming packets. Refer to [“Using ACLs to restrict SNMP access”](#) on page 114 for more information.

NOTE

To make configuration changes, including changes involving SNMP community strings, you must first configure a read-write community string using the CLI. Alternatively, you must configure another authentication method and log on to the CLI using a valid password for that method.

Displaying the SNMP community strings

To display the configured community strings, enter the following command at any CLI level.

```

Brocade#show snmp server
Contact: Marshall
Location: Copy Center
Community(ro): public
Community(rw): private
Traps
    Cold start: Enable
    Link up: Enable
    Link down: Enable
    Authentication: Enable
    Locked address violation: Enable
    Power supply failure: Enable
    Fan failure: Enable
    Temperature warning: Enable
    STP new root: Enable
    STP topology change: Enable
    ospf: Enable

Total Trap-Receiver Entries: 4
Trap-Receiver IP Address      Community
1          207.95.6.211
2          207.95.5.21

```

Syntax: show snmp server

NOTE

If display of the strings is encrypted, the strings are not displayed. Encryption is enabled by default.

User-based security model

SNMP version 3 (RFC 2570 through 2575) introduces a User-Based Security model (RFC 2574) for authentication and privacy services.

SNMP version 1 and version 2 use community strings to authenticate SNMP access to management modules. This method can still be used for authentication. In SNMP version 3, the User-Based Security model of SNMP can be used to secure against the following threats:

- Modification of information
- Masquerading the identity of an authorized entity
- Message stream modification
- Disclosure of information

SNMP version 3 also supports View-Based Access Control Mechanism (RFC 2575) to control access at the PDU level. It defines mechanisms for determining whether or not access to a managed object in a local MIB by a remote principal should be allowed. (refer to [“SNMP v3 configuration examples”](#) on page 436.)

Configuring your NMS

In order to use the SNMP version 3 features.

1. Make sure that your Network Manager System (NMS) supports SNMP version 3.
2. Configure your NMS agent with the necessary users.
3. Configure the SNMP version 3 features in Brocade devices.

Configuring SNMP version 3 on Brocade devices

Follow the steps given below to configure SNMP version 3 on Brocade devices.

1. Enter an engine ID for the management module using the **snmp-server engineid** command if you will not use the default engine ID. Refer to [“Defining the engine id”](#) on page 426.
2. Create views that will be assigned to SNMP user groups using the **snmp-server view** command. Refer to [“SNMP v3 configuration examples”](#) on page 436 for details.
3. Create ACL groups that will be assigned to SNMP user groups using the **access-list** command.
4. Create user groups using the **snmp-server group** command. Refer to [“Defining an SNMP group”](#) on page 427.
5. Create user accounts and associate these accounts to user groups using the **snmp-server user** command. Refer to [“Defining an SNMP user account”](#) on page 428.

If SNMP version 3 is not configured, then community strings by default are used to authenticate access.

Defining the engine id

A default engine ID is generated during system start up. To determine what the default engine ID of the device is, enter the **show snmp engineid** command and find the following line:

```
Local SNMP Engine ID: 800007c70300e05290ab60
```

See the section [“Displaying the Engine ID”](#) on page 434 for details.

The default engine ID guarantees the uniqueness of the engine ID for SNMP version 3. If you want to change the default engine ID, enter the **snmp-server engineid local** command.

```
Brocade(config)#snmp-server engineid local 800007c70300e05290ab60
```

Syntax: [no] **snmp-server engineid local** <hex-string>

The **local** parameter indicates that engine ID to be entered is the ID of this device, representing an SNMP management entity.

NOTE

Each user localized key depends on the SNMP server engine ID, so all users need to be reconfigured whenever the SNMP server engine ID changes.

NOTE

Since the current implementation of SNMP version 3 does not support Notification, remote engine IDs cannot be configured at this time.

The *<hex-string>* variable consists of 11 octets, entered as hexadecimal values. There are two hexadecimal characters in each octet. There should be an even number of hexadecimal characters in an engine ID.

The default engine ID has a maximum of 11 octets:

- Octets 1 through 4 represent the agent's SNMP management private enterprise number as assigned by the Internet Assigned Numbers Authority (IANA). The most significant bit of Octet 1 is "1". For example, "000007c7" is the ID for Brocade Communications, Inc. in hexadecimal. With Octet 1 always equal to "1", the first four octets in the default engine ID is always "800007c7" (which is 1991 in decimal).
- Octet 5 is always 03 in hexadecimal and indicates that the next set of values represent a MAC address.
- Octets 6 through 11 form the MAC address of the lowest port in the management module.

NOTE

Engine ID must be a unique number among the various SNMP engines in the management domain. Using the default engine ID ensures the uniqueness of the numbers.

Defining an SNMP group

SNMP groups map SNMP users to SNMP views. For each SNMP group, you can configure a read view, a write view, or both. Users who are mapped to a group will use its views for access control.

To configure an SNMP user group, enter a command such as the following.

```
Brocade(config)#snmp-server group admin v3 auth read all write all
```

Syntax: [no] **snmp-server group** *<groupname>* **v1** | **v2** | **v3** **auth** | **noauth** | **priv** [**access** *<standard-ACL-id>*] [**read** *<viewstring>* | **write** *<viewstring>*]

NOTE

This command is not used for SNMP version 1 and SNMP version 2. In these versions, groups and group views are created internally using community strings. (refer to [“SNMP community strings”](#) on page 422.) When a community string is created, two groups are created, based on the community string name. One group is for SNMP version 1 packets, while the other is for SNMP version 2 packets.

The **group** *<groupname>* parameter defines the name of the SNMP group to be created.

The **v1**, **v2**, or **v3** parameter indicates which version of SNMP is used. In most cases, you will be using v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether or not authentication will be required to access the supported views. If **auth** is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password will be required from the users.

The **access** *<standard-ACL-id>* parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** *<viewstring>* | **write** *<viewstring>* parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The `<viewstring>` variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

The value of `<viewstring>` is defined using the **snmp-server view** command. The SNMP agent comes with the "all" default view, which provides access to the entire MIB; however, it must be specified when creating the group. The "all" view also allows SNMP version 3 to be backwards compatible with SNMP version 1 and version 2.

NOTE

If you will be using a view other than the "all" view, that view must be configured before creating the user group. Refer to the section [“SNMP v3 configuration examples”](#) on page 436, especially for details on the include | exclude parameters.

Defining an SNMP user account

The **snmp-server user** command does the following:

- Creates an SNMP user.
- Defines the group to which the user will be associated.
- Defines the type of authentication to be used for SNMP access by this user.
- Specifies one of the following encryption types used to encrypt the privacy password:
 - Data Encryption Standard (DES) – A symmetric-key algorithm that uses a 56-bit key.
 - Advanced Encryption Standard (AES) – The 128-bit encryption standard adopted by the U.S. government. This standard is a symmetric cipher algorithm chosen by the National Institute of Standards and Technology (NIST) as the replacement for DES.

Here is an example of how to create an SNMP User account.

```
Brocade(config)#snmp-s user bob admin v3 access 2 auth md5 bobmd5 priv des bobdes
```

The CLI for creating SNMP version 3 users has been updated as follows.

Syntax: `[no] snmp-server user <name> <groupname> v3`
`[[access <standard-ACL-id>]`
`[[encrypted] [auth md5 <md5-password> | sha <sha-password>]`
`[priv [encrypted] des <des-password-key> | aes <aes-password-key>]]]`

The `<name>` parameter defines the SNMP user name or security name used to access the management module.

The `<groupname>` parameter identifies the SNMP group to which this user is associated or mapped. All users must be mapped to an SNMP group. Groups are defined using the **snmp-server group** command.

NOTE

The SNMP group to which the user account will be mapped should be configured before creating the user accounts; otherwise, the group will be created without any views. Also, ACL groups must be configured before configuring user accounts.

The **v3** parameter is required.

The **access <standard-ACL-id>** parameter is optional. It indicates that incoming SNMP packets are filtered based on the ACL attached to the user account.

NOTE

The ACL specified in a user account overrides the ACL assigned to the group to which the user is mapped. If no ACL is entered for the user account, then the ACL configured for the group will be used to filter packets.

The **encrypted** parameter means that the MD5 or SHA password will be a digest value. MD5 has 16 octets in the digest. SHA has 20. The digest string has to be entered as a hexadecimal string. In this case, the agent need not generate any explicit digest. If the **encrypted** parameter is not used, the user is expected to enter the authentication password string for MD5 or SHA. The agent will convert the password string to a digest, as described in RFC 2574.

The **auth md5 | sha** parameter is optional. It defines the type of encryption that the user must have to be authenticated. Choose between MD5 or SHA encryption. MD5 and SHA are two authentication protocols used in SNMP version 3.

The `<md5-password>` and `<sha-password>` define the password the user must use to be authenticated. These password must have a minimum of 8 characters. If the encrypted parameter is used, then the digest has 16 octets for MD5 or 20 octets for SHA.

NOTE

Once a password string is entered, the generated configuration displays the digest (for security reasons), not the actual password.

The **priv [encrypted]** parameter is optional after you enter the md5 or sha password. The **priv** parameter specifies the encryption type (DES or AES) used to encrypt the privacy password. If the **encrypted** keyword is used, do the following:

- If DES is the privacy protocol to be used, enter **des** followed by a 16-octet DES key in hexadecimal format for the `<des-password-key>`. If you include the encrypted keyword, enter a password string of at least 8 characters.
- If AES is the privacy protocol to be used, enter **aes** followed by the AES password key. For a small password key, enter 12 characters. For a big password key, enter 16 characters. If you include the encrypted keyword, enter a password string containing 32 hexadecimal characters.

Defining SNMP views

SNMP views are named groups of MIB objects that can be associated with user accounts to allow limited access for viewing and modification of SNMP statistics and system configuration. SNMP views can also be used with other commands that take SNMP views as an argument. SNMP views reference MIB objects using object names, numbers, wildcards, or a combination of the three. The numbers represent the hierarchical location of the object in the MIB tree. You can reference individual objects in the MIB tree or a subset of objects from the MIB tree.

To configure the number of SNMP views available on the Brocade device, enter the following command.

```
Brocade(config)#system-max view 15
```

Syntax: **system-max view** *<number-of-views>*

This command specifies the maximum number of SNMPv2 and v3 views that can be configured on a device. The number of views can be from 10 – 65536. The default is 10 views.

To add an SNMP view, enter one of the following commands.

```
Brocade(config)#snmp-server view Maynes system included
Brocade(config)#snmp-server view Maynes system.2 excluded
Brocade(config)#snmp-server view Maynes 2.3.*.6 included
Brocade(config)#write mem
```

NOTE

The **snmp-server view** command supports the MIB objects as defined in RFC 1445.

Syntax: [no] **snmp-server view** <name> <mib_tree> **included** | **excluded**

The <name> parameter can be any alphanumeric name you choose to identify the view. The names cannot contain spaces.

The <mib_tree> parameter is the name of the MIB object or family. MIB objects and MIB sub-trees can be identified by a name or by the numbers called Object Identifiers (OIDs) that represent the position of the object or sub-tree in the MIB hierarchy. You can use a wildcard (*) in the numbers to specify a sub-tree family.

The **included** | **excluded** parameter specifies whether the MIB objects identified by the <mib_family> parameter are included in the view or excluded from the view.

NOTE

All MIB objects are automatically excluded from any view unless they are explicitly included; therefore, when creating views using the **snmp-server view** command, indicate which portion of the MIB you want users to access.

For example, you may want to assign the view called “admin” a community string or user group. The “admin” view will allow access to the Brocade MIBs objects that begin with the 1.3.6.1.4.1.1991 object identifier. Enter the following command.

```
Brocade(config)#snmp-server view admin 1.3.6.1.4.1.1991 included
```

You can exclude portions of the MIB within an inclusion scope. For example, if you want to exclude the snAgentSys objects, which begin with 1.3.6.1.4.1.1991.1.1.2 object identifier from the admin view, enter a second command such as the following.

```
Brocade(config)#snmp-server view admin 1.3.6.1.4.1.1991.1.1.2 excluded
```

NOTE

Note that the exclusion is within the scope of the inclusion.

To delete a view, use the no parameter before the command.

SNMP version 3 traps

Brocade devices support SNMP notifications in SMIv2 format. This allows notifications to be encrypted and sent to the target hosts in a secure manner.

Defining an SNMP group and specifying which view is notified of traps

The SNMP group command allows configuration of a viewname for notification purpose, similar to the read and write view. The default viewname is "all", which allows access to the entire MIB.

To configure an SNMP user group, first configure SNMP v3 views using the **snmp-server view** command. Refer to “[SNMP v3 configuration examples](#)” on page 436. Then enter a command such as the following.

```
Brocade(config)#snmp-server group admin v3 auth read all write all
notify all
```

Syntax: [no] **snmp-server group** <groupname>
 v1 | **v2** | **v3**
 auth | **noauth** | **priv**
 [**access** <standard-ACL-id>] [**read** <viewstring> | **write** <viewstring> | **notify** <viewstring>]

The **group** <groupname> parameter defines the name of the SNMP group to be created.

The **v1**, **v2**, or **v3** parameter indicates which version of SNMP to use. In most cases, you will use v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether or not authentication will be required to access the supported views. If **auth** is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password will be required from the users.

The **access** <standard-ACL-id> parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** <viewstring> | **write** <viewstring> parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The **notify** view allows administrators to restrict the scope of varbind objects that will be part of the notification. All of the varbinds need to be in the included view for the notification to be created.

The <viewstring> variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

Defining the UDP port for SNMP v3 traps

The SNMP host command enhancements allow configuration of notifications in SMIv2 format, with or without encryption, in addition to the previously supported SMIv1 trap format.

You can define a port that receives the SNMP v3 traps by entering a command such as the following.

```
Brocade(config)#snmp-server host 192.168.4.11 version v3 auth security-name port
4/1
```

Syntax: [no] **snmp-server host** <ip-addr> | <ipv6-addr> **version** [**v1** | **v2c** <community-string> | **v3** **auth** | **noauth** | **priv** <security-name>] [**port** <trap-UDP-port-number>]

The <ip-addr> parameter specifies the IP address of the host that will receive the trap.

For **version**, indicate one of the following

For SNMP version 1, enter **v1** and the name of the community string (<community-string>). This string is encrypted within the system.

NOTE

If the configured version is v2c, then the notification is sent out in SMIv2 format, using the community string, but in cleartext mode. To send the SMIv2 notification in SNMPv3 packet format, configure v3 with auth or privacy parameters, or both, by specifying a security name. The actual authorization and privacy values are obtained from the security name.

For SNMP version 2c, enter **v2** and the name of the community string. This string is encrypted within the system.

For SNMP version 3, enter one of the following depending on the authorization required for the host:

- **v3 auth** <security-name>: Allow only authenticated packets.
- **v3 no auth** <security-name>: Allow all packets.
- **v3 priv** <security-name>: A password is required

For **port** <trap-UDP-port-number>, specify the UDP port number on the host that will receive the trap.

Trap MIB changes

To support the SNMP V3 trap feature, the Brocade Enterprise Trap MIB was rewritten in SMIv2 format, as follows:

- The MIB name was changed from FOUNDRY-SN-TRAP-MIB to FOUNDRY-SN-NOTIFICATION-MIB
- Individual notifications were changed to NOTIFICATION-TYPE instead of TRAP-TYPE.
- As per the SMIv2 format, each notification has an OID associated with it. The root node of the notification is snTraps (OID enterprise.foundry.0). For example, OID for snTrapRunningConfigChanged is {snTraps.73}. Earlier, each trap had a trap ID associated with it, as per the SMIv1 format.

Backward compatibility with SMIv1 trap format

The Brocade device will continue to support creation of traps in SMIv1 format, as before. To allow the device to send notifications in SMIv2 format, configure the device as described above. The default mode is still the original SMIv1 format.

Specifying an IPv6 host as an SNMP trap receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter a command such as the following.

```
Brocade(config)#snmp-server host ipv6 2001:efff:89::13
```

Syntax: `snmp-server host ipv6 <ipv6-address>`

The <ipv6-address> must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

SNMP v3 over IPv6

Some FastIron devices support IPv6 for SNMP version 3.

Restricting SNMP Access to an IPv6 Node

You can restrict SNMP access so that the Brocade device can only be accessed by the IPv6 host address that you specify. To do so, enter a command such as the following .

```
Brocade(config)#snmp-client ipv6 2001:efff:89::23
```

Syntax: **snmp-client ipv6** <ipv6-address>

The <ipv6-address> must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specifying an IPv6 host as an SNMP trap receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the Brocade device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter the **snmp-server host ipv6** command .

```
Brocade(config)#snmp-server host ipv6 2001:efff:89::13
```

Syntax: **snmp-server host ipv6** <ipv6-address>

The <ipv6-address> must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Viewing IPv6 SNMP server addresses

Many of the existing **show** commands display IPv6 addresses for IPv6 SNMP servers. The following example shows output for the **show snmp server** command.

Displaying SNMP Information

```
Brocade#show snmp server
  Contact:
  Location:
  Community(ro): .....
  Traps
    Warm/Cold start: Enable
    Link up: Enable
    Link down: Enable
    Authentication: Enable
    Locked address violation: Enable
    Power supply failure: Enable
    Fan failure: Enable
    Temperature warning: Enable
    STP new root: Enable
    STP topology change: Enable
    vsrp: Enable

  Total Trap-Receiver Entries: 4
  Trap-Receiver IP-Address      Port-Number Community
  1      192.147.201.100        162      .....
  2      4000::200              162      .....
  3      192.147.202.100        162      .....
  4      3000::200              162      .....
```

Displaying SNMP Information

This section lists the commands for viewing SNMP-related information.

Displaying the Engine ID

To display the engine ID of a management module, enter a command such as the following.

```
Brocade#show snmp engineid
Local SNMP Engine ID: 800007c70300e05290ab60
Engine Boots: 3
Engine time: 5
```

Syntax: show snmp engineid

The engine ID identifies the source or destination of the packet.

The engine boots represents the number of times that the SNMP engine reinitialized itself with the same engine ID. If the engineID is modified, the boot count is reset to 0.

The engine time represents the current time with the SNMP agent.

Displaying SNMP groups

To display the definition of an SNMP group, enter a command such as the following.

```
Brocade#show snmp group
groupname = exceptifgrp
security model = v3
security level = authNoPriv
ACL id = 2
readview = exceptif
writeview = <none>
```

Syntax: show snmp group

The value for security level can be one of the following.

Security level	Authentication
<none>	If the security model shows v1 or v2, then security level is blank. User names are not used to authenticate users; community strings are used instead.
noauthNoPriv	Displays if the security model shows v3 and user authentication is by user name only.
authNoPriv	Displays if the security model shows v3 and user authentication is by user name and the MD5 or SHA algorithm.

Displaying user information

To display the definition of an SNMP user account, enter a command such as the following.

```
Brocade#show snmp user
username = bob
ACL id = 2
group = admin
security model = v3
group ACL id = 0
authtype = md5
authkey = 3aca18d90b8d172760e2dd2e8f59b7fe
privtype = des, privkey = 1088359afb3701730173a6332d406eec
engine ID= 800007c70300e052ab0000
```

Syntax: show snmp user

Interpreting varbinds in report packets

If an SNMP version 3 request packet is to be rejected by an SNMP agent, the agent sends a report packet that contains one or more varbinds. The varbinds contain additional information, showing the cause of failures. An SNMP manager application decodes the description from the varbind. The following table presents a list of varbinds supported by the SNMP agent.

Varbind object Identifier	Description
1. 3. 6. 1. 6. 3. 11. 2. 1. 3. 0	Unknown packet data unit.
1. 3. 6. 1. 6. 3. 12. 1. 5. 0	The value of the varbind shows the engine ID that needs to be used in the snmp-server engineid command
1. 3. 6. 1. 6. 3. 15. 1. 1. 1. 0	Unsupported security level.

Varbind object Identifier	Description
1. 3. 6. 1. 6. 3. 15. 1. 1. 2. 0	Not in time packet.
1. 3. 6. 1. 6. 3. 15. 1. 1. 3. 0	Unknown user name. This varbind may also be generated: <ul style="list-style-type: none"> • If the configured ACL for this user filters out this packet. • If the group associated with the user is unknown.
1. 3. 6. 1. 6. 3. 15. 1. 1. 4. 0	Unknown engine ID. The value of this varbind would be the correct authoritative engineID that should be used.
1. 3. 6. 1. 6. 3. 15. 1. 1. 5. 0	Wrong digest.
1. 3. 6. 1. 6. 3. 15. 1. 1. 6. 0	Decryption error.

SNMP v3 configuration examples

The following sections present examples of how to configure SNMP v3.

Simple SNMP v3 configuration

```
Brocade(config)#snmp-s group admingrp v3 priv read all write all notify all
Brocade(config)#snmp-s user adminuser admingrp v3 auth md5 <auth password> priv
<privacy password>
Brocade(config)#snmp-s host <dest-ip> version v3 privacy adminuser
```

More detailed SNMP v3 configuration

```
Brocade(config)#snmp-server view internet internet included
Brocade(config)#snmp-server view system system included
Brocade(config)#snmp-server community ..... ro
Brocade(config)#snmp-server community ..... rw
Brocade(config)#snmp-server contact isc-operations
Brocade(config)#snmp-server location sdh-pillbox
Brocade(config)#snmp-server host 128.91.255.32 .....
Brocade(config)#snmp-server group ops v3 priv read internet write system
Brocade(config)#snmp-server group admin v3 priv read internet write internet
Brocade(config)#snmp-server group restricted v3 priv read internet
Brocade(config)#snmp-server user ops ops v3 encrypted auth md5
ab8e9cd6d46e7a270b8c9549d92a069 priv encrypted des
0elb153303b6188089411447dbc32de
Brocade(config)#snmp-server user admin admin v3 encrypted auth md5
0d8a2123f91bfbd8695fef16a6f4207b priv encrypted des
18e0cf359fce4fcd60df19c2b6515448
Brocade(config)#snmp-server user restricted restricted v3 encrypted auth md5
261fd8f56a3ad51c8bcecle4609f54dc priv encrypted des
d32e66152f89de9b2e0cb17a65595f43
```

Foundry Discovery Protocol (FDP) and Cisco Discovery Protocol (CDP) Packets

Table 77 lists individual Brocade switches and the discovery protocols they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 77 Supported discovery protocol features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Foundry Discovery Protocol (FDP) for IPv4 and IPv6 traffic	Yes	Yes	Yes	Yes	Yes
Cisco Discovery Protocol (CDP) for IPv4 and IPV6 traffic	Yes	Yes	Yes	Yes	Yes

FDP Overview

The Foundry Discovery Protocol (FDP) enables Brocade devices to advertise themselves to other Brocade devices on the network. When you enable FDP on a Brocade device, the device periodically advertises information including the following:

- Hostname (device ID)
- Product platform and capability
- Software version
- VLAN and Layer 3 protocol address information for the port sending the update. IP, IPX, and AppleTalk Layer 3 information is supported.

A Brocade device running FDP sends FDP updates on Layer 2 to MAC address 01-E0-52-CC-CC-CC. Other Brocade devices listening on that address receive the updates and can display the information in the updates. Brocade devices can send and receive FDP updates on Ethernet interfaces.

FDP is disabled by default.

NOTE

If FDP is not enabled on a Brocade device that receives an FDP update or the device is running a software release that does not support FDP, the update passes through the device at Layer 2.

FDP configuration

The following sections describe how to enable Foundry Discovery Protocol (FDP) and how to change the FDP update and hold timers.

Enabling FDP globally

To enable a Brocade device to globally send FDP packets, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)# fdp run
```

Syntax: [no] fdp run

The feature is disabled by default.

Enabling FDP at the interface level

By default, FDP is enabled at the interface level after FDP is enabled on the device.

When FDP is enabled globally, you can disable and re-enable FDP on individual ports.

Disable FDP by entering commands such as the following:

```
Brocade(config)# int e 2/1
Brocade(config-if-2/1)# no fdp enable
```

Enable or reenable FDP by entering commands such as the following:

```
Brocade(config-if-2/1)# fdp enable
```

Syntax: [no] fdp enable

Specifying the IP management address to advertise

When FDP is enabled, by default, the Brocade device advertises one IPv4 address and one IPv6 address to its FDP neighbors. If desired, you can configure the device to advertise only the IPv4 management address or only the IPv6 management address. You can set the configuration globally on a Layer 2 switch, or on an interface on a Layer 3 switch.

For example, to configure a Layer 2 switch to advertise the IPv4 address, enter the following command at the Global CONFIG level of the CLI:

```
Brocade(config)# fdp advertise ipv4
```

To configure a Layer 3 switch to advertise the IPv6 address, enter the following command at the Interface level of the CLI:

```
Brocade(config-if-2/1)# fdp advertise ipv6
```

Syntax: fdp advertise ipv4 | ipv6

Changing the FDP update timer

By default, a Brocade device enabled for FDP sends an FDP update every 60 seconds. You can change the update timer to a value from 5 – 900 seconds.

To change the FDP update timer, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# fdp timer 120
```

Syntax: [no] **fdp timer** <secs>

The <secs> parameter specifies the number of seconds between updates and can be from 5 – 900 seconds. The default is 60 seconds.

Changing the FDP hold time

By default, a Brocade device that receives an FDP update holds the information until one of the following events occurs:

- The device receives a new update.
- 180 seconds have passed since receipt of the last update. This is the hold time.

Once either of these events occurs, the device discards the update.

To change the FDP hold time, enter the **fdp holdtime** command at the global CONFIG level of the CLI.

```
Brocade(config)# fdp holdtime 360
```

Syntax: [no] **fdp holdtime** <secs>

The <secs> parameter specifies the number of seconds a Brocade device that receives an FDP update can hold the update before discarding it. You can specify from 10 – 255 seconds. The default is 180 seconds.

Displaying FDP information

You can display the following Foundry Discovery Protocol (FDP) information:

- FDP entries for Brocade neighbors
- Individual FDP entries
- FDP information for an interface on the device you are managing
- FDP packet statistics

NOTE

If the Brocade device has intercepted CDP updates, then the CDP information is also displayed.

Displaying neighbor information

To display a summary list of all the Brocade neighbors that have sent FDP updates to this Brocade device, enter the **show fdp neighbor** command.

```
Brocade# show fdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
(*) indicates a CDP device
```

Device ID	Local Int	Holdtm	Capability	Platform	Port ID
FastIronB	Eth 2/9	178	Router	FastIron Rou	Eth 2/9

Syntax: `show fdp neighbor [ethernet <port>] [detail]`

The **ethernet <port>** parameter lists the information for updates received on the specified port.

The **detail** parameter lists detailed information for each device.

The **show fdp neighbor** command, without optional parameters, displays the following information.

TABLE 78 Summary FDP and CDP neighbor information

This line...	Displays...
Device ID	The hostname of the neighbor.
Local Int	The interface on which this Brocade device received an FDP or CDP update for the neighbor.
Holdtm	The maximum number of seconds this device can keep the information received in the update before discarding it.
Capability	The role the neighbor is capable of playing in the network.
Platform	The product platform of the neighbor.
Port ID	The interface through which the neighbor sent the update.

To display detailed information, enter the **show fdp neighbor detail** command.

```
BrocadeA# show fdp neighbor detail
Device ID: FastIronB configured as default VLAN1, tag-type8100
Entry address(es):
  IP address: 192.168.0.13
  IPv6 address (Global): c:a:f:e:c:a:f:e
Platform: FastIron Router, Capabilities: Router
Interface: Eth 2/9
Port ID (outgoing port): Eth 2/9 is TAGGED in following VLAN(s):
  9 10 11
Holdtime : 176 seconds
Version :
Foundry, Inc. Router, IronWare Version 07.6.01b1T53 Compiled on Aug 29
2002 at 10:35:21 labeled as B2R07601b1
```

The **show fdp neighbor detail** command displays the following information.

TABLE 79 Detailed FDP and CDP neighbor information

Parameter	Definition
Device ID	The hostname of the neighbor. In addition, this line lists the VLAN memberships and other VLAN information for the neighbor port that sent the update to this device.
Entry address(es)	The Layer 3 protocol addresses configured on the neighbor port that sent the update to this device. If the neighbor is a Layer 2 Switch, this field lists the management IP address.
Platform	The product platform of the neighbor.
Capabilities	The role the neighbor is capable of playing in the network.
Interface	The interface on which this device received an FDP or CDP update for the neighbor.
Port ID	The interface through which the neighbor sent the update.
Holdtime	The maximum number of seconds this device can keep the information received in the update before discarding it.
Version	The software version running on the neighbor.

Displaying FDP entries

To display the detailed neighbor information for a specific device, enter the **show fdp entry FastIronx** command.

```
BrocadeA# show fdp entry FastIronB
Device ID: FastIronB configured as default VLAN1, tag-type8100
Entry address(es):
Platform: FastIron Router, Capabilities: Router
Interface: Eth 2/9
Port ID (outgoing port): Eth 2/9 is TAGGED in following VLAN(s):
 9 10 11
Holdtime : 176 seconds
Version :
Foundry, Inc. Router, IronWare Version 07.6.01b1T53 Compiled on Aug 29
2002 at 10:35:21 labeled as B2R07601b1
```

Syntax: **show fdp entry * | <device-id>**

The *** | <device-id>** parameter specifies the device ID. If you enter *****, the detailed updates for all neighbor devices are displayed. If you enter a specific device ID, the update for that device is displayed. For information about the display, refer to [Table 79](#).

Displaying FDP information for an interface

To display FDP information for an interface, enter a command such as the following.

```
BrocadeA# show fdp interface ethernet 2/3
FastEthernet2/3 is up, line protocol is up
Encapsulation ethernet
Sending FDP packets every 5 seconds
Holdtime is 180 seconds
```

This example shows information for Ethernet port 2/3. The port sends FDP updates every 5 seconds. Neighbors that receive the updates can hold them for up to 180 seconds before discarding them.

Syntax: `show fdp interface [ethernet <port>]`

The **ethernet <port>** parameter lists the information only for the specified interface.

Displaying FDP and CDP statistics

To display FDP and CDP packet statistics, enter the following command.

```
BrocadeA# show fdp traffic
CDP/FDP counters:
  Total packets output: 6, Input: 5
  Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
  No memory: 0, Invalid packet: 0, Fragmented: 0
  Internal errors: 0
```

Syntax: `show fdp traffic`

Clearing FDP and CDP information

You can clear the following FDP and CDP information:

- Information received in FDP and CDP updates
- FDP and CDP statistics

The same commands clear information for both FDP and CDP.

Clearing FDP and CDP neighbor information

To clear the information received in FDP and CDP updates from neighboring devices, enter the following command.

```
Brocade# clear fdp table
```

Syntax: `clear fdp table`

NOTE

This command clears all the updates for FDP and CDP.

Clearing FDP and CDP statistics

To clear FDP and CDP statistics, enter the following command.

```
Brocade# clear fdp counters
```

Syntax: `clear fdp counters`

CDP packets

Cisco Discovery Protocol (CDP) packets are used by Cisco devices to advertise themselves to other Cisco devices. By default, Brocade devices forward these packets without examining their contents. You can configure a Brocade device to intercept and display the contents of CDP packets. This feature is useful for learning device and interface information for Cisco devices in the network.

Brocade devices support intercepting and interpreting CDP version 1 and version 2 packets.

NOTE

The Brocade device can interpret only the information fields that are common to both CDP version 1 and CDP version 2.

NOTE

When you enable interception of CDP packets, the Brocade device drops the packets. As a result, Cisco devices will no longer receive the packets.

Enabling interception of CDP packets globally

To enable the device to intercept and display CDP packets, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)# cdp run
```

Syntax: [no] cdp run

The feature is disabled by default.

Enabling interception of CDP packets on an interface

You can disable and enable CDP at the interface level.

You can enter commands such as the following.

```
Brocade(config)# int e 2/1
Brocade(config-if-2/1)# cdp enable
```

Syntax: [no] cdp enable

By default, the feature is enabled on an interface once CDP is enabled on the device.

Displaying CDP information

You can display the following CDP information:

- Cisco neighbors
- CDP entries for all Cisco neighbors or a specific neighbor
- CDP packet statistics

Displaying neighbors

To display the Cisco neighbors the Brocade device has learned from CDP packets, enter the **show fdp neighbors** command.

```
Brocade# show fdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
(*) indicates a Cisco device

   Device ID      Local Int    Holdtm Capability Platform    Port ID
   -----
  (*)Router      Eth 1/1      124    R          cisco RSP4
FastEthernet5/0/0
```

To display detailed information for the neighbors, enter the **show fdp neighbors detail** command.

```
Brocade# show fdp neighbors detail
Device ID: Router
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4, Capabilities: Router
Interface: Eth 1/1, Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 150 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1,  RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

To display information about a neighbor attached to a specific port, enter a command such as the following.

```
Brocade# show fdp neighbors ethernet 1/1
Device ID: Router
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4, Capabilities: Router
Interface: Eth 1/1, Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 127 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1,  RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

Syntax: **show fdp neighbors** [**detail** | **ethernet** <port>]

Displaying CDP entries

To display CDP entries for all neighbors, enter the **show fdp entry** command.

```
Brocade# show fdp entry *
Device ID: Router
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4, Capabilities: Router
Interface: Eth 1/1, Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 124 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1, RELEASE SOFTWARE
(fcl)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

To display CDP entries for a specific device, specify the device ID, as shown in the following example.

```
Brocade# show fdp entry Router1
Device ID: Router1
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4, Capabilities: Router
Interface: Eth 1/1, Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 156 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1, RELEASE SOFTWARE
(fcl)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

Syntax: **show fdp entry *** | <device-id>

Displaying CDP statistics

To display CDP packet statistics, enter the **show fdp traffic** command.

```
Brocade# show fdp traffic
CDP counters:
  Total packets output: 0, Input: 3
  Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
  No memory: 0, Invalid packet: 0, Fragmented: 0
```

Syntax: **show fdp traffic**

Clearing CDP information

You can clear the following CDP information:

- Cisco Neighbor information
- CDP statistics

CDP packets

To clear the Cisco neighbor information, enter the **clear fdp table** command.

```
Brocade# clear fdp table
```

Syntax: clear fdp table

To clear CDP statistics, enter the following command.

```
Brocade# clear fdp counters
```

Syntax: clear fdp counters

LLDP and LLDP-MED

Table 80 lists the individual Brocade FastIron switches and the Link Layer Discovery Protocol (LLDP) features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 80 Supported LLDP features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
LLDP	Yes	Yes	Yes	Yes	Yes
LLDP-MED	Yes	Yes	Yes	Yes	Yes
Support for tagged LLDP packets	Yes	Yes	Yes	Yes	Yes
IPv4 management address advertisement	Yes	Yes	Yes	Yes	Yes
IPv6 management address advertisement	Yes	Yes	Yes	Yes	Yes
LLDP operating mode setting per port	Yes	Yes	Yes	Yes	Yes
Setting the maximum number of LLDP neighbors	Yes	Yes	Yes	Yes	Yes
SNMP and Syslog messages	Yes	Yes	Yes	Yes	Yes
LLDP transmission intervals	Yes	Yes	Yes	Yes	Yes
Holdtime multiplier for transmit TTL	Yes	Yes	Yes	Yes	Yes
Configuring the minimum time between port reinitializations	Yes	Yes	Yes	Yes	Yes
Fast start repeat count for LLDP-MED	Yes	Yes	Yes	Yes	Yes
Location ID for LLDP-MED	Yes	Yes	Yes	Yes	Yes
LLDP-MED network policy	Yes	Yes	Yes	Yes	Yes
LLDP statistics and configuration details	Yes	Yes	Yes	Yes	Yes

This chapter describes how to configure the following protocols:

Link layer discovery protocol (LLDP) – The Layer 2 network discovery protocol described in the IEEE 802.1AB standard, *Station and Media Access Control Connectivity Discovery*. This protocol enables a station to advertise its capabilities to, and to discover, other LLDP-enabled stations in the same 802 LAN segments.

LLDP media endpoint devices (LLDP-MED) – The Layer 2 network discovery protocol extension described in the ANSI/TIA-1057 standard, *LLDP for Media Endpoint Devices*. This protocol enables a switch to configure and manage connected Media Endpoint devices that need to send media streams across the network (e.g., IP telephones and security cameras).

LLDP enables network discovery between Network Connectivity devices (such as switches), whereas LLDP-MED enables network discovery at the edge of the network, between Network Connectivity devices and media Endpoint devices (such as IP phones).

The information generated by LLDP and LLDP-MED can be used to diagnose and troubleshoot misconfigurations on both sides of a link. For example, the information generated can be used to discover devices with misconfigured or unreachable IP addresses, and to detect port speed and duplex mismatches.

LLDP and LLDP-MED facilitate interoperability across multiple vendor devices. Brocade devices running LLDP can interoperate with third-party devices running LLDP.

The Brocade LLDP and LLDP-MED implementation adheres to the IEEE 802.1AB and TIA-1057 standards.

LLDP terms used in this chapter

Endpoint device – An LLDP-MED device located at the network edge, that provides some aspect of IP communications service based on IEEE 802 LAN technology. An Endpoint device is classified in one of three class types (I, II, or III) and can be an IP telephone, softphone, VoIP gateway, or conference bridge, among others.

LLDP agent – The protocol entity that implements LLDP for a particular IEEE 802 device. Depending on the configured LLDP operating mode, an LLDP agent can send and receive LLDP advertisements (frames), or send LLDP advertisements only, or receive LLDP advertisements only.

LLDPDU (LLDP Data Unit) – A unit of information in an LLDP packet that consists of a sequence of short variable length information elements, known as **TLVs**. LLDP pass-through is not supported in conformance to IEEE standard.

MIB (Management Information Base) – A virtual database that identifies each manageable object by its name, syntax, accessibility, and status, along with a text description and unique object identifier (OID). The database is accessible by a Network Management Station (NMS) using a management protocol such as the Simple Network Management Protocol (SNMP).

Network connectivity device – A forwarding 802 LAN device, such as a router, switch, or wireless access point.

Station – A node in a network.

TLV (Type-Length-Value) – An information element in an LLDPDU that describes the type of information being sent, the length of the information string, and the value (actual information) that will be transmitted.

TTL (Time-to-Live) – Specifies the length of time that the receiving device should maintain the information acquired through LLDP in its MIB.

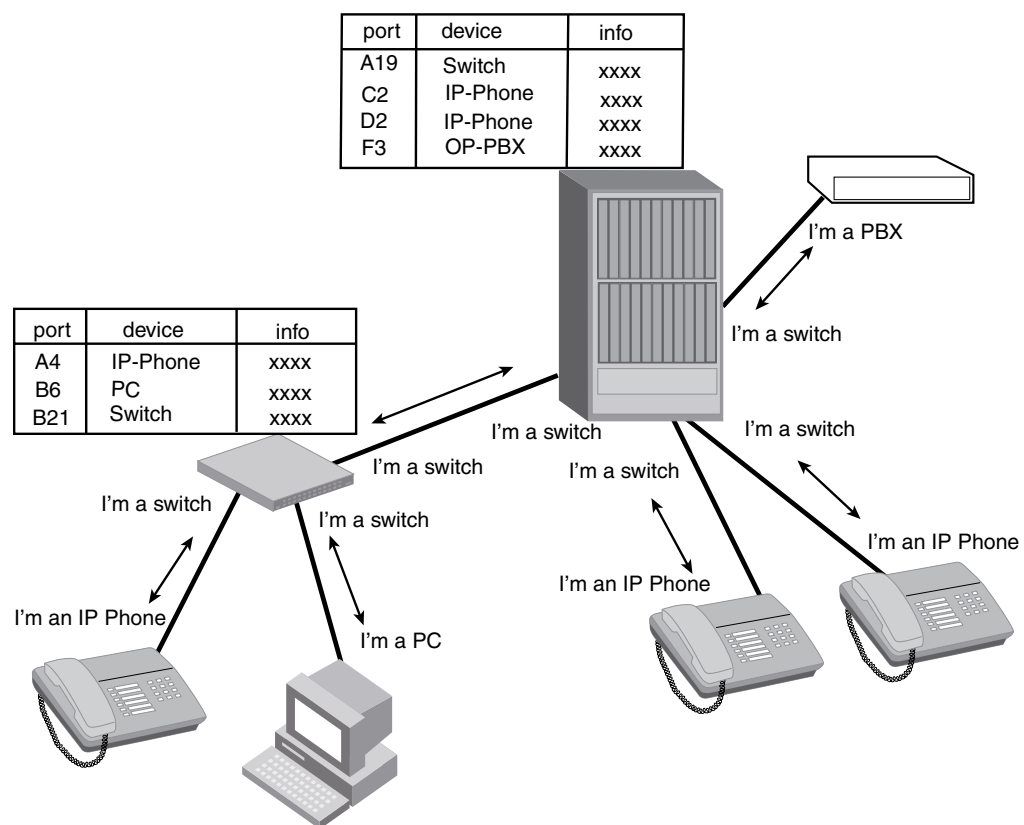
LLDP overview

LLDP enables a station attached to an IEEE 802 LAN/MAN to advertise its capabilities to, and to discover, other stations in the same 802 LAN segments.

The information distributed by LLDP (the advertisement) is stored by the receiving device in a standard Management Information Base (MIB), accessible by a Network Management System (NMS) using a management protocol such as the Simple Network Management Protocol (SNMP). The information also can be viewed from the CLI, using **show LLDP** commands.

Figure 34 illustrates LLDP connectivity

FIGURE 34 LLDP connectivity



Benefits of LLDP

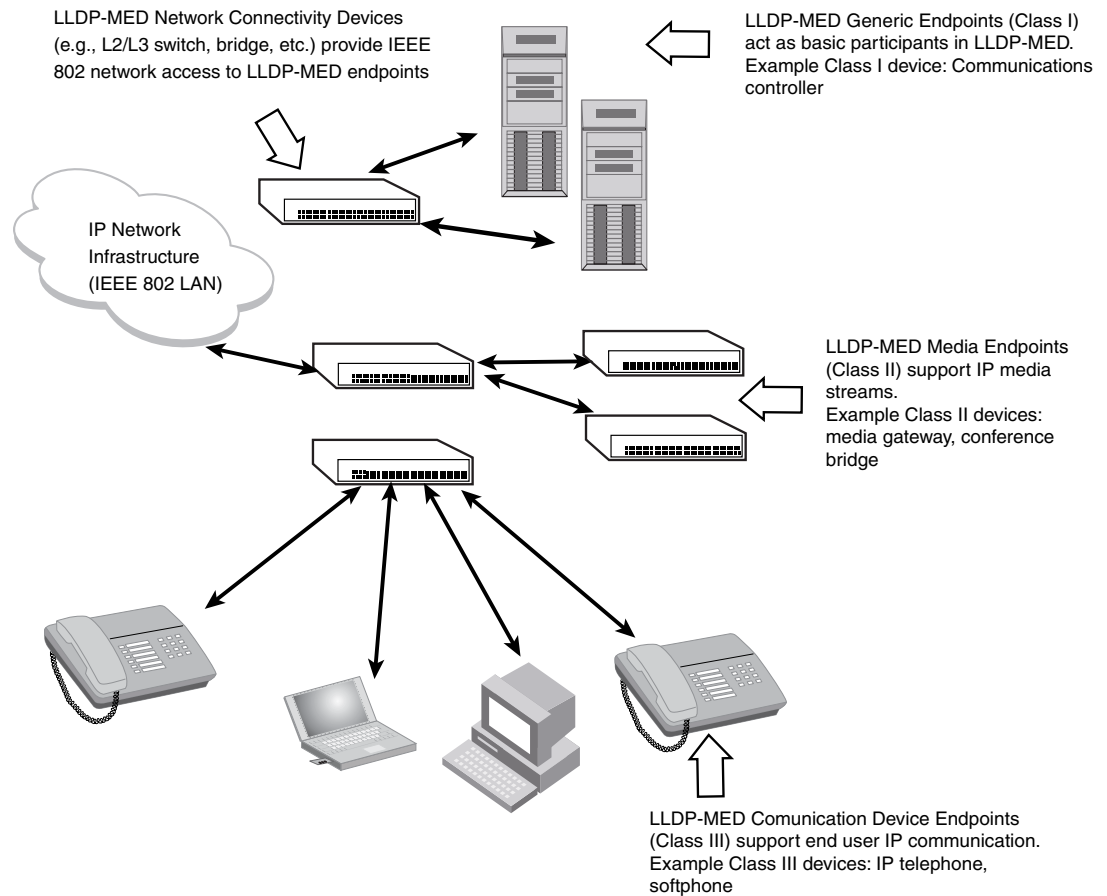
LLDP provides the following benefits:

- Network Management:
 - Simplifies the use of and enhances the ability of network management tools in multi-vendor environments
 - Enables discovery of accurate physical network topologies such as which devices are neighbors and through which ports they connect
 - Enables discovery of stations in multi-vendor environments
- Network Inventory Data:
 - Supports optional system name, system description, system capabilities and management address
 - System description can contain the device product name or model number, version of hardware type, and operating system
 - Provides device capability, such as switch, router, or WLAN access point
- Network troubleshooting:
 - Information generated by LLDP can be used to detect speed and duplex mismatches
 - Accurate topologies simplify troubleshooting within enterprise networks
 - Can discover devices with misconfigured or unreachable IP addresses

LLDP-MED overview

LLDP-MED is an extension to LLDP. This protocol enables advanced LLDP features in a Voice over IP (VoIP) network. Whereas LLDP enables network discovery between Network Connectivity devices, LLDP-MED enables network discovery between Network Connectivity devices and media Endpoints such as, IP telephones, softphones, VoIP gateways and conference bridges

.[Figure 35](#) demonstrates LLDP-MED connectivity.

FIGURE 35 LLDP-MED connectivity

Benefits of LLDP-MED

LLDP-MED provides the following benefits:

- Vendor-independent management capabilities, enabling different IP telephony systems to interoperate in one network.
- Automatically deploys network policies, such as Layer 2 and Layer 3 QoS policies and Voice VLANs.
- Supports E-911 Emergency Call Services (ECS) for IP telephony
- Collects Endpoint inventory information
- Network troubleshooting
 - Helps to detect improper network policy configuration

LLDP-MED class

An LLDP-MED class specifies an Endpoint type and its capabilities. An Endpoint can belong to one of three LLDP-MED class types:

- **Class 1 (Generic endpoint)** – A Class 1 Endpoint requires basic LLDP discovery services, but does not support IP media nor does it act as an end-user communication appliance. A Class 1 Endpoint can be an IP communications controller, other communication-related server, or other device requiring basic LLDP discovery services.
- **Class 2 (Media endpoint)** – A Class 2 Endpoint supports media streams and may or may not be associated with a particular end user. Device capabilities include media streaming, as well as all of the capabilities defined for Class 1 Endpoints. A Class 2 Endpoint can be a voice/media gateway, conference, bridge, media server, etc..
- **Class 3 (Communication endpoint)** – A Class 3 Endpoint supports end user IP communication. Capabilities include aspects related to end user devices, as well as all of the capabilities defined for Class 1 and Class 2 Endpoints. A Class 3 Endpoint can be an IP telephone, softphone (PC-based phone), or other communication device that directly supports the end user.

Discovery services defined in Class 3 include location identifier (ECS/E911) information and inventory management.

The LLDP-MED device class is advertised when LLDP-MED is enabled on a port.

Figure 35 illustrates LLDP-MED connectivity and supported LLDP-MED classes.

General LLDP operating principles

LLDP and LLDP-MED use the services of the Data Link sublayers, Logical Link Control and Media Access Control, to transmit and receive information to and from other **LLDP Agents** (protocol entities that implement LLDP).

LLDP is a one-way protocol. An LLDP agent can transmit and receive information to and from another LLDP agent located on an adjacent device, but it cannot solicit information from another LLDP agent, nor can it acknowledge information received from another LLDP agent.

LLDP operating modes

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. You can disable a port's ability to transmit and receive LLDP packets, or change the operating mode to one of the following:

- Transmit LLDP information only
- Receive LLDP information only

LLDP transmit mode

An LLDP agent sends LLDP packets to adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

An LLDP agent initiates the transmission of LLDP packets whenever the transmit countdown timing counter expires, or whenever LLDP information has changed. When a transmit cycle is initiated, the LLDP manager extracts the MIB objects and formats this information into TLVs. The TLVs are inserted into an LLDPDU, addressing parameters are prepended to the LLDPDU, and the information is sent out LLDP-enabled ports to adjacent LLDP-enabled devices.

LLDP receive mode

An LLDP agent receives LLDP packets from adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

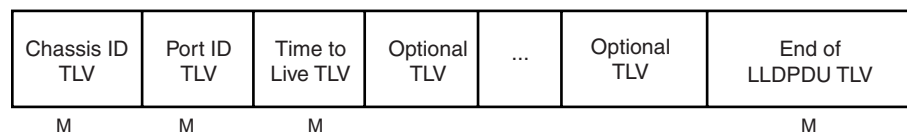
When an LLDP agent receives LLDP packets, it checks to ensure that the LLDPDUs contain the correct sequence of mandatory TLVs, then validates optional TLVs. If the LLDP agent detects any errors in the LLDPDUs and TLVs, it drops them in software. TLVs that are not recognized but do not contain basic formatting errors, are assumed to be valid and are assigned a temporary identification index and stored for future possible alter retrieval by network management. All validated TLVs are stored in the neighbor database.

LLDP packets

LLDP agents transmit information about a sending device/port in packets called LLDP Data Units (LLDPDUs). All the LLDP information to be communicated by a device is contained within a single 1500 byte packet. A device receiving LLDP packets is not permitted to combine information from multiple packets.

As shown in [Figure 36](#), each LLDPDU has three mandatory TLVs, an End of LLDPDU TLV, plus optional TLVs as selected by network management.

FIGURE 36 LLDPDU packet format



M = mandatory TLV (required for all LLDPDUs)

Each LLDPDU consists of an untagged Ethernet header and a sequence of short, variable length information elements known as type, length, value (TLV).

TLVs have Type, Length, and Value fields, where:

- **Type** identifies the kind of information being sent
- **Length** indicates the length (in octets) of the information string
- **Value** is the actual information being sent (for example, a binary bit map or an alpha-numeric string containing one or more fields).

TLV support

This section lists the LLDP and LLDP-MED TLV support.

LLDP TLVs

There are two types of LLDP TLVs, as specified in the IEEE 802.3AB standard:

- **Basic management TLVs** consist of both optional general system information TLVs as well as mandatory TLVs.

Mandatory TLVs cannot be manually configured. They are always the first three TLVs in the LLDPDU, and are part of the packet header.

General system information TLVs are optional in LLDP implementations and are defined by the Network Administrator.

Brocade devices support the following Basic Management TLVs:

- Chassis ID (mandatory)
- Port ID (mandatory)
- Time to Live (mandatory)
- Port description
- System name
- System description
- System capabilities
- Management address
- End of LLDPDU
- **Organizationally-specific TLVs** are optional in LLDP implementations and are defined and encoded by individual organizations or vendors. These TLVs include support for, but are not limited to, the IEEE 802.1 and 802.3 standards and the TIA-1057 standard.

Brocade devices support the following Organizationally-specific TLVs:

- **802.1 organizationally-specific TLVs**

Port VLAN ID

VLAN name TLV

- **802.3 organizationally-specific TLVs**

MAC/PHY configuration/status

Power through MDI

Link aggregation

Maximum frame size

LLDP-MED TLVs

Brocade devices honor and send the following LLDP-MED TLVs, as defined in the TIA-1057 standard:

- LLDP-MED capabilities
- Network policy
- Location identification
- Extended power-via-MDI

Mandatory TLVs

When an LLDP agent transmits LLDP packets to other agents in the same 802 LAN segments, the following mandatory TLVs are always included:

- Chassis ID
- Port ID
- Time to Live (TTL)

This section describes the above TLVs in detail.

Chassis ID

The Chassis ID identifies the device that sent the LLDP packets.

There are several ways in which a device may be identified. A chassis ID subtype, included in the TLV and shown in [Table 81](#), indicates how the device is being referenced in the Chassis ID field.

TABLE 81 Chassis ID subtypes

ID subtype	Description
0	Reserved
1	Chassis component
2	Interface alias
3	Port component
4	MAC address
5	Network address
6	Interface name
7	Locally assigned
8 – 255	Reserved

Brocade devices use chassis ID subtype 4, the base MAC address of the device. Other third party devices may use a chassis ID subtype other than 4. The chassis ID will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Chassis ID (MAC address): 0012.f233.e2c0
```

The chassis ID TLV is always the first TLV in the LLDPDU.

Port ID

The Port ID identifies the port from which LLDP packets were sent.

There are several ways in which a port may be identified, as shown in [Figure 82](#). A port ID subtype, included in the TLV, indicates how the port is being referenced in the Port ID field.

TABLE 82 Port ID subtypes

ID subtype	Description
0	Reserved
1	Interface alias
2	Port component
3	MAC address
4	Network address
5	Interface name
6	Agent circuit ID
7	Locally assigned
8 – 255	Reserved

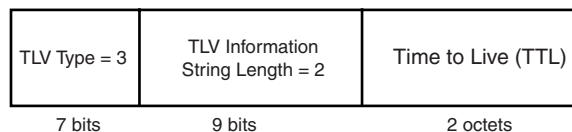
Brocade devices use port ID subtype 3, the permanent MAC address associated with the port. Other third party devices may use a port ID subtype other than 3. The port ID appears similar to the following on the remote device, and in the CLI display output on the Brocade device (show lldp local-info).

```
Port ID (MAC address): 0012.f233.e2d3
```

The LLDPDU format is shown in [“LLDPDU packet format”](#) on page 453.

The Port ID TLV format is shown below.

FIGURE 37 Port ID TLV packet format



TTL value

The Time to Live (TTL) Value is the length of time the receiving device should maintain the information acquired by LLDP in its MIB.

The TTL value is automatically computed based on the LLDP configuration settings. The TTL value will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (show lldp local-info).

```
Time to live: 40 seconds
```

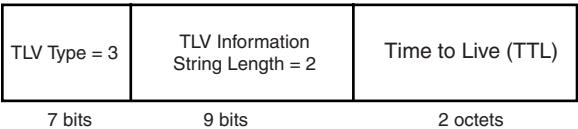
If the TTL field has a value other than zero, the receiving LLDP agent is notified to completely replace all information associated with the LLDP agent/port with the information in the received LLDPDU.

If the TTL field value is zero, the receiving LLDP agent is notified that all system information associated with the LLDP agent/port is to be deleted. This TLV may be used, for example, to signal that the sending port has initiated a port shutdown procedure.

The LLDPDU format is shown in [“LLDPDU packet format”](#) on page 453.

The TTL TLV format is shown below.

FIGURE 38 TTL TLV packet format



MIB support

Brocade devices support the following standard management information base (MIB) modules:

- LLDP-MIB
- LLDP-EXT-DOT1-MIB
- LLDP-EXT-DOT3-MIB
- LLDP-EXT-MED-MIB

Syslog messages

Syslog messages for LLDP provide management applications with information related to MIB data consistency and general status. These Syslog messages correspond to the `IldpRemTablesChange` SNMP notifications. Refer to [“Enabling LLDP SNMP notifications and Syslog messages”](#) on page 462.

Syslog messages for LLDP-MED provide management applications with information related to topology changes. These Syslog messages correspond to the `IldpXMedTopologyChangeDetected` SNMP notifications. Refer to [“Enabling SNMP notifications and Syslog messages for LLDP-MED topology changes”](#) on page 475.

LLDP configuration

This section describes how to enable and configure LLDP.

[Table 83](#) lists the LLDP global-level tasks and the default behavior/value for each task.

TABLE 83 LLDP global configuration tasks and default behavior /value

Global task	Default behavior / value when LLDP is enabled
Enabling LLDP on a global basis	Disabled
Specifying the maximum number of LLDP neighbors per device	Automatically set to 392 neighbors per device
Specifying the maximum number of LLDP neighbors per port	Automatically set to 4 neighbors per port
Enabling SNMP notifications and Syslog messages	Disabled
Changing the minimum time between SNMP traps and Syslog messages	Automatically set to 2 seconds when SNMP notifications and Syslog messages for LLDP are enabled
Enabling and disabling TLV advertisements	When LLDP transmit is enabled, by default, the Brocade device will automatically advertise LLDP capabilities, except for the system description, VLAN name, and power-via-MDI information, which may be configured by the system administrator. Also, if desired, you can disable the advertisement of individual TLVs.
Changing the minimum time between LLDP transmissions	Automatically set to 2 seconds
Changing the interval between regular LLDP transmissions	Automatically set to 30 seconds
Changing the holdtime multiplier for transmit TTL	Automatically set to 4
Changing the minimum time between port reinitializations	Automatically set to 2 seconds

LLDP configuration notes and considerations

- LLDP is supported on Ethernet interfaces only.
- If a port is 802.1X-enabled, the transmission and reception of LLDP packets will only take place while the port is authorized.
- Cisco Discovery Protocol (CDP) and Brocade Discovery Protocol (FDP) run independently of LLDP. Therefore, these discovery protocols can run simultaneously on the same device.
- By default, the Brocade device limits the number of neighbors per port to four, and staggers the transmission of LLDP packets on different ports, in order to minimize any high-usage spikes to the CPU.
- By default, the Brocade device forwards
- Ports that are in blocking mode (spanning tree) can still receive LLDP packets from a forwarding port.
- Auto-negotiation status indicates what is being advertised by the port for 802.3 auto-negotiation.

Enabling and disabling LLDP

LLDP is enabled by default on individual ports. However, to run LLDP, you must first enable it on a global basis (on the entire device).

To enable LLDP globally, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)#lldp run
```

Syntax: [no] lldp run

Enabling support for tagged LLDP packets

By default, Brocade devices do not accept tagged LLDP packets from other vendors' devices. To enable support, apply the command **lldp tagged-packets process** at the Global CONFIG level of the CLI. When enabled, the device will accept incoming LLDP tagged packets if the VLAN tag matches any of the following:

- a configured VLAN on the port
- the default VLAN for a tagged port
- the configured untagged VLAN for a dual-mode port

To enable support for tagged LLDP packets, enter the following command.

```
Brocade(config)#lldp tagged-packets process
```

Syntax: [no] lldp tagged-packets process

Changing a port LLDP operating mode

LLDP packets are not exchanged until LLDP is enabled on a global basis. When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. You can disable a port's ability to transmit and receive LLDP packets, or change the operating mode to one of the following:

- Transmit LLDP information only
- Receive LLDP information only

You can configure a different operating mode for each port on the Brocade device. For example, you could disable the receipt and transmission of LLDP packets on port e 2/1, configure port e 2/3 to only receive LLDP packets, and configure port e 2/5 to only transmit LLDP packets.

The following sections show how to change the operating mode.

Enabling and disabling receive and transmit mode

To disable the receipt and transmission of LLDP packets on individual ports, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#no lldp enable ports e 2/4 e 2/5
```

The above command disables LLDP on ports 2/4 and 2/5. These ports will not transmit nor receive LLDP packets.

To enable LLDP on a port after it has been disabled, enter the following command.

```
Brocade(config)#lldp enable ports e 2/4
```

Syntax: `[no] lldp enable ports ethernet <port-list> | all`

Use the `[no]` form of the command to disable the receipt and transmission of LLDP packets on a port.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

NOTE

When a port is configured to both receive and transmit LLDP packets and the MED capabilities TLV is enabled, LLDP-MED is enabled as well. LLDP-MED is not enabled if the operating mode is set to receive only or transmit only.

Enabling and disabling receive only mode

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. To change the LLDP operating mode from receive and transmit mode to receive only mode, simply disable the transmit mode. Enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#no lldp enable transmit ports e 2/4 e 2/5 e 2/6
```

The above command changes the LLDP operating mode on ports 2/4, 2/5, and 2/6 from transmit and receive mode to receive only mode.

To change a port LLDP operating mode from transmit only to receive only, first disable the transmit only mode, then enable the receive only mode. Enter commands such as the following.

```
Brocade(config)#no lldp enable transmit ports e 2/7 e 2/8 e 2/9
Brocade(config)#lldp enable receive ports e 2/7 e 2/8 e 2/9
```

The above commands change the LLDP operating mode on ports 2/7, 2/8, and 2/9, from transmit only to receive only. Note that if you do not disable the transmit only mode, you will configure the port to both transmit and receive LLDP packets.

NOTE

LLDP-MED is not enabled when you enable the receive only operating mode. To enable LLDP-MED, you must configure the port to both receive and transmit LLDP packets. Refer to [“Enabling and disabling receive and transmit mode”](#) on page 459.

Syntax: `[no] lldp enable receive ports ethernet <port-list> | all`

Use the `[no]` form of the command to disable the receive only mode.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Enabling and Disabling Transmit Only Mode

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. To change the LLDP operating mode to transmit only mode, simply disable the receive mode. Enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#no lldp enable receive ports e 2/4 e 2/5 e 2/6
```

The above command changes the LLDP operating mode on ports 2/4, 2/5, and 2/6 from transmit and receive mode to transmit only mode. Any incoming LLDP packets will be dropped in software.

To change a port LLDP operating mode from receive only to transmit only, first disable the receive only mode, then enable the transmit only mode. For example, enter commands such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#no lldp enable receive ports e 2/7 e 2/8
Brocade(config)#lldp enable transmit ports e 2/7 e 2/8
```

The above commands change the LLDP operating mode on ports 2/7 and 2/8 from receive only mode to transmit only mode. Any incoming LLDP packets will be dropped in software. Note that if you do not disable receive only mode, you will configure the port to both receive and transmit LLDP packets.

NOTE

LLDP-MED is not enabled when you enable the transmit only operating mode. To enable LLDP-MED, you must configure the port to both receive and transmit LLDP packets. Refer to [“Enabling and disabling receive and transmit mode”](#) on page 459.

Syntax: `[no] lldp enable transmit ports ethernet <port-list> | all`

Use the [no] form of the command to disable the *transmit only* mode.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Maximum number of LLDP neighbors

You can change the limit of the number of LLDP neighbors for which LLDP data will be retained, per device as well as per port.

Specifying the maximum number of LLDP neighbors per device

You can change the maximum number of neighbors for which LLDP data will be retained for the entire system.

For example, to change the maximum number of LLDP neighbors for the entire device to 26, enter the following command.

```
Brocade(config)#lldp max-total-neighbors 26
```

Syntax: [no] lldp max-total-neighbors <value>

Use the [no] form of the command to remove the static configuration and revert to the default value of 392.

where <value> is a number between 16 and 8192. The default number of LLDP neighbors per device is 392.

Use the **show lldp** command to view the configuration.

Specifying the maximum number of LLDP neighbors per port

You can change the maximum number of LLDP neighbors for which LLDP data will be retained for each port. By default, the maximum number is four and you can change this to a value between one and 64.

For example, to change the maximum number of LLDP neighbors to six, enter the following command.

```
Brocade(config)#lldp max-neighbors-per-port 6
```

Syntax: [no] lldp max-neighbors-per-port <value>

Use the [no] form of the command to remove the static configuration and revert to the default value of four.

where <value> is a number from 1 to 64. The default is number of LLDP neighbors per port is four.

Use the **show lldp** command to view the configuration.

Enabling LLDP SNMP notifications and Syslog messages

SNMP notifications and Syslog messages for LLDP provide management applications with information related to MIB data updates and general status.

When you enable LLDP SNMP notifications, corresponding Syslog messages are enabled as well. When you enable LLDP SNMP notifications, the device will send traps and corresponding Syslog messages whenever there are changes to the LLDP data received from neighboring devices.

LLDP SNMP notifications and corresponding Syslog messages are disabled by default. To enable them, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp enable snmp notifications ports e 4/2 to 4/6
```

The above command enables SNMP notifications and corresponding Syslog messages on ports 4/2 and 4/6. By default, the device will send no more than one SNMP notification and Syslog message within a five second period. If desired, you can change this interval. Refer to [“Specifying the minimum time between SNMP traps and Syslog messages”](#) on page 463.

Syntax: [no] lldp enable snmp notifications ports ethernet <port-list> | all

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Specifying the minimum time between SNMP traps and Syslog messages

When SNMP notifications and Syslog messages for LLDP are enabled, the device will send no more than one SNMP notification and corresponding Syslog message within a five second period. If desired, you can throttle the amount of time between transmission of SNMP traps (lldpRemTablesChange) and Syslog messages from five seconds up to a value equal to one hour (3600 seconds).

NOTE

Because LLDP Syslog messages are rate limited, some LLDP information given by the system will not match the current LLDP statistics (as shown in the **show lldp statistics** command output).

To change the minimum time interval between traps and Syslog messages, enter a command such as the following.

```
Brocade(config)#lldp snmp-notification-interval 60
```

When the above command is applied, the LLDP agent will send no more than one SNMP notification and Syslog message every 60 seconds.

Syntax: [no] lldp snmp-notification-interval <seconds>

where <seconds> is a value between 5 and 3600. The default is 5 seconds.

Changing the minimum time between LLDP transmissions

The LLDP transmit delay timer limits the number of LLDP frames an LLDP agent can send within a specified time frame. When you enable LLDP, the system automatically sets the LLDP transmit delay timer to two seconds. If desired, you can change the default behavior from two seconds to a value between 1 and 8192 seconds.

NOTE

The LLDP transmit delay timer must not be greater than one quarter of the LLDP transmission interval (CLI command **lldp transmit-interval**).

The LLDP transmit delay timer prevents an LLDP agent from transmitting a series of successive LLDP frames during a short time period, when rapid changes occur in LLDP. It also increases the probability that multiple changes, rather than single changes, will be reported in each LLDP frame.

To change the LLDP transmit delay timer, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp transmit-delay 7
```

The above command causes the LLDP agent to wait a minimum of seven seconds after transmitting an LLDP frame and before sending another LLDP frame.

Syntax: **[no] lldp transmit-delay <seconds>**

where <seconds> is a value between 1 and 8192. The default is two seconds. Note that this value must not be greater than one quarter of the LLDP transmission interval (CLI command **lldp transmit-interval**).

Changing the interval between regular LLDP transmissions

The LLDP transmit interval specifies the number of seconds between regular LLDP packet transmissions. When you enable LLDP, by default, the device will wait 30 seconds between regular LLDP packet transmissions. If desired, you can change the default behavior from 30 seconds to a value between 5 and 32768 seconds.

To change the LLDP transmission interval, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp transmit-interval 40
```

The above command causes the LLDP agent to transmit LLDP frames every 40 seconds.

Syntax: **[no] lldp transmit-interval <seconds>**

where <seconds> is a value from 5 to 32768. The default is 30 seconds.

NOTE

Setting the transmit interval or transmit holdtime multiplier, or both, to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high. This in turn can affect how long a receiving device will retain the information if it is not refreshed.

Changing the holdtime multiplier for transmit TTL

The holdtime multiplier for transmit TTL is used to compute the actual time-to-live (TTL) value used in an LLDP frame. The TTL value is the length of time the receiving device should maintain the information in its MIB. When you enable LLDP, the device automatically sets the holdtime multiplier for TTL to four. If desired, you can change the default behavior from four to a value between two and ten.

To compute the TTL value, the system multiplies the LLDP transmit interval by the holdtime multiplier. For example, if the LLDP transmit interval is 30 and the holdtime multiplier for TTL is 4, then the value 120 is encoded in the TTL field in the LLDP header.

To change the holdtime multiplier, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp transmit-hold 6
```

Syntax: [no] lldp transmit-hold <value>

where <value> is a number from 2 to 10. The default value is 4.

NOTE

Setting the transmit interval or transmit holdtime multiplier, or both, to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high. This in turn can affect how long a receiving device will retain the information if it is not refreshed.

Changing the minimum time between port reinitializations

The LLDP re-initialization delay timer specifies the minimum number of seconds the device will wait from when LLDP is disabled on a port, until it will honor a request to re-enable LLDP on that port. When you enable LLDP, the system sets the re-initialization delay timer to two seconds. If desired, you can change the default behavior from two seconds to a value between one and ten seconds.

To set the re-initialization delay timer, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp reinit-delay 5
```

The above command causes the device to wait five seconds after LLDP is disabled, before attempting to honor a request to re-enable it.

Syntax: [no] lldp reinit-delay <seconds>

where <seconds> is a value from 1 – 10. The default is two seconds.

LLDP TLVs advertised by the Brocade device

When LLDP is enabled on a global basis, the Brocade device will automatically advertise the following information, except for the features noted:

General system information:

- Management address
- Port description
- System capabilities
- System description (not automatically advertised)
- System name

802.1 capabilities:

- VLAN name (not automatically advertised)
- Untagged VLAN ID

802.3 capabilities:

- Link aggregation information
- MAC/PHY configuration and status
- Maximum frame size
- Power-via-MDI information (not automatically advertised)

The above TLVs are described in detail in the following sections.

NOTE

The system description, VLAN name, and power-via-MDI information TLVs are not automatically enabled. The following sections show how to enable these advertisements.

General system information for LLDP

Except for the system description, the Brocade device will advertise the following system information when LLDP is enabled on a global basis:

- Management address
- Port description
- System capabilities
- System description (not automatically advertised)
- System name

Management Address

A management address is normally an IPv4 or IPv6 address that can be used to manage the device. Management address advertising has two modes: default, or explicitly configured. The default mode is used when no addresses are configured to be advertised for a given port. If any addresses are configured to be advertised for a given port, then only those addresses are advertised. This applies across address types, so for example, if just one IPv4 address is explicitly configured to be advertised for a port, then no IPv6 addresses will be advertised for that port (since none were configured to be advertised), even if IPv6 addresses are configured within the system.

If no management address is explicitly configured to be advertised, the Brocade device will use the first available IPv4 address and the first available IPv6 address (so it may advertise IPv4, IPv6 or both). A Layer 3 switch will select the first available address of each type from those configured on the following types of interfaces, in the following order of preference:

- Physical port on which LLDP will be transmitting the packet
- Virtual router interface (VE) on a VLAN that the port is a member of
- Dedicated management port
- Loopback interface
- Virtual router interface (VE) on any other VLAN
- Other physical port
- Other interface

For IPv6 addresses, link-local and anycast addresses will be excluded from these searches.

If no IP address is configured on any of the above, the port's current MAC address will be advertised.

To advertise a IPv4 management address, enter a command such as the following:


```
Brocade(config)#lldp advertise management-address ipv4 209.157.2.1 ports e 1/4
```

The management address will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**):

```
Management address (IPv4): 209.157.2.1
```

Syntax: [no] lldp advertise management-address ipv4 <ipv4 address> **ports ethernet** <port list> | **all**

To support an IPv6 management address, there is a similar command that has equivalent behavior as the IPv4 command.

To advertise an IPv6 management address, enter a command such as the following:

```
Brocade(config)#lldp advertise management-address ipv6 1234:5678::90 ports e 2/7
```

Syntax: [no] lldp advertise management-address ipv6 <ipv6 address> **ports ethernet** <port list> | **all**

<ipv4 address> or <ipv6 address> or both are the addresses that may be used to reach higher layer entities to assist discovery by network management. In addition to management addresses, the advertisement will include the system interface number associated with the management address.

For <port list>, specify the port(s) in the format [<slotnum>/]<portnum>, where <slotnum> is required on chassis devices only. You can list all of the ports individually; use the keyword to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Port description

The port description TLV identifies the port from which the LLDP agent transmitted the advertisement. The port description is taken from the ifDescr MIB object from MIB-II.

By default, the port description is automatically advertised when LLDP is enabled on a global basis. To disable advertisement of the port description, enter a command such as the following.

```
Brocade(config)#no lldp advertise port-description ports e 2/4 to 2/12
```

The port description will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Port description: "GigabitEthernet20"
```

Syntax: [no] lldp advertise port-description **ports ethernet** <port-list> | **all**

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System capabilities

The system capabilities TLV identifies the primary functions of the device and indicates whether these primary functions are enabled. The primary functions can be one or more of the following (more than one for example, if the device is both a bridge and a router):

- Repeater
- Bridge
- WLAN access point
- Router
- Telephone
- DOCSIS cable device
- Station only (devices that implement end station capability)
- Other

System capabilities for Brocade devices are based on the type of software image in use (e.g., Layer 2 switch or Layer 3 router). The enabled capabilities will be the same as the available capabilities, except that when using a router image (base or full Layer 3), if the global route-only feature is turned on, the bridge capability will not be included, since no bridging takes place.

By default, the system capabilities are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise system-capabilities ports e 2/4 to 2/12
```

The system capabilities will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
System capabilities :   bridge
Enabled capabilities:   bridge
```

Syntax: [no] lldp advertise system-capabilities ports ethernet <port-list> | all

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System description

The system description is the network entity, which can include information such as the product name or model number, the version of the system hardware type, the software operating system level, and the networking software version. The information corresponds to the sysDescr MIB object in MIB-II.

To advertise the system description, enter a command such as the following.

```
Brocade(config)#lldp advertise system-description ports e 2/4 to 2/12
```

The system description will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ System description : "Brocade Communications, Inc., FESX424-PREM-PoE,
IronWare Version 04.0.00b256T3e1 Compiled on Sep 04 2007 at 0\
3:54:29 labeled as SXS04000b256"
```

NOTE

The contents of the show command output will vary depending on which TLVs are configured to be advertised.

Syntax: [no] lldp advertise system-description ports ethernet <port-list> | all

For *port-list* , specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System name

The system name is the system administratively assigned name, taken from the sysName MIB object in MIB-II. The sysName MIB object corresponds to the name defined with the CLI command **hostname**.

By default, the system name is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise system-name ports e 2/4 to 2/12
```

The system name will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
System name: "FESX424_POE"
```

Syntax: [no] lldp advertise system-name ports ethernet <port-list> | all

For *port-list* , specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.1 capabilities

Except for the VLAN name, the Brocade device will advertise the following 802.1 attributes when LLDP is enabled on a global basis:

- VLAN name (not automatically advertised)
- Untagged VLAN ID

VLAN name

The VLAN name TLV contains the name and VLAN ID of a VLAN configured on a port. An LLDPDU may include multiple instances of this TLV, each for a different VLAN.

To advertise the VLAN name, enter a command such as the following.

```
Brocade(config)#lldp advertise vlan-name vlan 99 ports e 2/4 to 2/12
```

The VLAN name will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
VLAN name (VLAN 99): "Voice-VLAN-99"
```

Syntax: [no] **lldp advertise vlan-name vlan** <vlan ID> **ports ethernet** <port-list> | **all**

For <vlan ID>, enter the VLAN ID to advertise.

For <port-list>, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Untagged VLAN ID

The port VLAN ID TLV advertises the Port VLAN Identifier (PVID) that will be associated with untagged or priority-tagged frames. If the port is not an untagged member of any VLAN (i.e., the port is strictly a tagged port), the value zero will indicate that.

By default, the port VLAN ID is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise port-vlan-id ports e 2/4 to 2/12
```

The untagged VLAN ID will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Port VLAN ID: 99
```

Syntax: [no] lldp advertise port-vlan-id ports ethernet <port-list> | all

For *port-list* , specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.3 capabilities

Except for Power-via-MDI information, the Brocade device will advertise the following 802.3 attributes when LLDP is enabled on a global basis:

- Link aggregation information
- MAC/PHY configuration and status
- Maximum frame size
- Power-via-MDI information (not automatically advertised)

Link aggregation TLV

The **link-aggregation** time, length, value (TLV) indicates the following:

- Whether the link is capable of being aggregated
- Whether the link is currently aggregated
- The primary trunk port

Brocade devices advertise link aggregation information about standard link aggregation (LACP) as well as static trunk configuration.

By default, link-aggregation information is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise link-aggregation ports e 2/12
```

Syntax: [no] lldp advertise link-aggregation ports ethernet <port-list> | all

The link aggregation advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Link aggregation: not capable
```

For *port-list* , specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

MAC and PHY configuration status

The MAC and PHY configuration and status TLV includes the following information:

- Auto-negotiation capability and status
- Speed and duplex mode
- Flow control capabilities for auto-negotiation
- Port speed down-shift and maximum port speed advertisement
- If applicable, indicates if the above settings are the result of auto-negotiation during link initiation or of a manual set override action

The advertisement reflects the effects of the following CLI commands:

- `speed-duplex`
- `flow-control`
- `gig-default`
- `link-config`

By default, the MAC/PHY configuration and status information are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise mac-phy-config-status ports e 2/4 to 2/12
```

The MAC/PHY configuration advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ 802.3 MAC/PHY      : auto-negotiation enabled
  Advertised capabilities: 10baseT-HD, 10baseT-FD, 100baseTX-HD,
100baseTX-FD,
  fdxSPause, fdxBPause, 1000baseT-HD, 1000baseT-FD
  Operational MAU type: 100BaseTX-FD
```

Syntax: `[no] lldp advertise mac-phy-config-status ports ethernet <port-list> | all`

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Maximum frame size

The maximum frame size TLV provides the maximum 802.3 frame size capability of the port. This value is expressed in octets and includes the four-octet Frame Check Sequence (FCS). The default maximum frame size is 1522. The advertised value may change depending on whether the **aggregated-vlan** or **jumbo** CLI commands are in effect.

NOTE

On 48GC modules in non-jumbo mode, the maximum size of ping packets is 1486 bytes and the maximum frame size of tagged traffic is no larger than 1581 bytes.

By default, the maximum frame size is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise max-frame-size ports e 2/4 to 2/12
```

The maximum frame size advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Maximum frame size: 1522 octets
```

Syntax: **[no] lldp advertise max-frame-size ports ethernet <port-list> | all**

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Power-via-MDI

The power-via-MDI TLV provides general information about Power over Ethernet (POE) capabilities and status of the port. It indicates the following:

- POE capability (supported or not supported)
- POE status (enabled or disabled)
- Power Sourcing Equipment (PSE) power pair – indicates which pair of wires is in use and whether the pair selection can be controlled. The Brocade implementation always uses pair A, and cannot be controlled.

- Power class – Indicates the range of power that the connected powered device has negotiated or requested.

NOTE

The power-via-MDI TLV described in this section applies to LLDP. There is also a power-via-MDI TLV for LLDP-MED devices, which provides extensive POE information. Refer to [“Extended power-via-MDI information”](#) on page 486.

To advertise the power-via-MDI information, enter a command such as the following.

```
Brocade(config)#lldp advertise power-via-mdi ports e 2/4 to 2/12
```

The power-via-MDI advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ 802.3 Power via MDI: PSE port, power enabled, class 0
Power Pair          : A (not controllable)
```

Syntax: [no] lldp advertise power-via-mdi ports ethernet <port-list> | all

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

LLDP-MED configuration

This section provides the details for configuring LLDP-MED.

[Table 84](#) lists the global and interface-level tasks and the default behavior/value for each task.

TABLE 84 LLDP-MED configuration tasks and default behavior / value

Task	Default behavior / value
Global CONFIG-level tasks	
Enabling LLDP-MED on a global basis	Disabled
Enabling SNMP notifications and Syslog messages for LLDP-MED topology change	Disabled

TABLE 84 LLDP-MED configuration tasks and default behavior / value

Task	Default behavior / value
Changing the Fast Start Repeat Count	The system automatically sets the fast start repeat count to 3 when a Network Connectivity Device receives an LLDP packet from an Endpoint that is newly connected to the network. NOTE: The LLDP-MED fast start mechanism is only intended to run on links between Network Connectivity devices and Endpoint devices. It does not apply to links between LAN infrastructure elements, including between Network Connectivity devices, or to other types of links.
Interface-level tasks	
Defining a location ID	Not configured
Defining a network policy	Not configured

Enabling LLDP-MED

When LLDP is enabled globally, LLDP-MED is enabled if the LLDP-MED capabilities TLV is also enabled. By default, the LLDP-MED capabilities TLV is automatically enabled. To enable LLDP, refer to [“Enabling and disabling LLDP”](#) on page 459.

NOTE

LLDP-MED is not enabled on ports where the LLDP operating mode is receive only or transmit only. LLDP-MED is enabled on ports that are configured to both receive and transmit LLDP packets and have the LLDP-MED capabilities TLV enabled.

Enabling SNMP notifications and Syslog messages for LLDP-MED topology changes

SNMP notifications and Syslog messages for LLDP-MED provide management applications with information related to topology changes. For example, SNMP notifications can alert the system whenever a remote Endpoint device is connected to or removed from a local port. SNMP notifications identify the local port where the topology change occurred, as well as the device capability of the remote Endpoint device that was connected to or removed from the port.

When you enable LLDP-MED SNMP notifications, corresponding Syslog messages are enabled as well. When you enable LLDP-MED SNMP notifications, the device will send traps and Syslog messages when an LLDP-MED Endpoint neighbor entry is added or removed.

SNMP notifications and corresponding Syslog messages are disabled by default. To enable them, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp enable snmp med-topo-change-notifications ports ethernet 4/4 to 4/6
```

Syntax: no lldp enable snmp med-topo-change-notifications ports ethernet <port-list> | all

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*

- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Changing the fast start repeat count

The fast start feature enables a Network Connectivity Device to initially advertise itself at a faster rate for a limited time when an LLDP-MED Endpoint has been newly detected or connected to the network. This feature is important within a VoIP network, for example, where rapid availability is crucial for applications such as emergency call service location (E911).

The fast start timer starts when a Network Connectivity Device receives the first LLDP frame from a newly detected Endpoint.

The **LLDP-MED fast start repeat count** specifies the number of LLDP packets that will be sent during the LLDP-MED fast start period. By default, the device will send three packets at one-second intervals. If desired, you can change the number of packets the device will send per second, up to a maximum of 10.

NOTE

The LLDP-MED fast start mechanism is only intended to run on links between Network Connectivity devices and Endpoint devices. It does not apply to links between LAN infrastructure elements, including between Network Connectivity devices, or to other types of links.

To change the LLDP-MED fast start repeat count, enter commands such as the following.

```
Brocade(config)#lldp med fast-start-repeat-count 5
```

The above command causes the device to send five LLDP packets during the LLDP-MED fast start period.

Syntax: [no] **lldp med fast-start-repeat-count** <value>

where value is a number from 1 to 10, which specifies the number of packets that will be sent during the LLDP-MED fast start period. The default is 3.

Defining a location id

The LLDP-MED Location Identification extension enables the Brocade device to set the physical location that an attached Class III Endpoint will use for location-based applications. This feature is important for applications such as IP telephony, for example, where emergency responders need to quickly determine the physical location of a user in North America that has just dialed 911.

For each port, you can define one or more of the following location ID formats:

- Geographic location (coordinate-based)
- Civic address
- Emergency Call Services (ECS) Emergency Location Identification Number (ELIN)

The above location ID formats are defined in the following sections.

Coordinate-based location

Coordinate-based location is based on the IETF RFC 3825 [6] standard, which specifies a Dynamic Host Configuration Protocol (DHCP) option for the coordinate-based geographic location of a client.

When you configure an Endpoint location information using the coordinate-based location, you specify the latitude, longitude, and altitude, along with **resolution indicators** (a measure of the accuracy of the coordinates), and the reference **datum** (the map used for the given coordinates).

To configure a coordinate-based location for an Endpoint device, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp med location-id coordinate-based latitude
-78.303 resolution 20 longitude 34.27 resolution 18 altitude meters 50 resolution
16 wgs84
```

Syntax: [no] lldp med location-id coordinate-based

latitude <degrees> **resolution** <bits>

longitude <degrees> **resolution** <bits>

altitude floors <number> **resolution** <bits> | **meters** <number> **resolution** <bits>
<datum>

latitude <degrees> is the angular distance north or south from the earth equator measured through 90 degrees. Positive numbers indicate a location north of the equator and negative numbers indicate a location south of the equator.

resolution <bits> specifies the precision of the value given for latitude. A smaller value increases the area within which the device is located. For latitude, enter a number between 1 and 34.

longitude <degrees> is the angular distance from the intersection of the zero meridian. Positive values indicate a location east of the prime meridian and negative numbers indicate a location west of the prime meridian.

resolution <bits> specifies the precision of the value given for longitude. A smaller value increases the area within which the device is located. For longitude resolution, enter a number between 1 and 34.

altitude floors <number> is the vertical elevation of a building above the ground, where 0 represents the floor level associated with the ground level at the main entrance and larger values represent floors that are above (higher in altitude) floors with lower values. For example, 2 for the 2nd floor. Sub-floors can be represented by non-integer values. For example, a mezzanine between floor 1 and floor 2 could be represented as 1.1. Similarly, the mezzanines between floor 4 and floor 5 could be represented as 4.1 and 4.2 respectively. Floors located below ground level could be represented by negative values.

resolution <bits> specifies the precision of the value given for altitude. A smaller value increases the area within which the device is located. For floors resolution, enter the value 0 if the floor is unknown, or 30 if a valid floor is being specified.

altitude meters <number> is the vertical elevation in number of meters, as opposed to floors.

resolution <bits> specifies the precision of the value given for altitude. A smaller value increases the area within which the device is located. For meters resolution, enter a value from 0 to 30.

<Datum> is the map used as the basis for calculating the location. Specify one of the following:

- **wgs84** – (geographical 3D) – World Geodesic System 1984, CRS Code 4327, Prime Meridian Name: Greenwich
- **nad83-navd88** – North American Datum 1983, CRS Code 4269, Prime Meridian Name: Greenwich; The associated vertical datum is the North American Vertical Datum of 1988 (NAVD88). Use this datum when referencing locations on land. If land is near tidal water, use nad83-mllw (below).
- **nad83-mllw** – North American Datum 1983, CRS Code 4269, Prime Meridian Name: Greenwich; The associated vertical datum is mean lower low water (MLLW). Use this datum when referencing locations on water, sea, or ocean.

Example coordinate-based location configuration

The following shows an example coordinate-based location configuration for the Sears Tower, at the following location.

103rd Floor
233 South Wacker Drive
Chicago, IL 60606

```
Brocade(config)#lldp med location-id coordinate-based latitude 41.87884
resolution 18 longitude 87.63602 resolution 18 altitude floors 103 resolution 30
wgs84
```

The above configuration shows the following:

- Latitude is 41.87884 degrees north (or 41.87884 degrees).
- Longitude is 87.63602 degrees west (or 87.63602 degrees).
- The latitude and longitude resolution of 18 describes a geo-location area that is latitude 41.8769531 to latitude 41.8789062 and extends from -87.6367188 to -87.6347657 degrees longitude. This is an area of approximately 373412 square feet (713.3 ft. x 523.5 ft.).
- The location is inside a structure, on the 103rd floor.
- The WGS 84 map was used as the basis for calculating the location.

Example coordinate-based location advertisement

The coordinate-based location advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ MED Location ID
  Data Format: Coordinate-based
  Latitude Resolution : 20 bits
  Latitude Value      : -78.303 degrees
  Longitude Resolution : 18 bits
  Longitude Value     : 34.27 degrees
  Altitude Resolution : 16 bits
  Altitude Value      : 50. meters
  Datum               : WGS 84
```

Configuring civic address location

When you configure a media Endpoint location using the address-based location, you specify the location the entry refers to, the country code, and the elements that describe the civic or postal address.

To configure a civic address-based location for LLDP-MED, enter commands such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp med location-id civic-address refers-to client country US
elem 1 CA elem 3 "Santa Clara" elem 6 "4980 Great America Pkwy" elem 24 95054 elem
27 5 elem 28 551 elem 29 office elem 23 "John Doe"
```

Syntax: `[no] lldp med location-id civic-address refers-to <elem> country <country code> elem <CA type> <value> [elem <CA type> <value>] [elem <CA type> <value>]...`

refers-to <elem> describes the location that the entry refers to. Specify one of the following:

- client
- dhcp-server
- network-element

where **dhcp-server** or **network-element** should only be used if it is known that the Endpoint is in close physical proximity to the DHCP server or network element.

<country code> is the two-letter ISO 3166 country code in capital ASCII letters.

Example

- CA – Canada
- DE – Germany
- JP – Japan
- KR – Korea
- US – United States

<CA type> is a value from 0 – 255, that describes the civic address element. For example, a CA type of 24 specifies a postal or zip code. Valid elements and their types are listed in [Table 85](#).

<value> is the actual value of the elem <CA type>, above. For example, 95123 for the postal or zip code. Acceptable values are listed in [Table 85](#), below.

NOTE

If the value of an element contains one or more spaces, use double quotation marks (") at the beginning and end of the string. For example, `elem 3 "Santa Clara"`.

TABLE 85 Elements used with civic address

Civic Address (CA) type	Description	Acceptable values / examples
0	Language	The ISO 639 language code used for presenting the address information.
1	National subdivisions (state, canton, region, province, or prefecture)	Examples: Canada – Province Germany – State Japan – Metropolis Korea – Province United States – State
2	County, parish, gun (JP), or district (IN)	Examples: Canada – County Germany – County Japan – City or rural area Korea – County United States – County
3	City, township, or shi (JP)	Examples: Canada – City or town Germany – City Japan – Ward or village Korea – City or village United States – City or town
4	City division, borough, city district, ward, or chou (JP)	Examples: Canada – N/A Germany – District Japan – Town Korea – Urban district United States – N/A
5	Neighborhood or block	Examples: Canada – N/A Germany – N/A Japan – City district Korea – Neighborhood United States – N/A
6	Street	Examples: Canada – Street Germany – Street Japan – Block Korea – Street United States – Street
16	Leading street direction	N (north), E (east), S (south), W (west), NE, NW, SE, SW
17	Trailing street suffix	N (north), E (east), S (south), W (west), NE, NW, SE, SW
18	Street suffix	Acceptable values for the United States are listed in the United States Postal Service Publication 28 [18], Appendix C. Example: Ave, Place
19	House number	The house number (street address) Example: 1234

TABLE 85 Elements used with civic address (Continued)

Civic Address (CA) type	Description	Acceptable values / examples
20	House number suffix	A modifier to the house number. It does not include parts of the house number. Example: A, 1/2
21	Landmark or vanity address	A string name for a location. It conveys a common local designation of a structure, a group of buildings, or a place that helps to locate the place. Example: UC Berkeley
22	Additional location information	An unstructured string name that conveys additional information about the location. Example: west wing
23	Name (residence and office occupant)	Identifies the person or organization associated with the address. Example: Textures Beauty Salon
24	Postal / zip code	The valid postal / zip code for the address. Example: 95054-1234
25	Building (structure)	The name of a single building if the street address includes more than one building or if the building name is helpful in identifying the location. Example: Law Library
26	Unit (apartment, suite)	The name or number of a part of a structure where there are separate administrative units, owners, or tenants, such as separate companies or families who occupy that structure. Common examples include suite or apartment designations. Example: Apt 27
27	Floor	Example: 4
28	Room number	The smallest identifiable subdivision of a structure. Example: 7A
29	Placetype	The type of place described by the civic coordinates. For example, a home, office, street, or other public space. Example: Office
30	Postal community name	When the postal community name is defined, the civic community name (typically CA type 3) is replaced by this value. Example: Alviso
31	Post office box (P.O. box)	When a P.O. box is defined, the street address components (CA types 6, 16, 17, 18, 19, and 20) are replaced with this value. Example: P.O. Box 1234
32	Additional code	An additional country-specific code that identifies the location. For example, for Japan, this is the Japan Industry Standard (JIS) address code. The JIS address code provides a unique address inside of Japan, down to the level of indicating the floor of the building.

TABLE 85 Elements used with civic address (Continued)

Civic Address (CA) type	Description	Acceptable values / examples
128	Script	The script (from ISO 15924 [14]) used to present the address information. Example: Latn NOTE: If not manually configured, the system assigns the default value Latn
255	Reserved	

Example civic address location advertisement

The Civic address location advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ MED Location ID
  Data Format: Civic Address
  Location of: Client
  Country    : "US"
  CA Type    : 1
  CA Value   : "CA"
  CA Type    : 3
  CA Value   : "Santa Clara"
  CA Type    : 6
  CA Value   : "4980 Great America Pkwy."
  CA Type    : 24
  CA Value   : "95054"
  CA Type    : 27
  CA Value   : "5"
  CA Type    : 28
  CA Value   : "551"
  CA Type    : 29
  CA Value   : "office"
  CA Type    : 23
  CA Value   : "John Doe"
```

Configuring emergency call service

The Emergency Call Service (ECS) location is used specifically for Emergency Call Services applications.

When you configure a media Endpoint location using the emergency call services location, you specify the Emergency Location Identification Number (ELIN) from the North America Numbering Plan format, supplied to the Public Safety Answering Point (PSAP) for ECS purposes.

To configure an ECS-based location for LLDP-MED, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)#lldp med location-id ecs-elin 4082071700
```

Syntax: [no] lldp med location-id ecs-elin <number> ports ethernet <port-list> | all

<number> is a number from 10 to 25 digits in length.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Example ECS ELIN location advertisements

The ECS ELIN location advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ MED Location ID
  Data Format: ECS ELIN
  Value      : 4082071700
```

Defining an LLDP-MED network policy

An LLDP-MED network policy defines an Endpoint VLAN configuration (VLAN type and VLAN ID) and associated Layer 2 and Layer 3 priorities that apply to a specific set of applications on a port.

NOTE

This feature applies to applications that have specific real-time network policy requirements, such as interactive voice or video services. It is not intended to run on links other than between Network Connectivity devices and Endpoints, and therefore does not advertise the multitude of network policies that frequently run on an aggregated link.

To define an LLDP-MED network policy for an Endpoint, enter a command such as the following.

```
Brocade(config)#lldp med network-policy application voice tagged vlan 99 priority
3 dscp 22 port e 2/6
```

The network policy advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ MED Network Policy
  Application Type : Voice
  Policy Flags    : Known Policy, Tagged
  VLAN ID        : 99
  L2 Priority     : 3
  DSCP Value     : 22
```

NOTE

Endpoints will advertise a policy as “unknown” in the **show lldp neighbor detail** command output, if it is a policy that is required by the Endpoint and the Endpoint has not yet received it.

LLDP-MED network policy configuration syntax

The CLI syntax for defining an LLDP-MED network policy differs for tagged, untagged, and priority tagged traffic. Refer to the appropriate syntax, below.

For tagged traffic

Syntax: `[no] lldp med network-policy application <application type> tagged vlan <vlan ID> priority <0 – 7> dscp <0 – 63> ports ethernet <port-list> | all`

For untagged traffic

Syntax: `[no] lldp med network-policy application <application type> untagged dscp <0 – 63> ports ethernet <port-list> | all`

For priority-tagged traffic

Syntax: `[no] lldp med network-policy application <application type> priority-tagged priority <0 – 7> dscp <0 – 63> ports ethernet <port-list> | all`

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

<application type> indicates the primary function of the applications defined by this network policy. Application type can be one of the following:

- **guest-voice** – Limited voice service for guest users and visitors with their own IP telephony handsets or similar devices that support interactive voice services.
- **guest-voice-signaling** – Limited voice service for use in network topologies that require a different policy for guest voice signaling than for guest voice media.
- **softphone-voice** – Softphone voice service for use with multi-media applications that work in association with VoIP technology, enabling phone calls direct from a PC or laptop. Softphones do not usually support multiple VLANs, and are typically configured to use an untagged VLAN or a single tagged data-specific VLAN. Note that when a network policy is defined for use with an untagged VLAN, the Layer 2 priority field is ignored and only the DSCP value is relevant.
- **streaming-video** – Applies to broadcast- or multicast-based video content distribution and similar applications that support streaming video services requiring specific network policy treatment. Video applications that rely on TCP without buffering would not be an intended use of this application type.
- **video-conferencing** – Applies to dedicated video conferencing equipment and similar devices that support real-time interactive video/audio services.
- **video-signaling** – For use in network topologies that require a separate policy for video signaling than for video media. Note that this application type should not be advertised if all the same network policies apply as those advertised in the video conferencing policy TLV.
- **voice** – For use by dedicated IP telephony handsets and similar devices that support interactive voice services.

- **voice-signaling** – For use in network topologies that require a different policy for voice signaling than for voice media. Note that this application type should not be advertised if all the same network policies apply as those advertised in the voice policy TLV.
- **tagged vlan** <vlan id> specifies the tagged VLAN that the specified application type will use.
- **untagged** indicates that the device is using an untagged frame format.
- **priority-tagged** indicates that the device uses priority-tagged frames. In this case, the device uses the default VLAN (PVID) of the ingress port.
- **priority** <0 – 7> indicates the Layer 2 priority value to be used for the specified application type. Enter 0 to use the default priority.
- **dscp** <0 – 63> specifies the Layer 3 Differentiated Service codepoint priority value to be used for the specified application type. Enter 0 to use the default priority.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

LLDP-MED attributes advertised by the Brocade device

LLDP-MED attributes are only advertised on a port if LLDP-MED is enabled (which is done by enabling the LLDP-MED capabilities TLV), the port operating mode is *receive* and *transmit* (the default), and the port has received an LLDP-MED advertisement from an Endpoint. By default, the Brocade device will automatically advertise the following LLDP-MED attributes when the above criteria are met:

- LLDP-MED capabilities
- Location ID
- Network policy
- Power-via-MDI information

NOTE

Although the Location ID and Network policy attributes are automatically advertised, they will have no effect until they are actually defined.

LLDP-MED capabilities

When enabled, LLDP-MED is enabled, and the LLDP-MED capabilities TLV is sent whenever any other LLDP-MED TLV is sent. When disabled, LLDP-MED is disabled and no LLDP-MED TLVs are sent.

The LLDP-MED capabilities advertisement includes the following information:

- The supported LLDP-MED TLVs
- The device type (Network Connectivity device or Endpoint (Class 1, 2, or 3))

By default, LLDP-MED information is automatically advertised when LLDP-MED is enabled. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise med-capabilities ports e 2/4 to 2/12
```

NOTE

Disabling the LLDP-MED capabilities TLV disables LLDP-MED.

To re-enable the LLDP-MED Capabilities TLV (and LLDP-MED) after it has been disabled, enter a command such as the following.

```
Brocade(config)#lldp advertise med-capabilities ports e 2/4 to 2/12
```

The LLDP-MED capabilities advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
      + MED capabilities: capabilities, networkPolicy, location, extendedPSE
MED device type : Network Connectivity
```

Syntax: [no] lldp advertise med-capabilities ports ethernet <port-list> | all

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Extended power-via-MDI information

The extended Power-via-MDI TLV enables advanced power management between LLDP-MED Endpoints and Network Connectivity Devices. This TLV provides significantly more information than the 802.1AB Power-via-MDI TLV referenced in [“Power-via-MDI”](#) on page 473. For example, this TLV enables an Endpoint to communicate a more precise required power level, thereby enabling the device to allocate less power to the Endpoint, while making more power available to other ports.

The LLDP-MED Power-via-MDI TLV advertises an Endpoint IEEE 802.3af power-related information, including the following:

- **Power type** – indicates whether the LLDP-MED device transmitting the LLPDU is a power sourcing device or a powered device:

- **Power sourcing device/equipment (PSE)** – This is the source of the power, or the device that integrates the power onto the network. Power sourcing devices/equipment have embedded POE technology. In this case, the power sourcing device is the Brocade POE device.
- **Powered device (PD)** – This is the Ethernet device that requires power and is situated on the other end of the cable opposite the power sourcing device.
- **Power source** – The power source being utilized by a PSE or PD, for example, primary power source, backup power source, or unknown.

For Endpoint devices, the power source information indicates the power capability of the Network Connectivity Device it is attached to. When the Network Connectivity device advertises that it is using its primary power source, the Endpoint should expect to have uninterrupted access to its available power. Likewise, if the Network Connectivity device advertises that it is using backup power, the Endpoint should not expect continuous power. The Endpoint may additionally choose to power down non-essential subsystems or to conserve power as long as the PSE is advertising that it is operating on backup power.

NOTE

Brocade devices always advertise the power source as “unknown”.

- **Power priority** – The in-line power priority level for the PSE or PD:
 - 3 – low
 - 2 – high
 - 1 – critical
 - unknown
- **Power level** – The total power, in tenths of watts, required by a PD from a PSE, or the total power a PSE is capable of sourcing over a maximum length cable based on its current configuration.

If the exact power is not known for a PSE or PD, it will advertise the power level associated with its 802.3af power class (listed in [Table 86](#)).

TABLE 86 802.3af power classes

Power class	Minimum power level output at the PSE	Maximum power levels at the PD
0	15.4 watts	0.44 – 12.95 watts
1	4.0 watts	0.44 – 3.84 watts
2	7.0 watts	3.84 – 6.49 watts
3	15.4 watts	6.49 – 12.95 watts

For a PD (Endpoint device), the power level represents the maximum power it can consume during normal operations in its current configuration, even if its actual power draw at that instance is less than the advertised power draw.

For a PSE (Network Connectivity device), the power level represents the amount of power that is available on the port at the time. If the PSE is operating in reduced power (i.e., it is using backup power), the reduced power capacity is advertised as long as the condition persists.

By default, LLDP-MED power-via-MDI information is automatically advertised when LLDP-MED is enabled, the port is a POE port, and POE is enabled on the port. To disable this advertisement, enter a command such as the following.

```
Brocade(config)#no lldp advertise med-power-via-mdi ports e 2/4 to 2/12
```

The LLDP-MED power-via-MDI advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ MED Extended Power via MDI
  Power Type      : PSE device
  Power Source    : Unknown Power Source
  Power Priority   : Low (3)
  Power Value     : 6.5 watts (PSE equivalent: 7005 mWatts)
```

Syntax: [no] lldp advertise med-power-via-mdi ports ethernet <port-list> | all

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Displaying LLDP statistics and configuration settings

You can use the following CLI **show** commands to display information about LLDP settings and statistics:

- **show lldp** – Displays a summary of the LLDP configuration settings.
- **show lldp statistics** – Displays LLDP global and per-port statistics.
- **show lldp neighbors** – Displays a list of the current LLDP neighbors.
- **show lldp neighbors detail** – Displays the details of the latest advertisements received from LLDP neighbors.
- **show lldp local-info** – Displays the details of the LLDP advertisements that will be transmitted on each port.

This above **show** commands are described in this section.

LLDP configuration summary

To display a summary of the LLDP configuration settings on the device, enter the **show lldp** command at any level of the CLI.

The following shows an example report.

LLDP-MED attributes advertised by the Brocade device

```
Brocade#show lldp
LLDP transmit interval      : 10 seconds
LLDP transmit hold multiplier : 4 (transmit TTL: 40 seconds)
LLDP transmit delay        : 1 seconds
LLDP SNMP notification interval : 5 seconds
LLDP reinitialize delay    : 1 seconds
LLDP-MED fast start repeat count : 3

LLDP maximum neighbors      : 392
LLDP maximum neighbors per port : 4
```

Syntax: show lldp

The following table describes the information displayed by the **show lldp statistics** command.

Field	Description
LLDP transmit interval	The number of seconds between regular LLDP packet transmissions.
LLDP transmit hold multiplier	The multiplier used to compute the actual time-to-live (TTL) value of an LLDP advertisement. The TTL value is the transmit interval multiplied by the transmit hold multiplier.
LLDP transmit delay	The number of seconds the LLDP agent will wait after transmitting an LLDP frame and before transmitting another LLDP frame.
LLDP SNMP notification interval	The number of seconds between transmission of SNMP LLDP traps (lldpRemTablesChange) and SNMP LLDP-MED traps (lldpXMedTopologyChangeDetected).
LLDP reinitialize delay	The minimum number of seconds the device will wait from when LLDP is disabled on a port, until a request to re-enable LLDP on that port will be honored.
LLDP-MED fast start repeat count	The number of seconds between LLDP frame transmissions when an LLDP-MED Endpoint is newly detected.
LLDP maximum neighbors	The maximum number of LLDP neighbors for which LLDP data will be retained, per device.
LLDP maximum neighbors per port	The maximum number of LLDP neighbors for which LLDP data will be retained, per port.

Displaying LLDP statistics

The **show lldp statistics** command displays an overview of LLDP neighbor detection on the device, as well as packet counters and protocol statistics. The statistics are displayed on a global basis.

The following shows an example report.

LLDP-MED attributes advertised by the Brocade device

```
Brocade#show lldp statistics
Last neighbor change time: 23 hours 50 minutes 40 seconds ago

Neighbor entries added      : 14
Neighbor entries deleted    : 5
Neighbor entries aged out   : 4
Neighbor advertisements dropped : 0
```

Port	Tx Pkts	Rx Pkts	Rx Pkts	Rx Pkts	Rx TLVs	Rx TLVs	Neighbors
	Total	Total	w/Errors	Discarded	Unrecognz	Discarded	Aged Out
1	60963	75179	0	0	0	0	4
2	0	0	0	0	0	0	0
3	60963	60963	0	0	0	0	0
4	60963	121925	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	60974	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0

Syntax: show lldp statistics

NOTE

You can reset LLDP statistics using the CLI command **clear LLDP statistics**. Refer to [“Resetting LLDP statistics”](#) on page 495.

The following table describes the information displayed by the **show lldp statistics** command.

Field	Description
Last neighbor change time	The elapsed time (in hours, minutes, and seconds) since a neighbor last advertised information. For example, the elapsed time since a neighbor was last added, deleted, or its advertised information changed.
Neighbor entries added	The number of new LLDP neighbors detected since the last reboot or since the last time the clear lldp statistics all command was issued.
Neighbor entries deleted	The number of LLDP neighbors deleted since the last reboot or since the last time the clear lldp statistics all command was issued.
Neighbor entries aged out	The number of LLDP neighbors dropped on all ports after the time-to-live expired. Note that LLDP entries age out naturally when a port cable or module is disconnected or when a port becomes disabled. However, if a disabled port is re-enabled, the system will delete the old LLDP entries.
Neighbor advertisements dropped	The number of valid LLDP neighbors the device detected, but could not add. This can occur, for example, when a new neighbor is detected and the device is already supporting the maximum number of neighbors possible. This can also occur when an LLDPDU is missing a mandatory TLV or is not formatted correctly.
Port	The local port number.
Tx Pkts Total	The number of LLDP packets the port transmitted.
Rx Pkts Total	The number of LLDP packets the port received.

Field	Description
Rx Pkts w/Errors	The number of LLDP packets the port received that have one or more detectable errors.
Rx Pkts Discarded	The number of LLDP packets the port received then discarded.
Rx TLVs Unrecognz	The number of TLVs the port received that were not recognized by the LLDP local agent. Unrecognized TLVs are retained by the system and can be viewed in the output of the show LLDP neighbors detail command or retrieved through SNMP.
Rx TLVs Discarded	The number of TLVs the port received then discarded.
Neighbors Aged Out	The number of times a neighbor information was deleted because its TTL timer expired.

Displaying LLDP neighbors

The **show lldp neighbors** command displays a list of the current LLDP neighbors per port.

The following shows an example report.

```
Brocade#show lldp neighbors
```

Lcl Port	Chassis ID	Port ID	Port Description	System Name
1	0004.1234.0fc0	0004.1234.0fc0	GigabitEthernet9/1	FastIron Supe~
1	00e0.5201.4000	00e0.5201.4000	GigabitEthernet0/1/1	FGS624XGP Swi~
3	00e0.5211.0200	00e0.5211.0203	GigabitEthernet4	FESX424+2XG S~
4	00e0.5211.0200	00e0.5211.0202	GigabitEthernet3	FESX424+2XG S~
4	00e0.5211.0200	00e0.5211.0210	GigabitEthernet17	FESX424+2XG S~
15	00e0.5211.0200	00e0.5211.020f	GigabitEthernet16	FESX424+2XG S~
16	00e0.5211.0200	00e0.5211.020e	GigabitEthernet15	FESX424+2XG S~
17	00e0.5211.0200	00e0.5211.0211	GigabitEthernet18	FESX424+2XG S~
18	00e0.5211.0200	00e0.5211.0210	GigabitEthernet17	FESX424+2XG S~

Syntax: show lldp neighbors

The following table describes the information displayed by the **show lldp neighbors** command.

Field	Description
Lcl Port	The local LLDP port number.
Chassis ID	The identifier for the chassis. Brocade devices use the base MAC address of the device as the Chassis ID.
Port ID	The identifier for the port. Brocade devices use the permanent MAC address associated with the port as the port ID.
Port Description	The description for the port. Brocade devices use the ifDescr MIB object from MIB-II as the port description.
System Name	The administratively-assigned name for the system. Brocade devices use the sysName MIB object from MIB-II, which corresponds to the CLI hostname command setting. NOTE: A tilde (~) at the end of a line indicates that the value in the field is too long to display in full and is truncated.

Displaying LLDP neighbors detail

The **show lldp neighbors detail** command displays the LLDP advertisements received from LLDP neighbors.

The following shows an example **show lldp neighbors detail** report.

NOTE

The **show lldp neighbors detail** output will vary depending on the data received. Also, values that are not recognized or do not have a recognizable format, may be displayed in hexadecimal binary form.

```
Brocade#show lldp neighbors detail ports e 1/9
Local port: 1/9
Neighbor: 0800.0f18.cc03, TTL 101 seconds
+ Chassis ID (network address): 10.43.39.151
+ Port ID (MAC address): 0800.0f18.cc03
+ Time to live: 120 seconds
+ Port description      : "LAN port"
+ System name          : "regDN 1015,MITEL 5235 DM"
+ System description   : "regDN 1015,MITEL 5235 DM,h/w rev 2,ASIC rev 1,f/w\
                        Boot 02.01.00.11,f/w Main 02.01.00.11"
+ System capabilities : bridge, telephone
  Enabled capabilities: bridge, telephone
+ Management address (IPv4): 10.43.39.151
+ 802.3 MAC/PHY       : auto-negotiation enabled
  Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                        100BaseTX-FD
  Operational MAU type   : 100BaseTX-FD
+ MED capabilities: capabilities, networkPolicy, extendedPD
  MED device type : Endpoint Class III
+ MED Network Policy
  Application Type : Voice
  Policy Flags     : Known Policy, Tagged
  VLAN ID         : 300
  L2 Priority      : 7
  DSCP Value       : 7
+ MED Extended Power via MDI
  Power Type       : PD device
  Power Source     : Unknown Power Source
  Power Priority    : High (2)
  Power Value      : 6.2 watts (PSE equivalent: 6656 mWatts)
+ MED Hardware revision : "PCB Version: 2"
+ MED Firmware revision : "Boot 02.01.00.11"
+ MED Software revision : "Main 02.01.00.11"
+ MED Serial number     : ""
+ MED Manufacturer      : "Mitel Corporation"
+ MED Model name        : "MITEL 5235 DM"
+ MED Asset ID          : ""
```

A backslash (\) at the end of a line indicates that the text continues on the next line.

Except for the following field, the fields in the above output are described in the individual TLV advertisement sections in this chapter.

Field	Description
Neighbor	The source MAC address from which the packet was received, and the remaining TTL for the neighbor entry.

Syntax: `show lldp neighbors detail [ports ethernet <port-list> | all]`

If you do not specify any ports or use the keyword **all**, by default, the report will show the LLDP neighbor details for all ports.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Displaying LLDP configuration details

The **show lldp local-info** command displays the local information advertisements (TLVs) that will be transmitted by the LLDP agent.

NOTE

The **show lldp local-info** output will vary based on LLDP configuration settings.

The following shows an example report.

```
Brocade#show lldp local-info ports e 20
Local port: 20
+ Chassis ID (MAC address): 0012.f233.e2c0
+ Port ID (MAC address): 0012.f233.e2d3
+ Time to live: 40 seconds
+ System name: "FESX424_POE"
+ Port description: "GigabitEthernet20"
+ System description : "Brocade Communications, Inc. FESX424-PREM-PoE, IronWare
V\
                                ersion 04.0.00b256T3e1 Compiled on Sep 04 2007 at 0\
                                3:54:29 labeled as SXS04000b256"
+ System capabilities : bridge
  Enabled capabilities: bridge
+ 802.3 MAC/PHY       : auto-negotiation enabled
  Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                           100BaseTX-FD, fdxSPause, fdxBPause, 1000BaseT-HD,
                           1000BaseT-FD
  Operational MAU type: 100BaseTX-FD
+ 802.3 Power via MDI: PSE port, power enabled, class 2
  Power Pair          : A (not controllable)
+ Link aggregation: not capable
+ Maximum frame size: 1522 octets
+ MED capabilities: capabilities, networkPolicy, location, extendedPSE
  MED device type : Network Connectivity
+ MED Network Policy
  Application Type  : Voice
  Policy Flags      : Known Policy, Tagged
```

LLDP-MED attributes advertised by the Brocade device

```
VLAN ID          : 99
L2 Priority       : 3
DSCP Value       : 22
+ MED Network Policy
  Application Type : Video Conferencing
  Policy Flags    : Known Policy, Tagged
  VLAN ID        : 100
  L2 Priority     : 5
  DSCP Value     : 10
+ MED Location ID
  Data Format: Coordinate-based location
  Latitude Resolution : 20 bits
  Latitude Value    : -78.303 degrees
  Longitude Resolution : 18 bits
  Longitude Value   : 34.27 degrees
  Altitude Resolution : 16 bits
  Altitude Value    : 50. meters
  Datum            : WGS 84
+ MED Location ID
  Data Format: Civic Address
  Location of: Client
  Country       : "US"
  CA Type       : 1
  CA Value      : "CA"
  CA Type       : 3
  CA Value      : "Santa Clara"
  CA Type       : 6
  CA Value      : "4980 Great America Pkwy."
  CA Type       : 24
  CA Value      : "95054"
  CA Type       : 27
  CA Value      : "5"
  CA Type       : 28
  CA Value      : "551"
  CA Type       : 29
  CA Value      : "office"
  CA Type       : 23
  CA Value      : "John Doe"
+ MED Location ID
  Data Format: ECS ELIN
  Value       : "1234567890"
+ MED Extended Power via MDI
  Power Type   : PSE device
  Power Source : Unknown Power Source
  Power Priority : Low (3)
  Power Value  : 6.5 watts (PSE equivalent: 7005 mWatts) + Port VLAN ID: 99
+ Management address (IPv4): 192.1.1.121
+ VLAN name (VLAN 99): "Voice-VLAN-99"
```

NOTE

The contents of the **show** output will vary depending on which TLVs are configured to be advertised.

A backslash (\) at the end of a line indicates that the text continues on the next line.

The fields in the above output are described in the individual TLV advertisement sections in this chapter.

Syntax: `show lldp local-info [ports ethernet <port-list> | all]`

If you do not specify any ports or use the keyword **all**, by default, the report will show the local information advertisements for all ports.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Resetting LLDP statistics

To reset LLDP statistics, enter the **clear lldp statistics** command at the Global CONFIG level of the CLI. The Brocade device will clear the global and per-port LLDP neighbor statistics on the device (refer to “[Displaying LLDP statistics](#)” on page 489).

```
Brocade#clear lldp statistics
```

Syntax: **clear lldp statistics** [**ports ethernet** <*port-list*> | **all**]

If you do not specify any ports or use the keyword **all**, by default, the system will clear lldp statistics on all ports.

For *port-list*, specify the ports in one of the following formats:

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Clearing cached LLDP neighbor information

The Brocade device clears cached LLDP neighbor information after a port becomes disabled and the LLDP neighbor information ages out. However, if a port is disabled then re-enabled before the neighbor information ages out, the device will clear the cached LLDP neighbor information when the port is re-enabled.

If desired, you can manually clear the cache. For example, to clear the cached LLDP neighbor information for port e 20, enter the following command at the Global CONFIG level of the CLI.

```
Brocade#clear lldp neighbors ports e 20
```

Syntax: **clear lldp neighbors** [**ports ethernet** <*port-list*> | **all**]

If you do not specify any ports or use the keyword **all**, by default, the system will clear the cached LLDP neighbor information for all ports.

For *port-list*, specify the ports in one of the following formats:

Clearing cached LLDP neighbor information

- FWS, FCX and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Hardware Component Monitoring

Table 87 lists the individual Brocade FastIron switches and the hardware monitoring features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images.

TABLE 87 Supported hardware monitoring features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Virtual cable testing (VCT)	Yes	Yes	Yes	No	No
Digital optical monitoring	Yes	Yes	Yes	Yes	Yes

NOTE

VCT is not supported on SX-FI48GPP, SX-FI-24GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI-8XG.

The procedures in this chapter describe how to configure the software to monitor hardware components.

Virtual cable testing

FastIron devices support **Virtual Cable Test** (VCT) technology. VCT technology enables the diagnosis of a conductor (wire or cable) by sending a pulsed signal into the conductor, then examining the reflection of that pulse. This method of cable analysis is referred to as Time Domain Reflectometry (TDR). By examining the reflection, the Brocade device can detect and report cable statistics such as local and remote link pair, cable length, and link status.

Virtual cable testing configuration notes

- This feature is supported on copper ports only. It is not supported on fiber ports.
- This feature is not supported on the SX-FI48GPP module running software release 07.2.02 or later.
- This feature is not supported on SX-FI2XG, SX-FI8XG, SX-FI24HF, SX-FI24GPP, and SX-FI48GPP modules running software release 07.3.00 or later.
- The port to which the cable is connected must be enabled when you issue the command to diagnose the cable. If the port is disabled, the command is rejected.
- If the port is operating at 100 Mbps half-duplex, the TDR test on one pair will fail.
- If the remote pair is set to forced 100 Mbps, any change in MDI/MDIX may cause the device to interpret the Multilevel Threshold-3 (MLT-3) as a reflected pulse, in which case, the device will report a faulty condition. In this scenario, it is recommended that you run the TDR test a few times for accurate results.

Virtual cable testing command syntax

To diagnose a cable using TDR, enter commands such as the following at the Privileged EXEC level of the CLI.

```
Brocade#phy cable-diag tdr 1
```

The above command diagnoses the cable attached to port 1.

When you issue the **phy-cable-diag** command, the command brings the port down for a second or two, then immediately brings the port back up.

Syntax: **phy cable-diag tdr** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Viewing the results of the cable analysis

To display the results of the cable analysis, enter a command such as the following at the Privileged EXEC level of the CLI.

```
Brocade>show cable-diag tdr 1
```

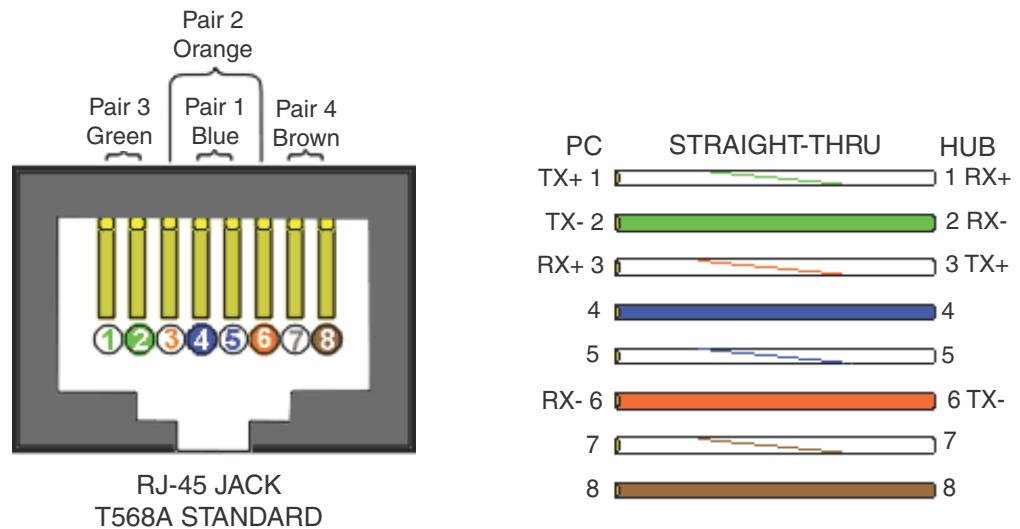
Port	Speed	Local pair	Pair Length	Remote pair	Pair status
01	1000M	Pair A	<50M	Pair B	Terminated
		Pair B	<50M	Pair A	Terminated
		Pair C	<50M	Pair D	Terminated
		Pair D	<50M	Pair C	Terminated

In the above output, **Local pair** indicates the assignment of wire pairs from left to right, where Pair A is the left-most pair. [Table 88](#) shows the **Local pair** mapping to the T568A pin/pair and color assignment from the TIA/EIA-568-B standard.

TABLE 88 Local pair definition

Local pair	T568A pair and color assignment
Pair A	Pair 3 (green)
Pair B	Pair 2 (orange)
Pair C	Pair 1 (blue)
Pair D	Pair 4 (brown)

[Figure 39](#) illustrates the T568A pin/pair assignment.

FIGURE 39 T568A pin/pair assignment

Syntax: `show cable-diag tdr <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

[Table 89](#) defines the fields shown in the command output.

TABLE 89 Cable statistics

This line...	Displays...
Port	The port that was tested.
Speed	The port current line speed.
Local pair	The local link name. Refer to Table 88 .
Pair Length	The cable length when terminated, or the distance to the point of fault when the line is not up.
Remote pair	The remote link name.
Pair status	The status of the link. This field displays one of the following: <ul style="list-style-type: none"> • Terminated: The link is up. • Shorted: A short is detected in the cable. • Open: An opening is detected in the cable. • ImpedMis: The impedance is mismatched. • Failed: The TDR test failed.

Digital optical monitoring

You can configure your Brocade device to monitor optical transceivers in the system, either globally or by specified ports. When this feature is enabled, the system will monitor the temperature and signal power levels for the optical transceivers in the specified ports. Console messages and Syslog messages are sent when optical operating conditions fall below or rise above the XFP, SFP, and SFP+ manufacturer recommended thresholds.

Digital optical monitoring configuration limitations

A Brocade chassis device can monitor a maximum of 24 SFPs and 12 XFPs.

Enabling digital optical monitoring

To enable optical monitoring on all Brocade-qualified optics installed in the device, use the following command.

```
Brocade(config)#optical-monitor
```

To enable optical monitoring on a specific port, use the following command.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#optical-monitor
```

To enable optical monitoring on a range of ports, use the following command.

```
Brocade(config)#interface ethernet 1/1 to 1/2
Brocade(config-mif-e10000-1/1-1/2)#optical-monitor
```

Syntax: `[no] optical-monitor`

Use the **no** form of the command to disable digital optical monitoring.

Setting the alarm interval

You can optionally change the interval between which alarms and warning messages are sent. The default interval is three minutes. To change the interval, use the following command.

```
Brocade(config)#interface ethernet 1/1 to 1/2
Brocade(config-mif-e10000-1/1-1/2)#optical-monitor 10
```

Syntax: `[no] optical-monitor [<alarm-interval>]`

For `<alarm-interval>`, enter a value between 1 and 65535. Enter 0 to disable alarms and warning messages.

NOTE

The commands **no optical-monitor** and **optical-monitor 0** perform the same function. That is, they both disable digital optical monitoring.

Displaying information about installed media

Use the **show media**, **show media slot**, and **show media ethernet** commands to obtain information about the media devices installed per device, per slot, and per port. The results displayed from these commands provide the Type, Vendor, Part number, Version and Serial number of the SFP, SFP+, or XFP optical device installed in the port. If there is no SFP, SFP+, or XFP optical device installed in a port, the “Type” field will display “EMPTY”.

On ICX 6430 and ICX 6450 devices, 1G copper ports will always be shown with the type as 1G M-C (Gig-Copper), even if the ports are not connected.

Use the **show media** command to obtain information about the media devices installed in a device.

```
Brocade#show media
Port 1/1/1:  Type : 1G M-C (Gig-Copper)
Port 1/1/2:  Type : 1G M-C (Gig-Copper)
Port 1/1/3:  Type : 1G M-C (Gig-Copper)
Port 1/1/4:  Type : 1G M-C (Gig-Copper)
Port 1/1/5:  Type : 1G M-C (Gig-Copper)
Port 1/1/6:  Type : 1G M-C (Gig-Copper)
Port 1/1/7:  Type : 1G M-C (Gig-Copper)
Port 1/1/8:  Type : 1G M-C (Gig-Copper)
Port 1/1/9:  Type : 1G M-C (Gig-Copper)
Port 1/1/10: Type : 1G M-C (Gig-Copper)
Port 1/1/11: Type : 1G M-C (Gig-Copper)
Port 1/1/12: Type : 1G M-C (Gig-Copper)
Port 1/1/13: Type : 1G M-C (Gig-Copper)
Port 1/1/14: Type : 1G M-C (Gig-Copper)
Port 1/1/15: Type : 1G M-C (Gig-Copper)
Port 1/1/16: Type : 1G M-C (Gig-Copper)
Port 1/1/17: Type : 1G M-C (Gig-Copper)
Port 1/1/18: Type : 1G M-C (Gig-Copper)
Port 1/1/19: Type : 1G M-C (Gig-Copper)
Port 1/1/20: Type : 1G M-C (Gig-Copper)
Port 1/1/21: Type : 1G M-C (Gig-Copper)
Port 1/1/22: Type : 1G M-C (Gig-Copper)
Port 1/1/23: Type : 1G M-C (Gig-Copper)
Port 1/1/24: Type : 1G M-C (Gig-Copper)
Port 1/2/1:  Type : 10GE SR 300m (SFP +)
Port 1/2/2:  Type : EMPTY
Port 1/2/3:  Type : 1G Twinax    1m (SFP)
Port 1/2/4:  Type : 1G Twinax    1m (SFP)
```

Use the **show media slot** command to obtain information about the media device installed in a slot.

```
Brocade#show media slot 1
Port    1/1: Type   : 1G M-SX(SFP)
          Vendor: Brocade Communications, Inc.  Version:
          Part#  : PL-XPL-VC-S13-19   Serial#: 425HC109
Port    1/2: Type   : 1G M-SX(SFP)
          Vendor: Brocade Communications, Inc.  Version:
          Part#  : PL-XPL-VC-S13-19   Serial#: 411HC0AH
Port    1/3: Type   : EMPTY
Port    1/4: Type   : 1G M-SX(SFP)
          Vendor: Brocade Communications, Inc.  Version: X1
          Part#  : FTRJ-8519-3       Serial#: H11654K
Port    1/5: Type   : EMPTY
Port    1/6: Type   : EMPTY
Port    1/7: Type   : 100M M-FX-IR(SFP)
```

```

Vendor: Brocade Communications, Inc. Version: A
Part# : FTLF1323P1BTR-FD Serial#: UCT000T
Port 1/8: Type : EMPTY
Port 1/9: Type : 100M M-FX-LR(SFP)
Vendor: Brocade Communications, Inc. Version: A
Part# : FTLF1323P1BTL-FD Serial#: UD3085J
Port 1/10: Type : EMPTY
Port 1/11: Type : 100M M-FX-SR(SFP)
Vendor: Brocade Communications, Inc. Version: A
Part# : FTLF1217P2BTL-F1 Serial#: UCQ003J
Port 1/12: Type : EMPTY
Port 1/13: Type : 100M M-FX-IR(SFP)
Vendor: Brocade Communications, Inc. Version: A
Part# : FTLF1323P1BTR-F1 Serial#: PCA2XC5

```

Use the **show media ethernet** command to obtain information about the media device installed in a port.

```

Brocade#show media e 1/17
Port 1/17: Type : 1G M-SX(SFP)
Vendor: Brocade Communications, Inc. Version:
Part# : PL-XPL-VC-S13-19 Serial#: 425HC109

```

Syntax: **show media** [**slot** <slot-num> | **ethernet** [<slot-num>/]<port-num>]

Viewing optical monitoring information

You can view temperature and power information for qualified XFPs, SFPs, and SFP+ installed in a FastIron device.

Use the **show optic** <port-number> command to view information about an XFP, SFP, or SFP+ installed in a particular port. The following shows example output.

Optical monitoring feature will not work in the following scenarios:

- The port is DOWN.
- The port is configured as a stacking port.
- The the optic module does not support optical monitoring.
- For ICX 6430 devices only:
 - If an SFP+ optic is inserted in an SFP only port, the optic will not initialize.
 - If an SFP optic is inserted in an SFP+ only port, the optic will not initialize.
 - If an optic is inserted into a device that supports both SFP and SFP+ optics, use the **speed-duplex** command to set the port speed correctly.

```

Brocade#show optic 13
Port Temperature Tx Power Rx Power Tx Bias Current
+---+-----+-----+-----+-----+
13 33.2968 C -005.4075 dBm -007.4328 dBm 6.306 mA
Normal Normal Normal Normal

```

Syntax: **show optic** <port-number>

Use the **show optic slot** <slot-number> command on a FastIron X Series chassis to view information about all qualified XFPs, SFPs, and SFP+ in a particular slot. The following shows example output.

```
Brocade>show optic slot 4
Port  Temperature      Tx Power      Rx Power      Tx Bias Current
-----+-----+-----+-----+-----+
4/1    30.8242 C    -001.8822 dBm -002.5908 dBm  41.790 mA
        Normal        Normal        Normal        Normal
4/2    31.7070 C    -001.4116 dBm -006.4092 dBm  41.976 mA
        Normal        Normal        Normal        Normal
4/3    30.1835 C           -000.5794 dBm   0.000 mA
        Normal    Low-Alarm        Normal    Low-Alarm
4/4     0.0000 C           0.000 mA
        Normal        Normal        Normal        Normal
```

Syntax: **show optic slot** <slot-number>

NOTE

The **show optic slot** <slot-number> command is supported on the FSX 800 and FSX 1600 only.

NOTE

The **show optic** function takes advantage of information stored and supplied by the manufacturer of the XFP, SFP, or SFP+ transceiver. This information is an optional feature of the Multi-Source Agreement standard defining the optical interface. Not all component suppliers have implemented this feature set. In such cases where the XFP, SFP, or SFP+ transceiver does not supply the information, a “Not Available” message will be displayed for the specific port on which the module is installed.

The following table describes the information displayed by the **show optic** command.

TABLE 90 Output from the show optic command

Field	Description
Port	The Brocade port number.
Temperature	<ul style="list-style-type: none"> The operating temperature, in degrees Celsius, of the optical transceiver. The alarm status, as described in Table 91.
Tx Power	<ul style="list-style-type: none"> The transmit power signal, in decibels (dB), of the measured power referenced to one milliwatt (mW). The alarm status, as described in Table 91.
Rx Power	<ul style="list-style-type: none"> The receive power signal, in decibels (dB), of the measured power referenced to one milliwatt (mW). The alarm status, as described in Table 91.
Tx Bias Current	<ul style="list-style-type: none"> The transmit bias power signal, in milliamperes (mA). The alarm status, as described in Table 91.

For Temperature, Tx Power, Rx Power, and Tx Bias Current in the **show optic** command output, values are displayed along with one of the following alarm status values: Low-Alarm, Low-Warn, Normal, High-Warn or High-Alarm. The thresholds that determine these status values are set by the manufacturer of the optical transceivers. [Table 91](#) describes each of these status values.

TABLE 91 Alarm status value description

Status value	Description
Low-Alarm	Monitored level has dropped below the "low-alarm" threshold set by the manufacturer of the optical transceiver.
Low-Warn	Monitored level has dropped below the "low-warn" threshold set by the manufacturer of the optical transceiver.
Normal	Monitored level is within the "normal" range set by the manufacturer of the optical transceiver.
High-Warn	Monitored level has climbed above the "high-warn" threshold set by the manufacturer of the optical transceiver.
High-Alarm	Monitored level has climbed above the "high-alarm" threshold set by the manufacturer of the optical transceiver.

Viewing optical transceiver thresholds

The thresholds that determine the alarm status values for an optical transceiver are set by the manufacturer of the XFP, SFP, or SFP+. To view the thresholds for a qualified optical transceiver in a particular port, use the **show optic threshold** command as shown below.

```
Brocade>show optic threshold 2/2
Port 2/2 sfp monitor thresholds:
Temperature High alarm          5a00          90.0000 C
Temperature Low alarm           d300         -45.0000 C
Temperature High warning        5500          85.0000 C
Temperature Low warning         d800         -40.0000 C
Supply Voltage High alarm       9088
Supply Voltage Low alarm       7148
Supply Voltage High warning     8ca0
Supply Voltage Low warning     7530
TX Bias High alarm              7530          60.000 mA
TX Bias Low alarm              01f4           1.000 mA
TX Bias High warning           61a8          50.000 mA
TX Bias Low warning            05dc           3.000 mA
TX Power High alarm            1f07         -001.0001 dBm
TX Power Low alarm             02c4         -011.4996 dBm
TX Power High warning          18a6         -001.9997 dBm
TX Power Low warning           037b         -010.5012 dBm
RX Power High alarm            2710          000.0000 dBm
RX Power Low alarm             0028         -023.9794 dBm
RX Power High warning          1f07         -001.0001 dBm
RX Power Low warning           0032         -023.0102 dBm
```

Syntax: **show optic threshold** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

For Temperature, Supply Voltage, TX Bias, TX Power, and RX Power, values are displayed for each of the following four alarm and warning settings: High alarm, Low alarm, High warning, and Low warning. The hexadecimal values are the manufacturer internal calibrations, as defined in the SFF-8472 standard. The other values indicate at what level (above the high setting or below the low setting) the system should send a warning message or an alarm. Note that these values are set by the manufacturer of the optical transceiver, and cannot be configured.

Syslog messages for optical transceivers

The system generates Syslog messages for optical transceivers in the following circumstances:

- The temperature, supply voltage, TX Bias, TX power, or TX power value goes above or below the high or low warning or alarm threshold set by the manufacturer.
- The optical transceiver does not support digital optical monitoring.
- The optical transceiver is not qualified, and therefore not supported by Brocade.

For details about the above Syslog messages, refer to [Appendix A, “Syslog messages”](#).

Digital optical monitoring

Syslog

[Table 92](#) lists individual Brocade switches and the Syslog features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 92 Supported Syslog features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Syslog messages	Yes	Yes	Yes	Yes	Yes
Real-time display of Syslog messages	Yes	Yes	Yes	Yes	Yes
Real-time display for Telnet or SSH sessions	Yes	Yes	Yes	Yes	Yes
Show log on all terminals	Yes	Yes	Yes	Yes	Yes
Time stamps	Yes	Yes	Yes	Yes	Yes
Multiple Syslog server logging (up to 6 Syslog servers)	Yes	Yes	Yes	Yes	Yes
Disabling logging of a message level	Yes	Yes	Yes	Yes	Yes
Changing the number of entries the local buffer can hold	Yes	Yes	Yes	Yes	Yes
Changing the log facility	Yes	Yes	Yes	Yes	Yes
Displaying Interface names in Syslog messages	Yes	Yes	Yes	Yes	Yes
Displaying TCP and UDP port numbers in Syslog messages	Yes	Yes	Yes	Yes	Yes
Retaining Syslog messages after a soft reboot	Yes	Yes	Yes	Yes	Yes
Clearing Syslog messages from the local buffer	Yes	Yes	Yes	Yes	Yes
Syslog messages for hardware errors	Yes	No	No	No	No

This chapter describes how to display Syslog messages and how to configure the Syslog facility, and lists the Syslog messages that Brocade devices can display during standard operation. Refer to [“Syslog”](#) on page 507 for a list of Syslog messages.

About Syslog messages

Brocade software can write syslog messages to provide information at the following severity levels:

- Emergencies
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

The device writes the messages to a local buffer.

You also can specify the IP address or host name of up to six Syslog servers. When you specify a Syslog server, the Brocade device writes the messages both to the system log and to the Syslog server.

Using a Syslog server ensures that the messages remain available even after a system reload. The Brocade local Syslog buffer is cleared during a system reload or reboot, but the Syslog messages sent to the Syslog server remain on the server.

NOTE

To enable the Brocade device to retain Syslog messages after a soft reboot (**reload** command). Refer to [“Retaining Syslog messages after a soft reboot”](#) on page 518.

The Syslog service on a Syslog server receives logging messages from applications on the local host or from devices such as a Layer 2 Switch or Layer 3 Switch. Syslog adds a time stamp to each received message and directs messages to a log file. Most Unix workstations come with Syslog configured. Some third party vendor products also provide Syslog running on NT.

Syslog uses UDP port 514 and each Syslog message thus is sent with destination port 514. Each Syslog message is one line with Syslog message format. The message is embedded in the text portion of the Syslog format. There are several subfields in the format. Keywords are used to identify each subfield, and commas are delimiters. The subfield order is insensitive except that the text subfield should be the last field in the message. All the subfields are optional.

Displaying Syslog messages

To display the Syslog messages in the device local buffer, enter the **show logging** command at any level of the CLI. The following shows an example display output.

```
Brocade>#show logging
```

```
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
```

```
Static Log Buffer:
```

```
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
```

```
Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet 4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

For information about the Syslog configuration information, time stamps, and dynamic and static buffers, refer to [“Displaying the Syslog configuration”](#) on page 510.

Enabling real-time display of Syslog messages

By default, to view Syslog messages generated by a Brocade device, you need to display the Syslog buffer or the log on a Syslog server used by the Brocade device.

You can enable real-time display of Syslog messages on the management console. When you enable this feature, the software displays a Syslog message on the management console when the message is generated. However, to enable display of real-time Syslog messages in Telnet or SSH sessions, you also must enable display within the individual sessions.

To enable real-time display of Syslog messages, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)#logging console
```

Syntax: [no] logging console

This command enables the real-time display of Syslog messages on the serial console. You can enter this command from the serial console or a Telnet or SSH session.

Enabling real-time display for a Telnet or SSH session

To also enable the real-time display for a Telnet or SSH session, enter the following command from the Privileged EXEC level of the session.

```
telnet@Brocade#terminal monitor
Syslog trace was turned ON
```

Syntax: terminal monitor

Notice that the CLI displays a message to indicate the status change for the feature. To disable the feature in the management session, enter the **terminal monitor** command again. The command toggles the feature on and off.

```
telnet@Brocade#terminal monitor
Syslog trace was turned OFF
```

Here is an example of how the Syslog messages are displayed.

```
telnet@Brocade#terminal monitor
Syslog trace was turned ON
SYSLOG: <9>Brocade, Power supply 2, power supply on left connector, failed

SYSLOG: <14>Brocade, Interface ethernet 6, state down

SYSLOG: <14>Brocade, Interface ethernet 2, state up
```

Displaying real-time Syslog messages

Any terminal logged on to a Brocade switch can receive real-time Syslog messages when the **terminal monitor** command is issued.

Syslog service configuration

The procedures in this section describe how to perform the following Syslog configuration tasks:

- Specify a Syslog server. You can configure the Brocade device to use up to six Syslog servers. (Use of a Syslog server is optional. The system can hold up to 1000 Syslog messages in an internal buffer.)
- Change the level of messages the system logs.
- Change the number of messages the local Syslog buffer can hold.
- Display the Syslog configuration.
- Clear the local Syslog buffer.

Logging is enabled by default, with the following settings:

- Messages of all severity levels (Emergencies – Debugging) are logged.
- By default, up to 50 messages are retained in the local Syslog buffer. This can be changed.
- No Syslog server is specified.

Displaying the Syslog configuration

To display the Syslog parameters currently in effect on a Brocade device, enter the following command from any level of the CLI.

```
Brocade>#show logging
```

```
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 3 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
              I=informational N=notification W=warning
```

```
Static Log Buffer:
```

```
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
```

```
Dynamic Log Buffer (50 entries):
```

```
Dec 15 18:46:17:I:Interface ethernet 1/4, state up
```

```
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
```

```
Dec 15 18:45:15:I:Warm start
```

Syntax: show logging

The Syslog display shows the following configuration information, in the rows above the log entries themselves.

TABLE 93 CLI display of Syslog buffer configuration

Field	Definition
Syslog logging	The state (enabled or disabled) of the Syslog buffer.
messages dropped	The number of Syslog messages dropped due to user-configured filters. By default, the software logs messages for all Syslog levels. You can disable individual Syslog levels, in which case the software filters out messages at those levels. Refer to “Disabling logging of a message level” on page 515. Each time the software filters out a Syslog message, this counter is incremented.
flushes	The number of times the Syslog buffer has been cleared by the clear logging command or equivalent Web Management Interface option. Refer to “Clearing the Syslog messages from the local buffer” on page 518.
overruns	The number of times the dynamic log buffer has filled up and been cleared to hold new entries. For example, if the buffer is set for 100 entries, the 101st entry causes an overrun. After that, the 201st entry causes a second overrun.
level	The message levels that are enabled. Each letter represents a message type and is identified by the key (level code) below the value. If you disable logging of a message level, the code for that level is not listed.
messages logged	The total number of messages that have been logged since the software was loaded.
level code	The message levels represented by the one-letter codes.

Static and dynamic buffers

The software provides two buffers:

- Static – logs power supply failures, fan failures, and temperature warning or shutdown messages
- Dynamic – logs all other message types

In the static log, new messages replace older ones, so only the most recent message is displayed. For example, only the most recent temperature warning message will be present in the log. If multiple temperature warning messages are sent to the log, the latest one replaces the previous one. The static buffer is not configurable.

The message types that appear in the static buffer do not appear in the dynamic buffer. The dynamic buffer contains up to the maximum number of messages configured for the buffer (50 by default), then begins removing the oldest messages (at the bottom of the log) to make room for new ones.

The static and dynamic buffers are both displayed when you display the log.

```
Brocade#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 3 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
              I=informational N=notification W=warning
```

Static Log Buffer:

```
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dec 15 19:00:14:A:Fan 2, fan on left connector, failed
```

Dynamic Log Buffer (50 entries):

```
Dec 15 18:46:17:I:Interface ethernet 4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

Notice that the static buffer contains two separate messages for fan failures. Each message of each type has its own buffer. Thus, if you replace fan 1 but for some reason that fan also fails, the software replaces the first message about the failure of fan 1 with the newer message. The software does not overwrite the message for fan 2, unless the software sends a newer message for fan 2.

Clearing log entries

When you clear log entries, you can selectively clear the static or dynamic buffer, or you can clear both. For example, to clear only the dynamic buffer, enter the following command at the Privileged EXEC level.

```
Brocade#clear logging dynamic-buffer
```

Syntax: clear logging [dynamic-buffer | static-buffer]

You can specify **dynamic-buffer** to clear the dynamic buffer or **static-buffer** to clear the static buffer. If you do not specify a buffer, both buffers are cleared.

Time stamps

The contents of the time stamp differ depending on whether you have set the time and date on the onboard system clock:

- If you have set the time and date on the onboard system clock, the date and time are shown in the following format.

mm dd hh:mm:ss

where

- *mm* – abbreviation for the name of the month
- *dd* – day
- *hh* – hours
- *mm* – minutes
- *ss* – seconds

For example, “Oct 15 17:38:03” means October 15 at 5:38 PM and 3 seconds.

- If you have not set the time and date on the onboard system clock, the time stamp shows the amount of time that has passed since the device was booted, in the following format.

<num>d<num>h<num>m<num>s

where

- <num>d – day
- <num>h – hours
- <num>m – minutes
- <num>s – seconds

For example, “188d1h01m00s” means the device had been running for 188 days, 11 hours, one minute, and zero seconds when the Syslog entry with this time stamp was generated.

Example of Syslog messages on a device with the onboard clock set

The example shows the format of messages on a device where the onboard system clock has been set. Each time stamp shows the month, the day, and the time of the system clock when the message was generated. For example, the system time when the most recent message (the one at the top) was generated was October 15 at 5:38 PM and 3 seconds.

```
Brocade#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 38 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
```

```
Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dec 15 19:00:14:A:Fan 2, fan on left connector, failed
```

```
Dynamic Log Buffer (50 entries):
Oct 15 17:38:03:warning:list 101 denied tcp 209.157.22.191(0)(Ethernet 18
0010.5alf.77ed) -> 198.99.4.69(http), 1 event(s)
Oct 15 07:03:30:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 18
0010.5alf.77ed) -> 198.99.4.69(http), 1 event(s)
Oct 15 06:58:30:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 18
0010.5alf.77ed) -> 198.99.4.69(http), 1 event(s)
```

Example of Syslog messages on a device with the onboard clock not set

The example shows the format of messages on a device where the onboard system clock is not set. Each time stamp shows the amount of time the device had been running when the message was generated. For example, the most recent message, at the top of the list of messages, was generated when the device had been running for 21 days, seven hours, two minutes, and 40 seconds.

```
Brocade#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 38 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
              I=informational N=notification W=warning

Static Log Buffer:

Dynamic Log Buffer (50 entries):
21d07h02m40s:warning:list 101 denied tcp 209.157.22.191(0)(Ethernet 4/18
0010.5alf.77ed) -> 198.99.4.69(http), 1 event(s)
19d07h03m30s:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 4/18
0010.5alf.77ed) -> 198.99.4.69(http), 1 event(s)
17d06h58m30s:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 4/18
0010.5alf.77ed) -> 198.99.4.69(http), 1 event(s)
```

Disabling or re-enabling Syslog

Syslog is enabled by default. To disable it, enter the **logging on** command at the global CONFIG level.

```
Brocade(config)#no logging on
```

Syntax: [no] **logging on** [<udp-port>]

The <udp-port> parameter specifies the application port used for the Syslog facility. The default is 514.

To re-enable logging, re-enter the **logging on** command.

```
Brocade(config)#logging on
```

This command enables local Syslog logging with the following defaults:

- Messages of all severity levels (Emergencies – Debugging) are logged.
- Up to 50 messages are retained in the local Syslog buffer.
- No Syslog server is specified.

Specifying a Syslog server

To specify a Syslog server, enter the **logging host** command.

```
Brocade(config)#logging host 10.0.0.99
```

Syntax: **logging host** <ip-addr> | <server-name>

Specifying an additional Syslog server

To specify an additional Syslog server, enter the **logging host** <ip-addr> command again. You can specify up to six Syslog servers.

```
Brocade(config)#logging host 10.0.0.99
```

Syntax: **logging host** <ip-addr> | <server-name>

Disabling logging of a message level

To change the message level, disable logging of specific message levels. You must disable the message levels on an individual basis.

For example, to disable logging of debugging and informational messages, enter the following commands.

```
Brocade(config)#no logging buffered debugging
Brocade(config)#no logging buffered informational
```

Syntax: [no] **logging buffered** <level> | <num-entries>

The <level> parameter can have one of the following values:

- alerts
- critical
- debugging
- emergencies
- errors
- informational
- notifications
- warnings

The commands in the example above change the log level to notification messages or higher. The software will not log informational or debugging messages. The changed message level also applies to the Syslog servers.

Changing the number of entries the local buffer can hold

You also can use the **logging buffered** command to change the number of entries the local Syslog buffer can store. For example.

```
Brocade(config)#logging buffered 100
Brocade(config)#write mem
Brocade(config)#exit
Brocade#reload
```

Syntax: **logging buffered** <num>

The default number of messages is 50. For FastIron Layer 2 switches, you can set the Syslog buffer limit from 1 – 100 entries. For FastIron Layer 3 switches, you can set the Syslog buffer limit from 1 – 1000 entries.

Local buffer configuration notes

- You must save the configuration and reload the software to place the change into effect.
- If you decrease the size of the buffer, the software clears the buffer before placing the change into effect.
- If you increase the size of the Syslog buffer, the software will clear some of the older locally buffered Syslog messages.

Changing the log facility

The Syslog daemon on the Syslog server uses a facility to determine where to log the messages from the Brocade device. The default facility for messages the Brocade device sends to the Syslog server is “user”. You can change the facility using the following command.

NOTE

You can specify only one facility. If you configure the Brocade device to use two Syslog servers, the device uses the same facility on both servers.

```
Brocade(config)#logging facility local0
```

Syntax: `logging facility <facility-name>`

The <facility-name> can be one of the following:

- kern – kernel messages
- user – random user-level messages
- mail – mail system
- daemon – system daemons
- auth – security or authorization messages
- syslog – messages generated internally by Syslog
- lpr – line printer subsystem
- news – netnews subsystem
- uucp – uucp subsystem
- sys9 – cron/at subsystem
- sys10 – reserved for system use
- sys11 – reserved for system use
- sys12 – reserved for system use
- sys13 – reserved for system use
- sys14 – reserved for system use
- cron – cron/at subsystem
- local0 – reserved for local use
- local1 – reserved for local use
- local2 – reserved for local use
- local3 – reserved for local use
- local4 – reserved for local use
- local5 – reserved for local use
- local6 – reserved for local use
- local7 – reserved for local use

Displaying interface names in Syslog messages

By default, an interface slot number (if applicable) and port number are displayed when you display Syslog messages. If you want to display the name of the interface instead of its number, enter the following command:

```
FastIron(config)# ip show-portname
```

This command is applied globally to all interfaces on Layer 2 Switches and Layer 3 Switches.

Syntax: [no] ip show-portname

By default, Syslog messages show the interface type, such as “ethernet”, “pos”, and so on. For example, you see the following

```
SYSLOG: <14>0d00h02m18s:ICX6610-48P Router System: Interface ethernet 1/1/5,
state up
```

However, if ip show-portname is configured and a name has been assigned to the port, the port name replaces the interface type as in the example below, where “port5_name” is the name of the port.

```
SYSLOG: <14>0d00h02m18s:ICX6610-48P Router System: Interface port5_name 1/1/5,
state up
```

Also, when you display the messages in the Syslog, you see the interface name under the Dynamic Log Buffer section. The actual interface number is appended to the interface name. For example, if the interface name is “lab” and its port number is “2”, you see “lab2” displayed as in the example below:

```
Brocade# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet Lab2, state up
Dec 15 18:45:15:I:Warm start
```

Displaying TCP or UDP port numbers in Syslog messages

The command **ip show-service-number-in-log** allows you to change the display of TCP or UDP application information from the TCP or UDP well-known port name to the TCP or UDP port number. For example, when this command is in effect, the Brocade device will display **http** (the well-known port name) instead of **80** (the port number) in the output of show commands, and other commands that contain application port information. By default, Brocade devices display TCP or UDP application information in named notation.

To display TCP or UDP port numbers instead of their names, enter the following command.

```
Brocade(config)#ip show-service-number-in-log
```

Syntax: [no] ip show-service-number-in-log

Retaining Syslog messages after a soft reboot

You can configure the device to save the System log (Syslog) after a soft reboot (**reload** command).

Syslog reboot configuration considerations

- If the Syslog buffer size was set to a different value using the CLI command **logging buffered**, the System log will be cleared after a soft reboot, even when this feature (**logging persistence**) is in effect. This will occur only with a soft reboot immediately following a Syslog buffer size change. A soft reboot by itself will not clear the System log. To prevent the system from clearing the System log, leave the number of entries allowed in the Syslog buffer unchanged.
- This feature does not save Syslog messages after a hard reboot. When the Brocade device is power-cycled, the Syslog messages are cleared.
- If **logging persistence** is enabled and you load a new software image on the device, you must first clear the log if you want to reload the device. (Refer to [“Clearing the Syslog messages from the local buffer”](#) on page 518.)

To configure the device to save the System log messages after a soft reboot, enter the following command.

```
Brocade(config)#logging persistence
```

Syntax: [no] **logging persistence**

Enter **no logging persistence** to disable this feature after it has been enabled.

Clearing the Syslog messages from the local buffer

To clear the Syslog messages stored in the local buffer of the Brocade device, enter the **clear logging** command.

```
Brocade#clear logging
```

Syntax: **clear logging**

Syslog messages for hardware errors

NOTE

This feature is supported on FastIron X Series devices only. It is **not** supported on FWS, FCX, and ICX devices.

FastIron Chassis devices support the display of hardware read and write errors encountered on a slot or module during bootup and during normal system operations. There are four types of errors, which may cause the system to disable or power down the modules on which they occur:

- Configuration read error
- Configuration write error
- Memory read error
- Memory write error

The following shows examples of some hardware errors in the **show logging** display output.

```
Brocade>#show logging
```

```
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
```

```
  Buffer logging: level ACDMEINW, 3 messages logged
```

```
  level code: A=alert C=critical D=debugging M=emergency E=error
```

```
               I=informational N=notification W=warning
```

```
Dynamic Log Buffer (50 lines):
```

```
0d00h00m27s:I:System: Interface ethernet mgmt1, state up
```

```
0d00h00m26s:N:powered On switch Fabric
```

```
0d00h00m17s:N:powered On switch Fabric
```

```
0d00h00m08s:I:System: Warm start
```

```
0d00h00m08s:I:SNMP: read-only community added by from session
```

```
0d00h00m02s:A:System: Module in slot 5 encountered unrecoverable PCI bridge  
validation failure. Module will be deleted.
```

```
0d00h00m02s:A:System: Module in slot 5 encountered unrecoverable PCI config read  
failure. Module will be deleted.
```

```
0d00h00m02s:A:System: Module in slot 5 encountered PCI config read error: Bus 10,  
Dev 3, Reg Offset 0.
```

```
0d00h00m00s:W:System: Fan speed changed automatically to 1
```

Syslog messages (alerts) for hardware errors are listed in [Table 1](#) on page 1995.

Syslog service configuration

Network Monitoring

Table 94 lists the individual FastIron switches and the network monitoring features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 94 Supported network monitoring features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Traffic counters for outbound traffic	Yes	No	No	No	Yes
Egress queue counters	No	No	Yes	Yes	No
Remote monitoring (RMON)	Yes	Yes	Yes	Yes	Yes
Specifying the maximum number of entries allowed in the RMON Control Table	Yes	No	Yes	Yes	Yes
sFlow version 2	Yes	Yes	Yes	Yes	ICX 6450 only
sFlow version 5 (default)	Yes	Yes	Yes	Yes	ICX 6450 only
sFlow support for IPv6 packets	Yes	Yes	Yes	Yes	ICX 6450 only
Uplink utilization lists	Yes	Yes	Yes	Yes	Yes

Basic system management

The following sections contain procedures for basic system management tasks.

Viewing system information

You can access software and hardware specifics for a Brocade Layer 2 Switch or Layer 3 Switch. For software specifics, refer to [“Software versions installed and running on a device”](#) on page 72.

To view the software and hardware details for the system, enter the **show version** command. The following shows an example output.

```

Brocade#show version

=====
Active Management CPU [Slot-9]:
  SW: Version 04.3.00b17T3e3 Copyright (c) 1996-2008 Brocade Communications,
  Inc., Inc.
    Compiled on Sep 25 2008 at 04:09:20 labeled as SXR04300b17
    (4031365 bytes) from Secondary sxr04300b17.bin
    BootROM: Version 04.0.00T3e5 (FEv2)
  HW: ANR-Chassis FastIron SX 1600-PREM (PROM-TYPE SX-FIL3U)
    Serial #: TE35069141
=====
SL 3: SX-FI424C 24-port Gig Copper
  Serial #: CY13073008
  P-ASIC 4: type 00D1, rev D2 subrev 00
  P-ASIC 5: type 00D1, rev D2 subrev 00
=====
SL 9: SX-FI8GMR4 8-port Management
  Serial #: CH37080003
  P-ASIC 16: type 00D1, rev D2 subrev 00
=====
SL 14: SX-FI42XGW 2-port 10G LAN/WAN
  Serial #: Invalid
  P-ASIC 26: type 01D1, rev 00 subrev 00
  P-ASIC 27: type 01D1, rev 00 subrev 00
=====
Active Management Module:
  660 MHz Power PC processor 8541 (version 32/0020) 66 MHz bus
  512 KB boot flash memory
  16384 KB code flash memory
  512 MB DRAM
The system uptime is 2 minutes 13 seconds
The system : started=warm start reloaded=by "reload"
*** NOT FOR PRODUCTION ***
*** AUTO SHUTDOWN IS OFF. PLEASE ACTIVATE WITH auto-shutdown ***

```

The following hardware details are listed in the output of the **show version** command:

- Chassis type
- PROM type (if applicable)
- Chassis serial number
- Management and interface module serial numbers and ASIC types

For a description of the software details in the output of the **show version** command, refer to [“Software versions installed and running on a device”](#) on page 72.

Syntax: **show version**

Viewing configuration information

You can view a variety of configuration details and statistics with the **show** option. The **show** option provides a convenient way to check configuration changes before saving them to flash.

The show options available will vary for Layer 2 Switches and Layer 3 Switches and by configuration level.

To determine the available show commands for the system or a specific level of the CLI, enter the following command.

```
Brocade#show ?
```

Syntax: **show** <option>

You also can enter “show” at the command prompt, then press the TAB key.

Viewing port statistics

Port statistics are polled by default every 10 seconds.

You can view statistics for ports by entering the following **show** commands:

- show interfaces
- show configuration
- show statistics

To display the statistics, enter a command such as the following.

```
Brocade#show statistics ethernet 1/3
Port  Link State      Dupl Speed Trunk Tag Priori MAC              Name
1/3    Up    Forward    Half 100M  None  No   level0 00e0.5200.0102

Port 1/3 Counters:
      InOctets          3200          OutOctets          256
      InPkts            50          OutPkts            4
InBroadcastPkts          0    OutBroadcastPkts          3
InMulticastPkts         48    OutMulticastPkts          0
InUnicastPkts           2    OutUnicastPkts          1
      InBadPkts          0
      InFragments        0
      InDiscards          0          OutErrors          0
      CRC                0          Collisions          0
      InErrors           0          LateCollisions          0
      InGiantPkts         0
      InShortPkts          0
      InJabber             0
InFlowCtrlPkts           0    OutFlowCtrlPkts          0
InBitsPerSec            264    OutBitsPerSec          16
InPktsPerSec             0    OutPktsPerSec           0
InUtilization            0.00%    OutUtilization          0.00%
```

Syntax: **show statistics** [ethernet [<port>]]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

[Table 95](#) lists the statistics displayed in the output of the **show statistics** command.

TABLE 95 Port statistics

Parameter	Description
Port configuration	
Port	The port number.
Link	The link state.
State	The STP state.
Dupl	The mode (full-duplex or half-duplex).
Speed	The port speed (10M, 100M, or 1000M).
Trunk	The trunk group number, if the port is a member of a trunk group.
Tag	Whether the port is a tagged member of a VLAN.
Priori	The QoS forwarding priority of the port (level0 – level7).
MAC	The MAC address of the port.
Name	The name of the port, if you assigned a name.
Statistics	
InOctets	The total number of good octets and bad octets received.
OutOctets	The total number of good octets and bad octets sent.
InPkts	The total number of packets received. The count includes rejected and local packets that are not sent to the switching core for transmission.
OutPkts	The total number of good packets sent. The count includes unicast, multicast, and broadcast packets.
InBroadcastPkts	The total number of good broadcast packets received.
OutBroadcastPkts	The total number of good broadcast packets sent.
InMulticastPkts	The total number of good multicast packets received.
OutMulticastPkts	The total number of good multicast packets sent.
InUnicastPkts	The total number of good unicast packets received.
OutUnicastPkts	The total number of good unicast packets sent.
InBadPkts	The total number of packets received for which one of the following is true: <ul style="list-style-type: none"> • The CRC was invalid. • The packet was oversized. • Jabbers: The packets were longer than 1518 octets and had a bad FCS. • Fragments: The packets were less than 64 octets long and had a bad FCS. • The packet was undersized (short).
InFragments	The total number of packets received for which both of the following was true: <ul style="list-style-type: none"> • The length was less than 64 bytes. • The CRC was invalid.
InDiscards	The total number of packets that were received and then dropped due to a lack of receive buffers.
OutErrors	The total number of packets with internal transmit errors such as TX underruns.

TABLE 95 Port statistics (Continued)

Parameter	Description
CRC	The total number of packets received for which all of the following was true: <ul style="list-style-type: none"> • The data length was between 64 bytes and the maximum allowable frame size. • No Collision or Late Collision was detected. • The CRC was invalid.
Collisions	The total number of packets received in which a Collision event was detected.
InErrors	The total number of packets received that had Alignment errors or phy errors.
LateCollisions	The total number of packets received in which a Collision event was detected, but for which a receive error (Rx Error) event was not detected.
InGiantPkts	The total number of packets for which all of the following was true: <ul style="list-style-type: none"> • The data length was longer than the maximum allowable frame size. • No Rx Error was detected. <p>NOTE: Packets are counted for this statistic regardless of whether the CRC is valid or invalid.</p>
InShortPkts	The total number of packets received for which all of the following was true: <ul style="list-style-type: none"> • The data length was less than 64 bytes. • No Rx Error was detected. • No Collision or Late Collision was detected. <p>NOTE: Packets are counted for this statistic regardless of whether the CRC is valid or invalid.</p>
InJabber	The total number of packets received for which all of the following was true: <ul style="list-style-type: none"> • The data length was longer than the maximum allowable frame size. • No Rx Error was detected. • The CRC was invalid.
InFlowCtrlPkts	The total number of flow control packets received.
OutFlowCtrlPkts	The total number of flow control packets transmitted.
InBitsPerSec	The number of bits received per second.
OutBitsPerSec	The number of bits sent per second.
InPktsPerSec	The number of packets received per second.
OutPktsPerSec	The number of packets sent per second.
InUtilization	The percentage of the port bandwidth used by received traffic.
OutUtilization	The percentage of the port bandwidth used by sent traffic.

Viewing STP statistics

You can view a summary of STP statistics for Layer 2 Switches and Layer 3 Switches. STP statistics are by default polled every 10 seconds.

To view spanning tree statistics, enter the **show span** command. To view STP statistics for a VLAN, enter the **span vlan** command.

Clearing statistics

You can clear statistics for many parameters using the **clear** command.

To determine the available **clear** commands for the system, enter the **clear** command at the Privileged EXEC level of the CLI.

```
Brocade#clear ?
```

Syntax: **clear** <option>

You also can enter “clear” at the command prompt, then press the TAB key.

Traffic counters for outbound traffic

You can configure traffic counters (also called transmit counters) that enable the Brocade device to count the following packet types on a port or port region:

- broadcast packets
- multicast packets
- unicast packets
- dropped packets due to congestion and egress filtering

Depending on the parameters specified with the traffic counter configuration, traffic counters record the number of outbound packets from any combination of the following sources:

- a specific port or all ports in a specific port region
- a specific VLAN or all VLANs
- a specific 802.1p priority queue or all priority queues

Traffic counters configuration notes

Consider the following rules when configuring traffic counters for outbound traffic.

- This feature is supported on FastIron X Series devices only.
- This feature is supported in the Layer 2, base Layer 3, and full Layer 3 codes.
- This feature applies to physical ports only, including 10 Gbps Ethernet ports and trunk ports. It does not apply to virtual interfaces.
- Once the enhanced traffic counters are read using the **show transmit-counter values** command, the counters are cleared (reset to zero).
- For each port region, you can enable a maximum of two traffic counters, regardless of whether traffic counters are enabled on individual ports or on all ports in the port region.
- Traffic counters increase for bridged filtered outbound traffic when any of the following conditions occur:
 - The port is disabled or the link is down.
 - The port or port region does not belong to the VLAN specified in the transmit counter configuration.
 - A Layer 2 protocol (e.g., spanning tree) has the port in a Blocked state.
 - The source port needs to be suppressed for multi-target packets.
 - The priority queue specified in the traffic counter is not allowed for some other reason.

- Unknown unicast and unregistered multicast packets are filtered.

Traffic counters configuration syntax

This section provides the syntax and configuration examples for enhanced traffic counters.

Example

To configure traffic counters for outbound traffic on a specific port, enter a command such as the following.

```
Brocade(config)#transmit-counter 4 port 18 only vlan 1 prio 7 enable
```

The above command creates and enables traffic counter 4 on port 18. The device will count the number of packets sent out on port 18 that are in VLAN 1 and have a priority queue of 7.

Example

To configure traffic counters for outbound traffic in a specific port region, enter a command such as the following.

```
Brocade(config)#transmit-counter 1 port 1 region vlan all prio all enable
```

The above command creates and enables traffic counter 1 on all ports that are in the same port region as port 1. The device will count the number of packets transmitted in this port region that belong to any VLAN and have any assigned priority queue.

Syntax: [no] **transmit-counter** <counter-ID> **port** [<slotnum>/]<port-num> **only** | **region** **vlan** <vlan-ID> | **all** **priority** <priority-queue> | **all** **enable**

Enter the **no** form of the command to remove the outbound traffic counter.

The <counter-ID> parameter identifies the traffic counter. You can configure up to 64 traffic counters. Enter a number from 1 – 64.

The <slotnum> parameter is required on chassis devices.

The <port-num> parameter is the port number to which enhanced traffic counters will apply. Enter the port number followed by **only** to apply the enhanced traffic counter to a specific port, or enter the port number followed by **region** to apply the enhanced traffic counter to all of the ports in the port region.

The <vlan-ID> parameter identifies the VLAN ID for which outbound traffic will be counted. Enter a number from 0 – 4095 or enter **all** to indicate all VLANs.

The <priority-queue> parameter identifies the 802.1p priority queue for which traffic will be counted. Enter a number from 0 – 7 or enter **all** to indicate all priority queues.

Displaying enhanced traffic counter profiles

To display the details of the traffic counters configured on your device, enter the **show transmit-counter profiles** command. The following shows an example output.

```
Brocade#show transmit-counter profiles
```

Tx Counter	Port(s)	Vlan Id	Priority	Device	Set
1	1 - 12	All	All	Dev 0	Set0
4	18	1	7	Dev 1	Set0
10	13 - 24	100	All	Dev 1	Set1

Displaying enhanced traffic counter statistics

To display the traffic counters for outbound traffic, enter the **show transmit-counter profiles** command.

NOTE

Once the enhanced traffic counters are displayed, the counters are cleared (reset to zero).

The following shows an example output.

```
Brocade#show transmit-counter values 1

Transmit Queue Counter Values for Counter 1:
Transmitted Frames:
  Known Unicast           : 17204
  Multicast & Unknown Unicast : 2797
  Broadcast               : 5
Dropped Frames:
  Bridge Egress Filtered   : 2
  Congestion Drops         : 0

Brocade#show transmit-counter values 4

Transmit Queue Counter Values for Counter 4:
Transmitted Frames:
  Known Unicast           : 124
  Multicast & Unknown Unicast : 2752
  Broadcast               : 0
Dropped Frames:
  Bridge Egress Filtered   : 37
  Congestion Drops         : 0
```

Syntax: **show transmit-counter values** *<number>*

where *<number>* identifies a valid enhanced traffic counter and is a value from 1 – 64.

TABLE 96 Outbound traffic counter statistics

This line...	Displays...
Transmitted frames	
Known Unicast	The number of known unicast packets transmitted.
Multicast & Unknown Unicast	The number of multicast and unknown unicast packets transmitted.
Broadcast	The number of broadcast packets transmitted.
Dropped Frames	

TABLE 96 Outbound traffic counter statistics (Continued)

This line...	Displays...
Bridge Egress Filtered	<p>The number of bridged outbound packets that were filtered and dropped. This number includes the number of packets that were dropped because of any one of the following conditions:</p> <ul style="list-style-type: none"> • The port was disabled or the link was down. • The port or port region does not belong to the VLAN specified in the transmit counter configuration. • A Layer 2 protocol (e.g., spanning tree) had the port in a Blocked state. • The source port was suppressed for multi-target packets. • The priority queue specified in the traffic counter was not allowed for some other reason. • Unknown unicast and unregistered multicast packets were filtered.
Congestion Drops	The number of outbound packets that were dropped because of traffic congestion.

Viewing egress queue counters on FCX devices

The **show interface** command displays the number of packets on a port that were queued for each QoS priority (traffic class) and dropped because of congestion.

NOTE

These counters do not include traffic on management ports or for a stack member unit that is down.

The egress queue counters display at the end of the **show interface** command output as shown in the following example.

```
Brocade#show interface e 1/1/1
GigabitEthernet1/1/1 is up, line protocol is up
  Hardware is GigabitEthernet, address is 0024.3877.8080 (bia 0024.3877.8080)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual none
  Member of L2 VLAN ID 52, port is untagged, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0, mac-learning is enabled
  Flow Control is config enabled, oper enabled, negotiation disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 96 bit times
  IP MTU 1500 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 256 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  215704 packets output, 13805066 bytes, 0 underruns
  Transmitted 0 broadcasts, 215704 multicasts, 0 unicasts
  0 output errors, 0 collisions
  Relay Agent Information option: Disabled
```

Egress queues:

Queue counters	Queued packets	Dropped Packets
0	0	0
1	0	0
2	1	0
3	0	0
4	0	0
5	0	0
6	0	0
7	215703	0

Syntax: **show interface** [ethernet <port>]

Specify the <port> variable in the format *stack-unit/slotnum/portnum*.

Table 97 defines the egress queue statistics displayed in the output.

TABLE 97 Egress queue statistics

Parameter	Description
Queue counters	The QoS traffic class.
Queued packets	The number of packets queued on the port for the given traffic class.
Dropped packets	The number of packets for the given traffic class that were dropped because of congestion.

Clearing the egress queue counters

You can clear egress queue statistics (reset them to zero), using the **clear statistics** and **clear statistics ethernet <port>** command.

Syntax: **clear statistics** [**ethernet <port>**]

Specify the <port> variable in the format *stack-unit/slotnum/portnum*.

RMON support

The Brocade RMON agent supports the following groups. The group numbers come from the RMON specification (RFC 1757):

- Statistics (RMON Group 1)
- History (RMON Group 2)
- Alarms (RMON Group 3)
- Events (RMON Group 9)

The CLI allows you to make configuration changes to the control data for these groups, but you need a separate RMON application to view and display the data graphically.

Maximum number of entries allowed in the RMON control table

You can specify the maximum number of entries allowed in the RMON control table, including alarms, history, and events. The default number of RMON entries allowed in the RMON control table is 1024 on the FESX, and 2048 on the FSX 800 and FSX 1600. The maximum number of RMON entries supported is 32768.

To set the maximum number of allowable entries to 3000 in the RMON history table, enter commands such as the following.

```
Brocade(config)#system-max rmon-entries 3000
Brocade(config)#write mem
Brocade(config)#exit
Brocade#reload
```

NOTE

You must save the change to the startup-config file and reload or reboot. The change does not take effect until you reload or reboot.

Syntax: **system-max rmon-entries <value>**

where *<value>* can be:

- 1536 – 32768 for FSX 800 and FSX 1600 devices
- 128 – 32768 for FESX devices

Statistics (RMON group 1)

Count information on multicast and broadcast packets, total packets sent, undersized and oversized packets, CRC alignment errors, jabbers, collision, fragments and dropped events is collected for each port on a Brocade Layer 2 Switch or Layer 3 Switch.

The statistics group collects statistics on promiscuous traffic across an interface. The interface group collects statistics on total traffic into and out of the agent interface.

No configuration is required to activate collection of statistics for the Layer 2 Switch or Layer 3 Switch. This activity is by default automatically activated at system start-up.

You can view a textual summary of the statistics for all ports by entering the following CLI command.

```
Brocade#show rmon statistics
Ethernet statistics 1 is active, owned by monitor
Interface 1/1 (ifIndex 1) counters
      Octets      0
      Drop events  0
      Broadcast pkts  0
      CRC alignment errors  0
      Oversize pkts  0
      Jabbers      0
      64 octets pkts  0
      128 to 255 octets pkts  0
      512 to 1023 octets pkts  0
      Packets      0
      Multicast pkts  0
      Undersize pkts  0
      Fragments    0
      Collisions   0
      65 to 127 octets pkts  0
      256 to 511 octets pkts  0
      1024 to 1518 octets pkts  0
```

Syntax: `show rmon statistics [ethernet <port>]`

NOTE

Though 48GC modules receive oversized packets and jabbers, they do not support count information for oversized packets and jabbers and the output of the **show rmon statistics** command reports 0 for both of these counters.

The *<port>* parameter specifies the port number. You can use the physical port number or the SNMP port number. The physical port number is based on the product. If you specify a physical port, specify the *<port>* variable in one of the following formats:

- FCX stackable switches – *<stack-unit/slotnum/portnum>*
- FSX 800 and FSX 1600 chassis devices – *<slotnum/portnum>*
- FESX compact switches – *<portnum>*

The SNMP numbers of the ports start at 1 and increase sequentially. For example, if you are using a Chassis device and slot 1 contains an 8-port module, the SNMP number of the first port in slot 2 is 9. The physical port number of the same port is 2/1.

This command shows the following information.

TABLE 98 Export configuration and statistics

Parameter	Definition
Octets	The total number of octets of data received on the network. This number includes octets in bad packets. This number does not include framing bits but does include Frame Check Sequence (FCS) octets.
Drop events	Indicates an overrun at the port. The port logic could not receive the traffic at full line rate and had to drop some packets as a result. The counter indicates the total number of events in which packets were dropped by the RMON probe due to lack of resources. This number is not necessarily the number of packets dropped, but is the number of times an overrun condition has been detected.
Packets	The total number of packets received. This number includes bad packets, broadcast packets, and multicast packets.
Broadcast pkts	The total number of good packets received that were directed to the broadcast address. This number does not include multicast packets.
Multicast pkts	The total number of good packets received that were directed to a multicast address. This number does not include packets directed to the broadcast address.
CRC alignment errors	The total number of packets received that were from 64 – 1518 octets long, but had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). The packet length does not include framing bits but does include FCS octets.
Undersize pkts	The total number of packets received that were less than 64 octets long and were otherwise well formed. This number does not include framing bits but does include FCS octets.
Fragments	The total number of packets received that were less than 64 octets long and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). It is normal for this counter to increment, since it counts both runts (which are normal occurrences due to collisions) and noise hits. This number does not include framing bits but does include FCS octets.
Oversize packets	The total number of packets received that were longer than 1518 octets and were otherwise well formed. This number does not include framing bits but does include FCS octets. NOTE: 48GC modules do not support count information on oversized packets and report 0.
Jabbers	The total number of packets received that were longer than 1518 octets and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). NOTE: This definition of jabber is different from the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition where any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms. This number does not include framing bits but does include FCS octets. NOTE: 48GC modules do not support count information on jabbers and report 0.
Collisions	The best estimate of the total number of collisions on this Ethernet segment.
64 octets pkts	The total number of packets received that were 64 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.

TABLE 98 Export configuration and statistics (Continued)

Parameter	Definition
65 to 127 octets pkts	The total number of packets received that were 65 – 127 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
128 to 255 octets pkts	The total number of packets received that were 128 – 255 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
256 to 511 octets pkts	The total number of packets received that were 256 – 511 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
512 to 1023 octets pkts	The total number of packets received that were 512 – 1023 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
1024 to 1518 octets pkts	The total number of packets received that were 1024 – 1518 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.

History (RMON group 2)

All active ports by default will generate two history control data entries per active Brocade Layer 2 Switch port or Layer 3 Switch interface. An active port is defined as one with a link up. If the link goes down the two entries are automatically deleted.

Two history entries are generated for each device:

- A sampling of statistics every 30 seconds
- A sampling of statistics every 30 minutes

The history data can be accessed and displayed using any of the popular RMON applications

A sample RMON history command and its syntax is shown below.

```
Brocade(config)#rmon history 1 interface 1 buckets 10 interval 10 owner nyc02
```

Syntax: **rmon history** <entry-number> **interface** <port> **buckets** <number> **interval** <sampling-interval> **owner** <text-string>

You can modify the sampling interval and the bucket (number of entries saved before overwrite) using the CLI. In the above example, owner refers to the RMON station that will request the information.

NOTE

To review the control data entry for each port or interface, enter the **show rmon history** command.

Alarm (RMON group 3)

Alarm is designed to monitor configured thresholds for any SNMP integer, time tick, gauge or counter MIB object. Using the CLI, you can define what MIB objects are monitored, the type of thresholds that are monitored (falling, rising or both), the value of those thresholds, and the sample type (absolute or delta).

An alarm event is reported each time that a threshold is exceeded. The alarm entry also indicates the action (event) to be taken if the threshold be exceeded.

A sample CLI alarm entry and its syntax is shown below.

```
Brocade(config)#rmon alarm 1 ifInOctets.6 10 delta rising-threshold 100 1 falling
threshold 50 1 owner nyc02
```

Syntax: **rmon alarm** <entry-number> <MIB-object.interface-num> <sampling-time>
 <sample-type>
 <threshold-type> <threshold-value> <event-number> <threshold-type> <threshold-value>
 <event-number>
owner <text-string>

Event (RMON group 9)

There are two elements to the Event Group—the **event control table** and the **event log table**.

The event control table defines the action to be taken when an alarm is reported. Defined events can be found by entering the CLI command, show event. The Event Log Table collects and stores reported events for retrieval by an RMON application.

A sample entry and syntax of the event control table is shown below.

```
Brocade(config)#rmon event 1 description 'testing a longer string' log-and-trap
public owner nyc02
```

Syntax: **rmon event** <event-entry> **description** <text-string> **log | trap | log-and-trap** **owner**
 <rmon-station>

sFlow

NOTE

FastIron devices support sFlow version 5 by default.

sFlow is a standards-based protocol that allows network traffic to be sampled at a user-defined rate for the purpose of monitoring traffic flow patterns and identifying packet transfer rates on user-specified interfaces.

When sFlow is enabled on a Layer 2 or Layer 3 switch, the system performs the following sFlow-related tasks:

- Samples traffic flows by copying packet header information
- Identifies ingress and egress interfaces for the sampled flows
- Combines sFlow samples into UDP packets and forwards them to the sFlow collectors for analysis
- Forwards byte and packet count data, or counter samples, to sFlow collectors

sFlow is described in RFC 3176, “InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks”.

On FWS and FCX Series devices, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from FastIron X Series devices, which support seven priorities instead of eight when sFlow is enabled. In this case, QoS queue 1 is reserved for sFlow and is not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0.

sFlow version 5

sFlow version 5 enhances and modifies the format of the data sent to the sFlow collector. sFlow version 5 introduces several new sFlow features and also defines a new datagram syntax used by the sFlow agent to report flow samples and interface counters to the sFlow collector.

sFlow version 5 adds support for the following:

- sFlow version 5 datagrams
- Sub-agent support
- Configurable sFlow export packet size
- Support for the new data field and sample type length in flow samples
- Configurable interval for exporting Brocade-specific data structure

sFlow version 5 is backward-compatible with sFlow version 2. By default, the sFlow agent exports sFlow version 5 flow samples by default, but you can configure the device to export the data in sFlow version 2 format. You can switch between sFlow version 2 and sFlow version 5 formats. The sFlow collector automatically parses each incoming sample and decodes it based on the version number.

The configuration procedures for sFlow version 5 are the same as for sFlow version 2, except where explicitly noted. Configuration procedures for sFlow are in the section [“Configuring and enabling sFlow”](#) on page 539. The features and CLI commands that are specific to sFlow version 5 are described in the section [“sFlow version 5 feature configuration”](#) on page 546.

sFlow support for IPv6 packets

The Brocade implementation of sFlow features support IPv6 packets. This support includes extended router information and extended gateway information in the sampled packet. Note that sFlow support for IPv6 packets exists only on devices running software that supports IPv6.

The configuration procedures for this feature are the same as for IPv4, except where the collector is a link-local address on a Layer 3 switch. For details refer to [“Specifying the collector”](#) on page 539.

Extended router information

IPv6 sFlow sampled packets include the following extended router information:

- IP address of the next hop router
- Outgoing VLAN ID
- Source IP address prefix length
- Destination IP address prefix length

Note that in IPv6 devices, the prefix lengths of the source and destination IP addresses are collected if BGP is configured and the route lookup is completed. In IPv4 devices, this information is collected only if BGP is configured on the devices.

Extended gateway information

If BGP is enabled, extended gateway information is included in IPv6 sFlow sampled packets, including the following BGP information about a packet destination route:

- The autonomous system (AS) number for the router
- The source IP AS of the route
- The source peer AS for the route
- The AS patch to the destination

NOTE

AS communities and local preferences are not included in the sampled packets.

To obtain extended gateway information, use “struct extended_gateway” as described in RFC 3176.

IPv6 packet sampling

IPv6 sampling is performed by the packet processor. The system uses the sampling rate setting to selectively mark the monitoring bit in the header of an incoming packet. Marked packets tell the CPU that the packets are subject to sFlow sampling.

sFlow configuration considerations

This section lists the sFlow configuration considerations on Brocade devices.

On FWS and FCX Series devices, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from FastIron X Series devices, which support seven priorities instead of eight when sFlow is enabled. In this case, QoS queue 1 is reserved for sFlow and is not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0.

If an FCX stack is rebooted, sFlow is disabled on standby and member units until the configuration is synchronized between the Active and Standby Controllers.

sFlow and hardware support

- Brocade devices support sFlow packet sampling of inbound traffic only. These devices do not sample outbound packets. However, Brocade devices support byte and packet count statistics for both traffic directions.
- sFlow is supported on all Ethernet ports (10/100, Gbps, and 10 Gbps)

sFlow and CPU utilization

Enabling sFlow may cause a slight and noticeable increase of up to 20% in CPU utilization. In typical scenarios, this is normal behavior for sFlow, and does not affect the functionality of other features on the switch.

sFlow and source address

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the IP address of the device that sent the data:

- On a Layer 2 Switch, agent_address is the Layer 2 Switch management IP address. You must configure the management IP address in order to export sFlow data from the device. If the switch has both an IPv4 and IPv6 address, the agent_address is the IPv4 address. If the switch has an IPv6 address only, the agent_address is the global IPv6 address.
- On a Layer 3 Switch with IPv6 interfaces only, sFlow looks for an IPv6 address in the following order, and uses the first address found:
 - The first IPv6 address on the lowest-numbered loopback interface
 - The first IPv6 address on the lowest-numbered VE interface
 - The first IPv6 address on any interface
- On a Layer 3 Switch with both IPv4 and IPv6 interfaces, or with IPv4 interfaces only, sFlow looks for an IP address in the following order, and uses the first address found:
 - The IPv4 router ID configured by the **ip router-id** command
 - The first IPv4 address on the lowest-numbered loopback interface
 - The first IPv4 address on the lowest-numbered virtual interface
 - The first IPv4 address on any interface

NOTE

The device uses the router ID only if the device also has an IP interface with the same address. Router ID is not supported on IPv6 devices.

NOTE

If an IP address is not already configured when you enable sFlow, the feature uses the source address 0.0.0.0. To display the agent_address, enable sFlow, then enter the **show sflow** command. Refer to [“Enabling sFlow forwarding”](#) on page 545 and [“Displaying sFlow information”](#) on page 549.

NOTE

In sFlow version 5, you can set an arbitrary IPv4 or IPv6 address as the sFlow agent IP address. Refer to [“Specifying the sFlow agent IP address”](#) on page 547.

sFlow and source port

By default, sFlow sends data to the collector out of UDP source port 8888, but you can specify a different source port. For more information, refer to [“Changing the sFlow source port”](#) on page 545.

sFlow and sampling rate

The **sampling rate** is the average ratio of the number of packets incoming on an sFlow enabled port, to the number of flow samples taken from those packets. sFlow sampling can affect performance in some configurations.

Note that on the FastIron devices, the configured sampling rate and the actual rate are the same. The software does not adjust the configured sampling rate as on other Brocade devices.

sFlow and port monitoring

- FWS and FCX Series devices support sFlow and port monitoring together on the same port.

- FastIron X Series devices support port monitoring and sFlow together on the same device. The caveat is that these features cannot be configured together within the same port region on non-third generation modules. The following third-generation SX modules support sFlow and mirroring on the same port:
 - SX-FI48GPP
 - SX-FI-24GPP
 - SX-FI-24HF
 - SX-FI-2XG
 - SX-FI-8XG

Configuring and enabling sFlow

NOTE

The commands in this section apply to sFlow version 2 and sFlow version 5. CLI commands that are specific to sFlow version 5 are documented in [“sFlow version 5 feature configuration”](#) on page 546.

To configure sFlow, perform the following tasks:

- Optional – If your device supports sFlow version 5, change the version used for exporting sFlow data
- Specify collector information. The collector is the external device to which you are exporting the sFlow data. You can specify up to four collectors.
- Optional – Change the polling interval
- Optional – Change the sampling rate
- Optional – Change the sFlow source port
- Enable sFlow globally
- Enable sFlow forwarding on individual interfaces
- Enable sFlow forwarding on individual trunk ports
- If your device supports sFlow version 5, configure sFlow version 5 features

NOTE

If you change the router ID or other IP address value that sFlow uses for its agent_address, you need to disable and then re-enable sFlow to cause the feature to use the new source address.

Specifying the collector

sFlow exports traffic statistics to an external collector. You can specify up to four collectors. You can specify more than one collector with the same IP address if the UDP port numbers are unique. You can have up to four unique combinations of IP addresses and UDP port numbers.

Specifying an sFlow collector on IPv4 devices

To specify an sFlow collector on an IPv4 device, enter a command such as the following.

```
Brocade(config)#sflow destination 10.10.10.1
```

This command specifies a collector with IPv4 address 10.10.10.1, listening for sFlow data on UDP port 6343.

Syntax: [no] sflow destination <ip-addr> [<dest-udp-port>]

The <ip-addr> parameter specifies the IP address of the collector.

The <dest-udp-port> parameter specifies the UDP port on which the sFlow collector will be listening for exported sFlow data. The default port number is 6343.

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the device that sent the data. Refer to [“sFlow and source address”](#) on page 537.

Specifying an sFlow collector on IPv6 devices

To specify an sFlow collector on an IPv6 device, enter a command such as the following.

```
Brocade(config)#sflow destination ipv6 2003:0:0::0b:02a
```

This command specifies a collector with IPv6 address 2003:0::0b:02a, listening for sFlow data on UDP port 6343.

Syntax: [no] sflow destination ipv6 <ip-addr> [<dest-udp-port>]

The <ip-addr> parameter specifies the IP address of the collector.

The <dest-udp-port> parameter specifies the UDP port on which the sFlow collector will be listening for exported sFlow data. The default port number is 6343.

If the IPv6 address you specify is a link-local address on a Layer 3 switch, you must also specify the **outgoing-interface ethernet** <port-num> or the **ve** <port-num>. This identifies the outgoing interface through which the sampled packets will be sent.

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the device that sent the data. Refer to “[sFlow and source address](#)” on page 537.

Changing the polling interval

The polling interval defines how often sFlow byte and packet counter data for a port are sent to the sFlow collectors. If multiple ports are enabled for sFlow, the Brocade device staggers transmission of the counter data to smooth performance. For example, if sFlow is enabled on two ports and the polling interval is 20 seconds, the Brocade device sends counter data every ten seconds. The counter data for one of the ports are sent after ten seconds, and counter data for the other port are sent after an additional ten seconds. Ten seconds later, new counter data for the first port are sent. Similarly, if sFlow is enabled on five ports and the polling interval is 20 seconds, the Brocade device sends counter data every four seconds.

The default polling interval is 20 seconds. You can change the interval to a value from 1 to any higher value. The interval value applies to all interfaces on which sFlow is enabled. If you set the polling interval to 0, counter data sampling is disabled.

To change the polling interval, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#sflow polling-interval 30
```

Syntax: [no] sflow polling-interval <secs>

The <secs> parameter specifies the interval and can be from 1 to any higher value. The default is 20 seconds. If you specify 0, counter data sampling is disabled.

Changing the sampling rate

The sampling rate is the average ratio of the number of packets incoming on an sFlow-enabled port, to the number of flow samples taken from those packets.

You can change the default (global) sampling rate. You also can change the rate on an individual port, overriding the default sampling rate of 512. With a sampling rate of 512, on average, one in every 512 packets forwarded on an interface is sampled.

Configuration considerations

The sampling rate is a fraction in the form $1/N$, meaning that, on average, one out of every N packets will be sampled. The **sflow sample** command at the global level or port level specifies N , the denominator of the fraction. Thus a higher number for the denominator means a lower sampling rate since fewer packets are sampled. Likewise, a lower number for the denominator means a higher sampling rate because more packets are sampled. For example, if you change the denominator from 512 to 128, the sampling rate increases because four times as many packets will be sampled.

NOTE

Brocade recommends that you do not change the denominator to a value lower than the default. Sampling requires CPU resources. Using a low denominator for the sampling rate can cause high CPU utilization.

Configured rate and actual rate

When you enter a sampling rate value, this value is the *configured rate* as well as the *actual sampling rate*.

Change to global rate

If you change the global sampling rate, the change is applied to all sFlow-enabled ports **except** those ports on which you have already explicitly set the sampling rate. For example, suppose that sFlow is enabled on ports 1/1, 1/2, and 5/1. If you configure the sampling rate on port 1/1 but leave the other two ports using the default rate, then a change to the global sampling rate applies to ports 1/2 and 5/1 but not port 1/1. sFlow assumes that you want to continue using the sampling rate you explicitly configured on an individual port even if you globally change the sampling rate for the other ports.

Module rate

While different ports on a module may be configured to have different sampling rates, the hardware for the module will be programmed to take samples at a single rate (the module sampling rate). The module sampling rate will be the highest sampling rate (i.e. lowest number) configured for any of the ports on the module.

When ports on a given module are configured with different sampling rates, the CPU discards some of the samples supplied by the hardware for ports with configured sampling rates which are lower than the module sampling rate. This is referred to as subsampling, and the ratio between the port sampling rate and the module sampling rate is known as the subsampling factor. For example, if the module in slot 4 has sFlow enabled on ports 4/2 and 4/8, and port 4/2 is using the default sampling rate of 512, and port 4/8 is configured explicitly for a rate of 2048, then the module sampling rate will be 512 because this is the highest port sampling rate (lowest number). The subsampling factor for port 4/2 will be 1, meaning that every sample taken by the hardware will be exported, while the subsampling factor for port 4/8 will be 4, meaning that one out of every four samples taken by the hardware will be exported. Whether a port's sampling rate is configured explicitly, or whether it uses the global default setting, has no effect on the calculations.

You do not need to perform any of these calculations to change a sampling rate. For simplicity, the syntax information in this section lists the valid sampling rates. You can display the rates you entered for the default sampling rate, module rates, and all sFlow-enabled ports by entering the **show sflow** command. Refer to [“Displaying sFlow information”](#) on page 549.

Sampling rate for new ports

When you enable sFlow on a port, the port's sampling rate is set to the global default sampling rate. This also applies to ports on which you disable and then re-enable sFlow. The port does not retain the sampling rate it had when you disabled sFlow on the port, even if you had explicitly set the sampling rate on the port.

Changing the default sampling rate

To change the default (global) sampling rate, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#sflow sample 2048
```

Syntax: [no] sflow sample <num>

The <num> parameter specifies the average number of packets from which each sample will be taken. The software rounds the value you enter to the next higher odd power of 2. This value becomes the actual default sampling rate and is one of the following:

- 2
- 8
- 32
- 128
- 512
- 2048
- 4096
- 8192
- 32768
- 131072
- 524288
- 2097152
- 8388608
- 33554432
- 134217728
- 536870912
- 2147483648

For example, if the configured sampling rate is 1000, then the actual rate is 2048 and 1 in 2048 packets are sampled by the hardware.

Changing the sampling rate of a module

You cannot change a module sampling rate directly. You can change a module sampling rate only by changing the sampling rate of a port on that module.

Changing the sampling rate on a port

You can configure an individual port to use a different sampling rate than the global default sampling rate. This is useful in cases where ports have different bandwidths. For example, if you are using sFlow on 10/100 ports and Gbps Ethernet ports, you might want to configure the Gbps ports to use a higher sampling rate (and thus gather fewer samples per number of packets) than the 10/100 ports.

To change the sampling rate on an individual port, enter a command such as the following at the configuration level for the port.

```
Brocade(config-if-1/1)#sflow sample 8192
```

Syntax: [no] sflow sample <num>

The <num> parameter specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. The actual sampling rate becomes one of the values listed in [“Changing the default sampling rate”](#).

NOTE

Configuring a sampling rate on a port that is the primary port of a trunk applies that same sampling rate to all ports in the trunk.

Changing the sampling rate for a trunk port

You can configure an individual static trunk port to use a different sampling rate than the global default sampling rate. This feature is also supported on LACP trunk ports. This feature is useful in cases where ports have different bandwidths. For example, if you are using sFlow on 10/100 ports and Gbps Ethernet ports, you might want to configure the Gbps ports to use a higher sampling rate (and thus gather fewer samples per number of packets) than the 10/100 ports.

To configure a static trunk port to use a different sampling rate than the global default sampling rate, enter commands such as the following:

```
Brocade(config)#trunk e 4/1 to 4/8
Brocade(config-trunk-4/1-4/8)sflow sample 8192
```

Syntax: [no] sflow sample <num>

The <num> parameter specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. The actual sampling rate becomes one of the values listed in [“Changing the default sampling rate”](#).

NOTE

Configuring a sampling rate on only the port that is the primary port of a trunk automatically applies that same sampling rate to all ports in the trunk.

Changing the sFlow source port

By default, sFlow sends data to the collector using UDP source port 8888, but you can change the source UDP port to any port number in the range 1025-65535.

To change the source UDP port, enter a command such as the following:

```
Brocade(config)#sflow source-port 8000
```

Syntax: [no] sflow source-port <num>

The <num> parameter specifies the sFlow source port.

Enabling sFlow forwarding

sFlow exports data only for the interfaces on which you enable sFlow forwarding. You can enable sFlow forwarding on Ethernet interfaces.

To enable sFlow forwarding, perform the following:

- Globally enable the sFlow feature
- Enable sFlow forwarding on individual interfaces
- Enable sFlow forwarding on individual trunk ports

NOTE

Before you enable sFlow, make sure the device has an IP address that sFlow can use as its source address. Refer to [“sFlow and source address”](#) on page 537 for the source address requirements.

NOTE

When you enable sFlow forwarding on an 802.1X-enabled interface, the samples taken from the interface include the username used to obtain access to either or both the inbound and outbound ports, if that information is available. For information about 802.1X, refer to [Chapter 43, “802.1X Port Security”](#).

Command syntax for enabling sFlow forwarding

This section shows how to enable sFlow forwarding.

Globally enabling sFlow forwarding

To enable sFlow forwarding, you must first enable it on a global basis, then on individual interfaces or trunk ports, or both.

To globally enable sFlow forwarding, enter the following command.

```
Brocade(config)#sflow enable
```

You can now enable sFlow forwarding on individual ports as described in the next two sections.

Syntax: [no] sflow enable

Enabling sFlow forwarding on individual interfaces

To enable sFlow forwarding enter commands such as the following.

```
Brocade(config)#sflow enable
Brocade(config)#interface ethernet 1/1 to 1/8
Brocade(config-mif-1/1-1/8)#sflow forwarding
```

These commands globally enable sFlow, then enable sFlow forwarding on Ethernet ports 1/1 – 1/8. You must use both the **sflow enable** and **sflow forwarding** commands to enable the feature.

Syntax: [no] sflow enable

Syntax: [no] sflow forwarding

Enabling sFlow forwarding on individual trunk ports

This feature is supported on individual ports of a static trunk group. It is also supported on LACP trunk ports.

NOTE

When you enable sFlow forwarding on a trunk port, only the primary port of the trunk group forwards sFlow samples.

To enable sFlow forwarding on a trunk port, enter commands such as the following.

```
Brocade(config)#sflow enable
Brocade(config)#trunk e 4/1 to 4/8
Brocade(config-trunk-4/1-4/8)#config-trunk-ind
Brocade(config-trunk-4/1-4/8)#sflow forwarding e 4/2
```

These commands globally enable sFlow, then enable sFlow forwarding on trunk port e 4/2. You must use both the **sflow enable** and **sflow forwarding** commands to enable the feature.

Syntax: [no] sflow enable

Syntax: [no] sflow forwarding

sFlow version 5 feature configuration

NOTE

The commands in this section are supported when sFlow version 5 is enabled on the device. These commands are not supported with sFlow version 2. sFlow version 5 also supports all of the sFlow configuration commands in [“Configuring and enabling sFlow”](#) on page 539.

When sFlow version 5 is enabled on the device, you can do the following:

- Specify the sFlow version (version 2 or version 5)
- Specify the sFlow agent IP address
- Specify the maximum flow sample size
- Export CPU and memory usage Information to the sFlow collector
- Specify the polling interval for exporting CPU and memory usage information to the sFlow collector
- Export CPU-directed data (management traffic) to the sFlow collector

Egress interface ID for sampled broadcast and multicast packets

For broadcast and multicast traffic, the egress interface ID for sampled traffic is always 0x80000000. When broadcast and multicast packets are sampled, they are usually forwarded to more than one port. However, the output port field in an sFlow datagram supports the display of one egress interface ID only. Therefore, the sFlow version 5 agent always sets the output port ID to 0x80000000 for broadcast and multicast packets that are sampled.

Specifying the sFlow version format

If your device supports sFlow version 5, you can optionally specify the version used for exporting sFlow data. Refer [“Specifying the sFlow agent IP address”](#).

Specifying the sFlow agent IP address

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the device (the sFlow agent) that sent the data. By default, the device automatically selects the sFlow agent IP address based on the configuration, as described in the section [“sFlow and source address”](#) on page 537. Alternatively, you can configure the device to instead use an arbitrary IPv4 or IPv6 address as the sFlow agent IP address.

To specify an IPv4 address as the sFlow agent IP address, enter a command such as the following

```
Brocade(config)#sflow agent-ip 10.10.10.1
```

Syntax: [no] sflow agent-ip <ipv4-addr>

The <ipv4-addr> specifies the address of the device that sent the data.

To specify an IPv6 address as the sFlow agent IP address, enter a command such as the following.

```
Brocade(config)#sflow agent-ip FE80::240:D0FF:FE48:4672
```

Syntax: [no] sflow agent-ip <ipv6-addr>

The <ipv6-addr> specifies the address of the device that sent the data.

Specifying the version used for exporting sFlow data

By default, when sFlow is enabled globally on the Brocade device, the sFlow agent exports sFlow data in version 5 format. You can change this setting so that the sFlow agent exports data in version 2 format. You can switch between versions without rebooting the device or disabling sFlow.

NOTE

When the sFlow version number is changed, the system will reset sFlow counters and flow sample sequence numbers.

To specify the sFlow version used for exporting sFlow data, enter the following command.

```
Brocade(config)#sflow version 2
```

Syntax: [no] sflow version 2 | 5

The default is 5.

Specifying the maximum flow sample size

With sFlow version 5, you can specify the maximum size of the flow sample sent to the sFlow collector. If a packet is larger than the specified maximum size, then only the contents of the packet up to the specified maximum number of bytes is exported. If the size of the packet is smaller than the specified maximum, then the entire packet is exported.

For example, to specify 1024 bytes as the maximum flow sample size, enter the following command.

```
Brocade(config)# sflow max-packet-size 1024
```

Syntax: [no] sflow max-packet-size <size>

For both sFlow version 2 and version 5, the default maximum flow sample size is 256 bytes.

For sFlow version 5, the maximum flow sample size is 1300 bytes.

Exporting CPU and memory usage information to the sFlow collector

With sFlow version 5, you can optionally configure the sFlow agent on the Brocade device to export information about CPU and memory usage to the sFlow collector.

To export CPU usage and memory usage information, enter the following command.

```
Brocade(config)# sflow export system-info
```

Syntax: [no] sflow export system-info

By default, CPU usage information and memory usage information are not exported.

Specifying the polling interval for exporting CPU and memory usage information to the sFlow collector

The polling interval defines how often sFlow data for a port is sent to the sFlow collector. With sFlow version 5, you can optionally set the polling interval used for exporting CPU and memory usage information.

For example, to set the polling interval for exporting CPU and memory usage information to 30 seconds, enter the following command.

```
Brocade(config)# sflow export system-info 30
```

Syntax: [no] sflow export system-info <seconds>

You can specify a polling interval from 5 seconds to 1,800 seconds (30 minutes). The default polling interval for exporting CPU and memory usage information is 300 seconds (5 minutes).

Exporting CPU-directed data (management traffic) to the sFlow collector

You can select which and how often data destined to the CPU (for example, Telnet sessions) is sent to the sFlow collector.

CLI commands allow you to do the following:

- Enable the sFlow agent to export CPU-directed data
- Specify the sampling rate for exported CPU-directed data

Enabling the sFlow agent to export CPU-directed data

To enable the sFlow agent on a Brocade device to export data destined to the CPU to the sFlow collector, enter the following command.

```
Brocade(config)# sflow export cpu-traffic
```

Syntax: [no] sflow export cpu-traffic

By default, this feature is disabled. The sFlow agent does not send data destined to the CPU to the sFlow collector.

Specifying the sampling rate for exported CPU-directed data

The sampling rate is the average ratio of the number of packets incoming on an sFlow-enabled port, to the number of flow samples taken from those packets. You can optionally set the sampling rate for CPU-directed data exported to the sFlow collector. For example, to set this sampling rate to 2048, enter the following command.

```
Brocade(config)# sflow export cpu-traffic 2048
```

Syntax: [no] sflow export cpu-traffic <rate>

The default sampling rate depends on the Brocade device being configured. Refer to [“Changing the sampling rate”](#) on page 541 for the default sampling rate for each kind of Brocade device.

Displaying sFlow information

To display sFlow configuration information and statistics, enter the following command at any level of the CLI.

```

Brocade#show sflow
sFlow version:5
sFlow services are enabled.
sFlow agent IP address: 123.123.123.1
4 collector destinations configured:
Collector IP 192.168.4.204, UDP 6343
Collector IP 192.168.4.200, UDP 6333
Collector IP 192.168.4.202, UDP 6355
Collector IP 192.168.4.203, UDP 6565
Configured UDP source port: 33333
Polling interval is 0 seconds.
Configured default sampling rate: 1 per 512 packets
Actual default sampling rate: 1 per 512 packets
The maximum sFlow sample size:512
exporting cpu-traffic is enabled
exporting cpu-traffic sample rate:16
exporting system-info is enabled
exporting system-info polling interval:20 seconds
10552 UDP packets exported
24127 sFlow samples collected.
sFlow ports: ethe 1/2 to 1/12 ethe 1/15 ethe 1/25 to 1/26 ethe 4/1 ethe 5/10 to
5/20 ethe 8/1 ethe 8/4
Module Sampling Rates
-----
Slot 1 configured rate=512, actual rate=512
Slot 3 configured rate=0, actual rate=0
Slot 4 configured rate=10000, actual rate=32768
Slot 5 configured rate=512, actual rate=512
Slot 7 configured rate=0, actual rate=0
Slot 8 configured rate=512, actual rate=512
Port Sampling Rates
-----
Port 8/4, configured rate=512, actual rate=512, Subsampling factor=1
Port 8/1, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/20, configured rate=3000, actual rate=8192, Subsampling factor=16
Port 5/19, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/18, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/17, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/16, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/15, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/14, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/13, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/12, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/11, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/10, configured rate=512, actual rate=512, Subsampling factor=1
Port 4/1, configured rate=10000, actual rate=32768, Subsampling factor=1
Port 1/26, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/25, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/15, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/12, configured rate=512, actual rate=512, Subsampling factor=1

...continued on next page...

```

...continued from previous page...

Port 1/11, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/10, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/9, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/8, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/7, configured rate=1000, actual rate=2048, Subsampling factor=4
 Port 1/6, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/5, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/4, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/3, configured rate=512, actual rate=512, Subsampling factor=1
 Port 1/2, configured rate=1000, actual rate=2048, Subsampling factor=4

Syntax: show sflow

The show sflow command displays the following information.

TABLE 99 sFlow information

Parameter	Definition
sFlow version	The version of sFlow enabled on the device, which can be one of the following: <ul style="list-style-type: none"> • 2 • 5
sFlow services	The feature state, which can be one of the following: <ul style="list-style-type: none"> • disabled • enabled
sFlow agent IP address	The IP address that sFlow is using in the agent_address field of packets sent to the collectors. Refer to “sFlow and source address” on page 537.
Collector	The collector information. The following information is displayed for each collector: <ul style="list-style-type: none"> • IP address • UDP port If more than one collector is configured, the line above the collectors indicates how many have been configured.
Configured UDP source port	The UDP source port used to send data to the collector.
Polling interval	The port counter polling interval.
Configured default sampling rate	The configured global sampling rate. If you changed the global sampling rate, the value you entered is shown here. The actual rate calculated by the software based on the value you entered is listed on the next line, “Actual default sampling rate”.
Actual default sampling rate	The actual default sampling rate.
The maximum sFlow sample size	The maximum size of a flow sample sent to the sFlow collector.
exporting cpu-traffic	Indicates whether or not the sFlow agent is configured to export data destined to the CPU (e.g., Telnet sessions) to the sFlow collector: <ul style="list-style-type: none"> • enabled • disabled
exporting cpu-traffic sample rate	The sampling rate for CPU-directed data, which is the average ratio of the number of incoming packets on an sFlow-enabled port, to the number of flow samples taken from those packets.

TABLE 99 sFlow information (Continued)

Parameter	Definition
exporting system-info	Indicates whether or not the sFlow agent is configured to export information about CPU and memory usage to the sFlow collector: <ul style="list-style-type: none"> • enabled • disabled
exporting system-info polling interval	Specifies the interval, in seconds, that sFlow data is sent to the sFlow collector.
UDP packets exported	The number of sFlow export packets the Brocade device has sent. NOTE: Each UDP packet can contain multiple samples.
sFlow samples collected	The number of sampled packets that have been sent to the collectors.
sFlow ports	The ports on which you enabled sFlow.
Module Sampling Rates	The configured and actual sampling rates for each module. If a module does not have any sFlow-enabled ports, the rates are listed as 0.
Port Sampling Rates	The configured and actual sampling rates for each sFlow-enabled port. The Subsampling factor indicates how many times the sampling rate of the port's module is multiplied to achieve the port's sampling rate. Because of the way the actual sampling rates are computed, the Subsampling factors are always whole numbers.

Clearing sFlow statistics

To clear the UDP packet and sFlow sample counters in the **show sflow** display, enter the following command.

```
Brocade#clear statistics
```

Syntax: clear statistics

This command clears the values in the following fields of the **show sflow** display:

- UDP packets exported
- sFlow samples collected

NOTE

This command also clears the statistics counters used by other features.

Utilization list for an uplink port

You can configure uplink utilization lists that display the percentage of a given uplink port bandwidth that is used by a specific list of downlink ports. The percentages are based on 30-second intervals of RMON packet statistics for the ports. Both transmit and receive traffic is counted in each percentage.

NOTE

This feature is intended for ISP or collocation environments in which downlink ports are dedicated to various customers' traffic and are isolated from one another. If traffic regularly passes between the downlink ports, the information displayed by the utilization lists does not provide a clear depiction of traffic exchanged by the downlink ports and the uplink port.

Each uplink utilization list consists of the following:

- Utilization list number (1, 2, 3, or 4)
- One or more uplink ports
- One or more downlink ports

Each list displays the uplink port and the percentage of that port bandwidth that was utilized by the downlink ports over the most recent 30-second interval.

You can configure up to four bandwidth utilization lists.

Utilization list for an uplink port command syntax

To configure an uplink utilization list, enter commands such as the following. The commands in this example configure a link utilization list with port 1/1 as the uplink port and ports 1/2 and 1/3 as the downlink ports.

```
Brocade(config)#relative-utilization 1 uplink eth 1/1 downlink eth 1/2 to 1/3
Brocade(config)#write memory
```

Syntax: [no] **relative-utilization** <num> **uplink ethernet** <port> [to <port> | <port>...] **downlink ethernet** <port> [to <port> | [<port>...]

The <num> parameter specifies the list number. You can configure up to four lists. Specify a number from 1 – 4.

The **uplink ethernet** parameters and the port numbers you specify after the parameters indicate the uplink ports.

The **downlink ethernet** parameters and the port numbers you specify after the parameters indicate the downlink ports.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying utilization percentages for an uplink

After you configure an uplink utilization list, you can display the list to observe the percentage of the uplink bandwidth that each of the downlink ports used during the most recent 30-second port statistics interval. The number of packets sent and received between the two ports is listed, as well as the ratio of each individual downlink port packets relative to the total number of packets on the uplink.

To display an uplink utilization list, enter a command such as the following at any level of the CLI.

```
Brocade#show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 3011
packet count ratio (%)
  1/ 2:60   1/ 3:40
```

In this example, ports 1/2 and 1/3 are sending traffic to port 1/1. Port 1/2 and port 1/3 are isolated (not shared by multiple clients) and typically do not exchange traffic with other ports except for the uplink port, 1/1.

Syntax: `show relative-utilization <num>`

The `<num>` parameter specifies the list number.

NOTE

The example above represents a pure configuration in which traffic is exchanged only by ports 1/2 and 1/1, and by ports 1/3 and 1/1. For this reason, the percentages for the two downlink ports equal 100%. In some cases, the percentages do not always equal 100%. This is true in cases where the ports exchange some traffic with other ports in the system or when the downlink ports are configured together in a port-based VLAN.

In the following example, ports 1/2 and 1/3 are in the same port-based VLAN.

```
Brocade#show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 3011
packet count ratio (%)
  1/ 2:100   1/ 3:100
```

Here is another example showing different data for the same link utilization list. In this example, port 1/2 is connected to a hub and is sending traffic to port 1/1. Port 1/3 is unconnected.

```
Brocade#show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 2996
packet count ratio (%)
  1 /2:100   1/ 3:---
```


Basic Layer 2 Features

Table 100 lists the individual Brocade FastIron switches and the basic Layer 2 features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 100 Supported basic Layer 2 features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
16,000 MAC addresses per switch	Yes	Yes	Yes	Yes	Yes (ICX 6450 only) 8000 for ICX 6430
32,000 MAC addresses per switch	Yes (when flow-based MAC address learning is enabled)	No	Yes	Yes	No
MAC learning rate control	No	Yes	Yes	Yes	Yes
Multi-port static MAC address	Yes	Yes	Yes	Yes	Yes
Static MAC entries with option to set traffic priority	Yes	Yes	Yes	Yes	Yes
Flow-based MAC address learning	Yes To enable, use the mac-learning -flow-based command.	No	Yes Enabled by default on FCX devices. There is no CLI command to enable or disable it.	No	No
Port-based VLANs	Yes	Yes	Yes	Yes	Yes
Address locking (for MAC addresses)	Yes	Yes	Yes	Yes	Yes
MAC address filter override of 802.1X	Yes	Yes	Yes	Yes	Yes
MAC address filtering (filtering on source and destination MAC addresses)	Yes	Yes	Yes	Yes	Yes
MAC address move notification	Yes	Yes	Yes	Yes	Yes
Ability to disable MAC learning	Yes	Yes	Yes	Yes	Yes
Dynamic buffer allocation for QoS priorities	Yes	Yes	Yes	Yes	Yes
Remote Fault Notificatoin (RFN) for 1G fiber	Yes	Yes	Yes	Yes	No

TABLE 100 Supported basic Layer 2 features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Link Fault Signaling (LFS) for 10G	Yes	Yes	Yes	Yes	ICX 6450 only
Layer 2 jumbo frames	Yes	Yes	Yes	Yes	Yes
User-configurable buffer profiles	No	No	Yes	Yes	Yes
Buffer profile for VoIP on FastIron Stackable devices	No	Yes	Yes	Yes	No

The procedures in this chapter describe how to configure basic Layer 2 parameters.

Brocade devices are configured at the factory with default parameters that allow you to begin using the basic features of the system immediately. However, many of the advanced features such as VLANs or routing protocols for the device must first be enabled at the system (global) level before they can be configured. If you use the Command Line Interface (CLI) to configure system parameters, you can find these system level parameters at the Global CONFIG level of the CLI.

NOTE:

- Before assigning or modifying any router parameters, you must assign the IP subnet (interface) addresses for each port.
- For information about configuring IP addresses, DNS resolver, DHCP assist, and other IP-related parameters, refer to [Chapter 26, “IP Configuration”](#).
- For information about the Syslog buffer and messages, refer to [Appendix A, “Syslog messages”](#).

About port regions

This section describes port regions on FastIron switches.

FastIron X Series device port regions

Ports on the FastIron X Series devices are grouped into regions. For a few features, you will need to know the region to which a port belongs. However, for most features, a port region does not affect configuration or operation of the feature.

NOTE

Port regions do not apply to trunk group configurations on the FastIron X Series devices. However, port regions do apply to port monitoring and unknown unicast configurations on FastIron X Series devices.

FastIron Edge Switch X424 and X424HF:

- Ports 1 – 12 belong to port region 0
- Ports 13 – 24 belong to port region 1
- Port 25 belongs to port region 2
- Port 26 belongs to port region 3

FastIron Edge Switch X448:

- Ports 1 – 12 belong to port region 0
- Ports 13 – 24 belong to port region 1
- Ports 25 – 36 belong to port region 2
- Ports 37 – 48 belong to port region 3
- Port 49 belongs to port region 4
- Port 50 belongs to port region 5

FCX and FWS device port regions

The port region rules for FWS and FCX devices are as follows:

- For all platforms, a 24-port Gbps module has one port region. In addition, any 10 Gbps ports on the device also belong to this single port region.
- For all platforms, the 48-port Gbps module has two port regions:
 - Ports 1 - 24 belong to port region 0
 - Ports 25 - 48 belong to port region 1
- For FCX648 devices with two 10 Gbps XFP ports, and a two 10 Gbps CX4 stacking ports:
 - The two 10 Gbps XFP ports belong to port region 0 (along with ports 1 -24)
 - The two 10 Gbps CX4 stacking ports belong to port region 1 (along with ports 25 - 48)
- For FCX648 devices with four 10 Gbps SFP+ ports:
 - 10 Gbps SFP+ ports 3 and 4 belong to port region 0 (along with ports 1 -24)
 - 10 Gbps SFP+ ports 1 and 2 ports belong to port region 1 (along with ports 25 - 48)

ICX 6610 device port regions

- A 24-port Gbps has the following port regions:
 - Ports 1-6 and SFP ports 1-4 belong to region 0
 - Ports 7-12 and SFP ports 5-8 belong to region 1
 - Ports 13-18 belong to region 2
 - Ports 19-24 belong to region 3
- A 48-port Gbps has the following port regions:
 - Ports 1-6 and 25-30 and SFP ports 1-4 belong to region 0
 - Ports 7-12 and 31-36 and SFP ports 5-8 belong to region 1
 - Ports 13-18 and 37-42 belong to region 2
 - Ports 19-24 and 43-45 belong to region 3

ICX 6430 device port regions

- A 24-port Gbps module has one port region. The four SFP ports on the device also belong to this single port region.

- A 48-port Gbps module has two port regions:
 - Ports 1-24 and SFP ports 1& 2 belong to port region 0
 - Ports 25-48 and SFP ports 3 & 4 belong to port region 1

ICX 6450 device port regions

- A 24-port Gbps module has one port region. The four SFP+ ports on the device also belong to this single port region.
- A 48-port Gbps module has two port regions:
 - Ports 1-24 and SFP+ ports 1& 2 belong to port region 0
 - Ports 25-48 and SFP+ ports 3 & 4 belong to port region 1

Enabling or disabling the Spanning Tree Protocol (STP)

STP (IEEE 802.1D bridge protocol) is supported on all Brocade devices. STP detects and eliminates logical loops in the network. STP also ensures that the least cost path is taken when multiple paths exist between ports or VLANs. If the selected path fails, STP searches for and then establishes an alternate path to prevent or limit retransmission of data.

NOTE

This section provides instructions for enabling and disabling STP. For configuration procedures and more information about STP, refer to [Chapter 27, “Spanning Tree Protocol”](#) in this guide.

STP must be enabled at the system level to allow assignment of this capability on the VLAN level. On devices running Layer 2 code, STP is enabled by default. On devices running Layer 3 code, STP is disabled by default.

To enable STP for all ports on a Brocade device, enter the following command.

```
Brocade(config)#spanning tree
```

Syntax: [no] spanning-tree

You can also enable and disable spanning tree on a port-based VLAN and on an individual port basis, and enable advanced STP features. Refer to [Chapter 27, “Spanning Tree Protocol”](#).

Modifying STP bridge and port parameters

You can modify the following STP Parameters:

- Bridge parameters – forward delay, maximum age, hello time, and priority
- Port parameters – priority and path cost

For configuration details, refer to [“Changing STP bridge and port parameters”](#) on page 1091.

Management MAC address for stackable devices

In an IronStack, the management MAC address of the Active Controller is always used as the STP bridge ID. The Active Controller management MAC address is always used for control protocols for the following reasons:

- Unlike standalone devices, each stack member has a different range of MAC addresses.
- In a stack, the management MAC address is software generated, and is always the MAC address of the Active Controller first port. This ensures consistency across the stack during resets, assuming that the Active Controller is always the same unit.
- This helps avoid the disruption of frequent topology changes in the stack.

For more information about stacking and Brocade stackable devices, see [Chapter 7, “Brocade Stackable Devices”](#).

MAC learning rate control

NOTE

The MAC learning rate control feature is not supported on FastIron X Series devices.

You can set a rate limit to control CPU address updating. The range for this rate limit is 200 to 50,000 per second. The MAC learning rate limit applies to each packet processor, which means that for a system with two packet processors, each processor can send address messages to the CPU at the established rate limit.

Syntax: `[no] cpu-limit addr-msgs <msgsRateLimit>`

NOTE

Actual rates in hardware may have a variance of +200 or -100.

Changing the MAC age time and disabling MAC address learning

To change the MAC address age timer, enter a command such as the following.

```
Brocade(config)#mac-age-time 60
```

Syntax: [no] **mac-age-time** <secs>

<secs> specifies the number of seconds. Possible values differ depending on the version of software running on your device, as follows:

- On FastIron WS and Brocade FCX Series devices, learned MAC address entries do not age out until they are unused for 300 – 600 seconds. If necessary, you can change the MAC address age timer to 0 or a value from 60 – 600 (seconds), in 60-second intervals. For example, you can specify 60 or 120, but not 100. If you set the MAC age time to 0, aging is disabled.
- On FastIron X Series devices, you can configure 0 or a value from 10 – 86,400 (seconds), in 10-second intervals. If you set the MAC age time to 0, aging is disabled.
- On ICX devices, you can configure a value between 0 to 570 seconds.

NOTE: Usually, the actual MAC age time is from one to two times the configured value. For example, if you set the MAC age timer to 60 seconds, learned MAC entries age out after remaining unused for between 60 – 120 seconds. However, if all of the following conditions are met, then the MAC entries age out after a longer than expected duration:

- The MAC age timer is greater than 630 seconds.
- The number of MAC entries is over 6000.
- All MAC entries are learned from the same packet processor.
- All MAC entries age out at the same time.

Disabling the automatic learning of MAC addresses

By default, when a packet with an unknown Source MAC address is received on a port, the Brocade device learns this MAC address on the port.

You can prevent a physical port from learning MAC addresses by entering the following command.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e1000-3/1)#mac-learn-disable
```

Syntax: [no] **mac-learn disable**

Use the no form of the command to allow a physical port to learn MAC addresses.

MAC address learning configuration notes and feature limitations

- This command is not available on virtual routing interfaces. Also, if this command is configured on the primary port of a trunk, MAC address learning will be disabled on all the ports in the trunk.
- Entering the **mac-learn-disable** command on tagged ports disables MAC learning for that port in all VLANs to which that port is a member. For example, if tagged port 3/1 is a member of VLAN 10, 20, and 30 and you issue the **mac-learn-disable** command on port 3/1, port 3/1 will not learn MAC addresses, even if it is a member of VLAN 10, 20, and 30.

Displaying the MAC address table

To display the MAC table, enter the **show mac-address** command.

```
Brocade#show mac-address
Total active entries from all ports = 3
Total static entries from all ports = 1
  MAC-Address      Port      Type      VLAN
1234.1234.1234      15      Static      1
0004.8038.2f24      14      Dynamic      1
0004.8038.2f00      13      Dynamic      1
0010.5a86.b159      10      Dynamic      1
```

In the output of the **show mac-address** command, the *Type* column indicates whether the MAC entry is static or dynamic. A static entry is one you create using the **static-mac-address** command. A dynamic entry is one that is learned by the software from network traffic.

The output of the **show mac-address** command on FESX and FSX devices include an *Index* column which indicates the index where the entry exists in the hardware MAC table.

NOTE

The **show mac-address** command output does not include MAC addresses for management ports, since these ports do not support typical MAC learning and MAC-based forwarding.

Static MAC entry configuration

Static MAC addresses can be assigned to Brocade devices.

NOTE

Brocade devices running Layer 3 code also support the assignment of static IP Routes, static ARP, and static RARP entries. For details on configuring these types of static entries, refer to [“Adding a static IP route”](#) on page 1186 and [“Adding a static ARP entry”](#) on page 1187.

You can manually input the MAC address of a device to prevent it from being aged out of the system address table.

This option can be used to prevent traffic for a specific device, such as a server, from flooding the network with traffic when it is down. Additionally, the static MAC address entry is used to assign higher priorities to specific MAC addresses.

You can specify traffic priority (QoS) and VLAN membership (VLAN ID) for the MAC Address as well as specify the device type of either router or host.

The default and maximum configurable MAC table sizes can differ depending on the device. To determine the default and maximum MAC table sizes for your device, display the system parameter values. Refer to [“Displaying and modifying system parameter default settings”](#) on page 578.

Multi-port static MAC address

Many applications, such as Microsoft NLB, Juniper IPS, and Netscreen Firewall, use the same MAC address to announce load-balancing services. As a result, a switch must be able to learn the same MAC address on several ports. Multi-port static MAC allows you to statically configure a MAC address on multiple ports using a single command.

Multi-port static MAC address configuration notes

- This feature is applicable for Layer 2 traffic.
- This feature can be used to configure unicast as well as IPv4 and IPv6 multicast MAC addresses on one or more ports. However, when a multicast MAC address is configured, the corresponding MAC address entry cannot be used for IGMP snooping. For IPv4 multicast addresses (range 0100.5e00.000 to 0100.5e7f.ffff) and IPv6 multicast addresses (range 3333.0000.0000 to 3333.ffff.ffff), use IGMP/MLD snooping. Other multicast addresses can also be configured on the ports using this feature.
- FastIron devices support a maximum of 15 multi-port static MAC addresses.
- Hosts or physical interfaces normally join multicast groups dynamically, but you can also statically configure a host or an interface to join a multicast group.

Configuring a multi-port static MAC address

For example, to add a static entry for a server with a MAC address of 0045.5563.67ff and a priority of 7, enter the following command.

```
Brocade(config)#static-mac-address 0045.5563.67ff ethernet 4/2 ethernet 4/3
ethernet 4/4 priority 7
```

To specify a range of ports, enter the following command.

```
Brocade(config)#static-mac-address 0045.5563.67ff ethernet 4/2 to 4/6 priority 7
```

Syntax: [no] **static-mac-address** <mac-addr> **ethernet** [<slotnum>/]<portnum> **ethernet** [<slotnum>/]<portnum> **ethernet** [<slotnum>/]<portnum> [**priority** <num>]

or

Syntax: [no] **static-mac-address** <mac-addr> **ethernet** [<slotnum>/]<portnum> **to ethernet** [<slotnum>/]<portnum> [**priority** <num>]

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The priority <num> is optional and can be a value from 0 – 7 (0 is lowest priority and 7 is highest priority). The default priority is 0.

NOTE

The location of the **static-mac-address** command in the CLI depends on whether you configure port-based VLANs on the device. If the device does not have more than one port-based VLAN (VLAN 1, which is the default VLAN that contains all the ports), the **static-mac-address** command is at the global CONFIG level of the CLI. If the device has more than one port-based VLAN, then the **static-mac-address** command is not available at the global CONFIG level. In this case, the command is available at the configuration level for each port-based VLAN.

VLAN-based static MAC entries configuration

You can configure a VLAN to drop packets that have a particular source or destination MAC address.

You can configure a maximum of 2048 static MAC address drop entries on a Brocade device.

Use the CLI command **show running-config** to view the static MAC address drop entries currently configured on the device.

Configuring a VLAN to drop static MAC entries

To configure a VLAN to drop packets with a source or destination MAC address of 1145.5563.67FF, enter the following commands.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#static-mac-address 1145.5563.67FF drop
```

Syntax: [no] **static-mac-address** <mac-addr> **drop**

Use the **no** form of the command to remove the static MAC address drop configuration.

Clearing MAC address entries

You can remove learned MAC address entries from the MAC address table. The types of MAC address that can be removed are as follows:

- All MAC address entries
- All MAC address entries for a specified Ethernet port
- All MAC address entries for a specified VLAN
- All specified MAC address entry in all VLANs

For example, to remove entries for the MAC address 000d.cd80.00d0 in all VLANs, enter the following command at the Privilege EXEC level of the CLI.

```
Brocade#clear mac-address 000d.cb80.00d0
```

Syntax: **clear mac-address** <mac-address> | **ethernet** <port-num> | **vlan** <vlan-num>

If you enter **clear mac-address** without any parameter, the software removes all MAC address entries.

Use the <mac-address> parameter to remove a specific MAC address from all VLANs. Specify the MAC address in the following format: HHHH.HHHH.HHHH.

Use the **ethernet** <port-num> parameter to remove all MAC addresses for a specific Ethernet port.

Use the **vlan** <num> parameter to remove all MAC addresses for a specific VLAN.

Flow-based MAC address learning

NOTE

Flow-based MAC address learning is supported on FastIron X Series and Brocade FCX Series devices. It does not apply to FastIron WS Series devices. However, on Brocade FCX Series devices, this feature is enabled by default. There is no command to enable or disable it. Therefore, the CLI commands in this section apply to FastIron X Series devices only.

This section describes flow-based MAC address learning and how to enable it on a FastIron X Series switch.

Flow-based learning overview

With regular MAC address learning, when a new MAC address is learned, it is programmed in the same location (hardware index) in all packet processors in a FastIron Layer 2 or Layer 3 switch. There are multiple packet processors (one per port region) in a compact switch, and in each module in a chassis-based switch. With regular MAC address learning, MAC addresses are *global*, meaning the hardware MAC table is identical across all packet processors.

With the introduction of **flow-based MAC address learning**, when a new source MAC address is learned, it is programmed only in the source packet processor (the processor that received the packet). The destination MAC address gets added to other packet processors on demand, whenever a traffic flow that needs it is detected. With flow-based MAC address learning, the MAC address is programmed in different hardware locations and the hardware MAC table is different across all packet processors.

The benefits of flow-based learning

With global MAC address learning, all MAC addresses are programmed in all packet processors, even though they may not be required and are never used by all packet processors. Global MAC address learning wastes some space in the hardware MAC table and limits the number of supported MAC addresses to 16K.

With flow-based MAC address learning, MAC addresses are learned and programmed selectively, only in the packet processors that need them. Since the MAC addresses are distributed across several packet processors, flow-based learning frees up space in the hardware MAC table and increases the number of supported MAC addresses from 16K to 32K.

How flow-based learning works

When a packet processor, for example, *PP 1*, receives an incoming packet with source MAC address *X*, it sends a *new address* message to the CPU. The system learns MAC address *X* by adding it to the software MAC table in the CPU, then programming it in the hardware MAC table in the *source* packet processor, in this case *PP 1*. If the MAC address is learned on a trunk port, the MAC address is also programmed on all of the packet processors that have ports in the same trunk group.

When another packet processor, let call it *PP 2*, receives an incoming packet and the packet destination MAC address matches *source MAC address X*, it floods the packet in hardware as an unknown unicast packet and copies the packet to the CPU. The system locates the MAC address in the software MAC table, then programs the MAC address in the hardware MAC table in *PP 2*. If the

MAC address is learned on a trunk port, the MAC address is also programmed on all of the packet processors that have ports in the same trunk group. Once the MAC address is programmed in hardware, subsequent packets with this destination MAC are forwarded as known unicast packets and are not copied to the CPU.

Flow-based MAC addresses are aged out by the source packet processor according to the MAC age time learned on the local port. Furthermore, when a flow-based MAC address is aged out from the source packet processor, it is also aged out from all other packet processors on which the address is programmed. In the above example, when MAC address X is aged out from *PP 1*, it is also aged out from *PP2*.

NOTE

Even when flow-based MAC address learning is enabled, some MAC addresses, including but not limited to control MACs, static MACs, multicast MACs, and MAC addresses resolved through ARP, will continue to be global MAC addresses. These MAC addresses are always programmed in all packet processors in a Layer 2 or Layer 3 switch.

NOTE

Global MAC addresses have priority over dynamic flow-based MAC addresses. To ensure that global MAC addresses are in sync across all packet processors, flow-based MAC addresses may be overwritten in one or more packet processors. The MAC addresses will be relearned and reprogrammed using the flow-based method as needed by incoming traffic flows.

Flow-based learning configuration considerations

When configuring flow-based MAC learning, consider the rules and limitations in this section.

- Flow-based MAC learning is not supported with the following features:
 - Disabling the automatic learning of MAC addresses (CLI command **mac-learn-disable**).
 - Globally disabling Layer 2 switching (CLI command **route-only**)
- When flow-based MAC learning is enabled, unknown unicast packets are copied to the CPU. Therefore, flow-based MAC learning should not be enabled if a continuous high rate of unknown unicast packet flooding is expected, as this will cause high CPU utilization.
- Unknown unicast flooding can occur for a known destination MAC address, if the system fails to program that destination MAC address because the hardware MAC table or hash bucket is full. This condition can also lead to high CPU utilization.
- A source MAC address is learned only on the ingress (source) packet processor. The MAC address is added to other packet processors as needed by their incoming traffic flows. During a brief period until the destination MAC address is successfully added to the hardware MAC table, unknown unicast flooding is expected on the VLAN.
- When a flow-based MAC address moves, it is deleted from all of the packet processors, then relearned on each packet processor individually, as needed by incoming traffic flows.
- The software MAC address table in the CPU uses a hashing algorithm. Because hash collisions can occur and may consume software resources, the FastIron may not be able to support up to 32K MAC addresses.
- The system can scale up to 32K MAC addresses, however, each packet processor is limited to a maximum of 16K MAC addresses. This limit still applies, as this is a hardware limitation.

Configuring flow-based MAC address learning

To configure flow-based MAC address learning, simply enable it globally. If necessary, increase the capacity of the MAC address table as well.

Enabling flow-based MAC address learning

To enable flow-based MAC address learning, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)#mac-learning-flow-based
```

This command enables flow-based MAC address learning. All dynamically-learned MAC addresses are flushed from the hardware and software MAC tables and are subsequently learned using flow-based MAC address learning.

Syntax: [no] mac-learning-flow-based

Use the **no** form of the command to disable flow-based MAC address learning. When disabled, all dynamically-learned MAC addresses are flushed from the hardware and software MAC tables and are subsequently learned using global MAC address learning.

Increasing the capacity of the MAC address table (optional)

After enabling support for flow-based MACs, you can increase the capacity of the MAC address table of up to 32K MAC addresses. By default, up to 32K MAC addresses are supported.

NOTE

FCX devices do not support flow-based MACs and the capacity of the MAC address table cannot be altered on FCX devices.

To increase the capacity of the MAC table, enter commands such as the following.

```
Brocade(config)#system-max mac 32768
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
```

NOTE

You must save the configuration and reload the software to place the **system-max mac** change into effect.

Syntax: system-max mac <max-flow-MACs>

The <max-flow-MACs> parameter specifies the maximum number of MAC addresses in the MAC table. For flow-based MACs, the minimum value is 16K and the default value is 32K.

Use the command **show default values** to display the default, maximum, and currently configured values for the MAC address table.

Displaying information about flow-based MACs

The **show mac-address** command includes information related to flow-based MAC address learning. The following shows an example **show mac** output.

```
Brocade# show mac
Total active entries from all ports = 15
MAC-Address      Port          Type          Index
0000.0000.0001   1/1           Dynamic       NA
0000.0000.0002   1/1           Dynamic       NA
```

In the above example, since both MAC address entries are flow-based and are located on different packet processors (hardware index), the **Index** field displays **NA** (not applicable).

Syntax: show mac

To display all of the packet processors that have a particular flow-based MAC address, use the **show mac-address vlan** command.

```
Brocade#show mac-address vlan 1 0000.0000.0001
Total active entries from all ports = 16
MAC-Address      Port          Type          Index
0000.0000.0001   1/1           Dynamic       NA
```

```
Present in following devices (at hw index) :-
0  (8196 )          4  (8196 )
```

In the above example, the MAC address 0000.0000.0001 is programmed in packet processors 0 and 4, and the hardware index is 8196.

Syntax: show mac-address vlan <vlan-num> <mac address>

Clearing flow-based MAC address entries

To remove dynamically-learned MAC addresses from the MAC table, use the CLI command **clear mac**. This command clears all dynamically-learned MACs from the hardware and software MAC tables.

Enabling port-based VLANs

When using the CLI, port and protocol-based VLANs are created by entering one of the following commands at the global CONFIG level of the CLI.

To create a port-based VLAN, enter commands such as the following.

```
Brocade(config)#vlan 222 by port
Brocade(config)#vlan 222 name Mktg
```

Syntax: vlan <num> by port

Syntax: vlan <num> name <string>

The **<num>** parameter specifies the VLAN ID. The valid range for VLAN IDs starts at 1 on all systems but the upper limit of the range differs depending on the device. In addition, you can change the upper limit on some devices using the **system max-vlans...** command.

The `<string>` parameter is the VLAN name and can be a string up to 32 characters. You can use blank spaces in the name if you enclose the name in double quotes (for example, “Product Marketing”).)

You can configure up to 4063 port-based VLANs on a device running Layer 2 code or 4061 port-based VLANs on a device running Layer 3 code. Each port-based VLAN can contain either tagged or untagged ports. A port cannot be a member of more than one port-based VLAN unless the port is tagged. On both device types, valid VLAN IDs are 1 – 4095. You can configure up to the maximum number of VLANs within that ID range.

NOTE

VLAN IDs 4087, 4090, and 4093 are reserved for Brocade internal use only. VLAN 4094 is reserved for use by Single STP. Also, if you are running an earlier release, VLAN IDs 4091 and 4092 may be reserved for Brocade internal use only. If you want to use VLANs 4091 and 4092 as configurable VLANs, you can assign them to different VLAN IDs. For more information, refer to [“Assigning different VLAN IDs to reserved VLANs 4091 and 4092”](#) on page 767.

NOTE

The second command is optional and also creates the VLAN if the VLAN does not already exist. You can enter the first command after you enter the second command if you first exit to the global CONFIG level of the CLI.

Assigning IEEE 802.1Q tagging to a port

When a port is tagged, it allows communication among the different VLANs to which it is assigned. A common use for this might be to place an email server that multiple groups may need access to on a tagged port, which in turn, is resident in all VLANs that need access to the server.

NOTE

Tagging does not apply to the default VLAN.

When using the CLI, ports are defined as either tagged or untagged at the VLAN level.

Command syntax for assigning 802.1Q tagging to a port

Suppose you want to make port 5 a member of port-based VLAN 4, a tagged port. To do so, enter the following.

```
Brocade(config)#vlan 4
Brocade(config-vlan-4)#tagged e 5
```

Syntax: `tagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum>] [ethernet [<slotnum>/]<portnum>...]`

The `<slotnum>` parameter is required on chassis devices.

Defining MAC address filters

MAC layer filtering enables you to build access lists based on MAC layer headers in the Ethernet/IEEE 802.3 frame. You can filter on the source and destination MAC addresses. The filters apply to incoming traffic only.

You configure MAC address filters globally, then apply them to individual interfaces. To apply MAC address filters to an interface, you add the filters to that interface MAC address filter group.

The device takes the action associated with the first matching filter. If the packet does not match any of the filters in the access list, the default action is to drop the packet. If you want the system to permit traffic by default, you must specifically indicate this by making the last entry in the access list a permit filter. An example is given below.

Syntax: `mac filter <last-index-number> permit any any.`

For devices running Layer 3 code, the MAC address filter is applied to all inbound Ethernet packets, including routed traffic. This includes those port associated with a virtual routing interface. However, the filter is not applied to the virtual routing interface. It is applied to the physical port.

When you create a MAC address filter, it takes effect immediately. You do not need to reset the system. However, you do need to save the configuration to flash memory to retain the filters across system resets.

MAC address filters configuration notes and limitations

- MAC address filtering on FastIron devices is performed in hardware.
- MAC address filtering on FastIron devices differ from other Brocade devices in that you can only filter on source and destination MAC addresses. Other Brocade devices allow you to also filter on the encapsulation type and frame type.
- MAC address filtering applies to all traffic, including management traffic. To exclude management traffic from being filtered, configure a MAC address filter that explicitly permits all traffic headed to the management MAC (destination) address. The MAC address for management traffic is always the MAC address of port 1.
- MAC address filters that have a global **deny** statement can cause the device to block all BPDUs. In this case, include exception statements for control protocols in the MAC address filter configuration.

The following configuration notes apply to Brocade Layer 3 devices:

- MAC address filters apply to both switched and routed traffic. If a routing protocol (for example, OSPF) is configured on an interface, the configuration must include a MAC address filter rule that allows the routing protocol MAC and the neighbor system MAC address.
- You cannot use MAC address filters to filter Layer 4 information.
- MAC address filters are supported on tagged ports in the base Layer 3, edge Layer 3, and full Layer 3 software images.

MAC address filters command syntax

To configure and apply a MAC address filter, enter commands such as the following.

```
Brocade(config)# mac filter 1 deny 3565.3475.3676 ffff.0000.0000
Brocade(config)# mac filter 2 deny any ffff.ffff.ffff ffff.ffff.ffff
Brocade(config)# mac filter 3 deny any 0180.c200.0000 ffff.ffff.fff0
Brocade(config)# mac filter 4 deny any 0000.1234.5678 ffff.ffff.ffff
Brocade(config)# mac filter 5 deny any 0000.2345.6789 ffff.ffff.ffff
Brocade(config)# mac filter 1024 permit any any
Brocade(config)# int e 1
Brocade(config-if-e1000-1)# mac filter-group 1 to 5 1024
```

These commands configure filter 1 to deny traffic with a source MAC address that begins with “3565” to any destination, and configure filters 2 through 5 to deny traffic with the specified destination MAC addresses. Filter 1024 permits all traffic that is not denied by any other filter.

NOTE

Once you apply a MAC address filter to a port, the device drops all Ethernet traffic on the port that does not match a MAC permit filter on the port.

Syntax: `[no] mac filter <filter-num> permit | deny <src-mac> <mask> | any <dest-mac> <mask> | any`

You can configure up to 507 MAC filters for `<filter-num>`, although the output of the **show default values** command shows 512.

The **permit | deny** argument determines the action the software takes when a match occurs.

The `<src-mac> <mask> | any` parameter specifies the source MAC address. You can enter a specific address value and a comparison mask or the keyword **any** to filter on all MAC addresses. Specify the mask using f (ones) and zeros. For example, to match on the first two bytes of the address aabb.ccdd.eeff, use the mask ffff.0000.0000. In this case, the filter matches on all MAC addresses that contain "aabb" as the first two bytes. The filter accepts any value for the remaining bytes of the MAC address. If you specify **any**, do not specify a mask. In this case, the filter matches on all MAC addresses.

The `<dest-mac> <mask> | any` parameter specifies the destination MAC address. The syntax rules are the same as those for the `<src-mac> <mask> | any` parameter.

Syntax: `[no] mac filter log-enable`

Globally enables logging for filtered packets.

Syntax: `[no] mac filter-group log-enable`

Enables logging for filtered packets on a specific port.

Syntax: `[no] mac filter-group <filter-number> [to <filter-number> | <filter-number>...]`

Applies MAC address filters to a port.

When applying the filter-group to the interface, specify each line to be applied separately or use the **to** keyword to apply a consecutive range of filter lines, for example, 1 3 to 8 10.

NOTE

The filters must be applied as a group. For example, if you want to apply four filters to an interface, they must all appear on the same command line.

NOTE

You cannot add or remove individual filters in the group. To add or remove a filter on an interface, apply the filter group again containing all the filters you want to apply to the port.

NOTE

If you apply a filter group to a port that already has a filter group applied, the older filter group is replaced by the new filter group.

When a MAC address filter is applied to or removed from an interface, a Syslog message such as the following is generated.

```
SYSLOG: <14>Jan 1 00:00:00 10.44.9.11 MAC Filter applied to port 0/1/2 by tester
from telnet session (filter id=5 ).
SYSLOG: <14>Jan 1 00:00:00 10.44.9.11 MAC Filter removed from port 0/1/2 by tester
from telnet session (filter id=5 ).
```

The Syslog messages indicate that a MAC address filter was applied to the specified port by the specified user during the specified session type. Session type can be Console, Telnet, SSH, Web, SNMP, or others. The filter IDs that were added or removed are listed.

Enabling logging of management traffic permitted by MAC address filters

You can configure the Brocade device to generate Syslog entries and SNMP traps for management traffic that is permitted by MAC address filters. **Management traffic** applies to packets that are destined for the CPU, such as control packets. You can enable logging of permitted management traffic on a global basis or an individual port basis.

The first time an entry in a MAC address filter permits a management packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for management packets permitted by MAC address filters are at the warning level of the Syslog.

When the first Syslog entry for a management packet permitted by a MAC address filter is generated, the software starts a five-minute timer. After this, the software sends Syslog messages every five minutes. The messages list the number of management packets permitted by each MAC address filter during the previous five-minute interval. If a MAC address filter does not permit any packets during the five-minute interval, the software does not generate a Syslog entry for that MAC address filter.

NOTE

For a MAC address filter to be eligible to generate a Syslog entry for permitted management packets, logging must be enabled for the filter. The Syslog contains entries only for the MAC address filters that permit packets and have logging enabled.

When the software places the first entry in the log, the software also starts the five-minute timer for subsequent log entries. Thus, five minutes after the first log entry, the software generates another log entry and SNMP trap for permitted management packets.

MAC address filter logging command syntax

To configure MAC address filter logging globally, enter the following CLI commands at the global CONFIG level.

```
Brocade(config)#mac filter log-enable
Brocade(config)#write memory
```

Syntax: [no] mac filter log-enable

To configure MAC address filter logging for MAC address filters applied to ports 1 and 3, enter the following CLI commands.

```
Brocade(config)#int ethernet 1
Brocade(config-if-e1000-1)#mac filter-group log-enable
Brocade(config-if-e1000-1)#int ethernet 3
Brocade(config-if-e1000-3)#mac filter-group log-enable
Brocade(config-if-e1000-3)#write memory
```

Syntax: [no] mac filter-group log-enable**MAC address filter override for 802.1X-enabled ports**

The MAC address filtering feature on an 802.1X-enabled port allows 802.1X and non-802.1X devices to share the same physical port. For example, this feature enables you to connect a PC and a non-802.1X device, such as a Voice Over IP (VOIP) phone, to the same 802.1X-enabled port on the Brocade device. The IP phone will bypass 802.1X authentication and the PC will require 802.1X authentication.

To enable this feature, first create a MAC address filter, then bind it to an interface on which 802.1X is enabled. The MAC address filter includes a mask that can match on any number of bytes in the MAC address. The mask can eliminate the need to enter MAC addresses for all non-802.1X devices connected to the Brocade device, and the ports to which these devices are connected.

MAC address filter override configuration notes

- This feature is supported on untagged, tagged, and dual-mode ports.
- You can configure this feature on ports that have ACLs and MAC address filters defined.

MAC address filter override configuration syntax

To configure MAC address filtering on an 802.1X-enabled port, enter commands such as the following.

```
Brocade#(config)#mac filter 1 permit 0050.04ab.9429 ffff.ffff.0000 any
Brocade#(config)#int e1/2
Brocade#(config-if-e1000-1/2)#dot1x auth-filter 1 3 to 5 10
```

The first line defines a MAC address filter that matches on the first four bytes (ffff.ffff.0000) of the source MAC address 0050.04ab.9429, and any destination MAC address. The permit action creates an 802.1X session in the FORCE AUTHORIZE state, meaning that the device is placed unconditionally in the authorized state, bypassing 802.1X authentication and allowing all traffic from the specified MAC address. If no match is found, the implicit action is to authenticate the client.

The last line binds MAC address filters 1, 3, 4, 5, and 10 to interface 2.

Syntax: `mac filter <filter-num> permit | deny <src-mac> <mask> | any <dest-mac> <mask> | any`

Syntax: `dot1x auth-filter <filter-list>`

The **permit** | **deny** argument determines the action the software takes when a match occurs. In the previous example, the *permit* action creates an 802.1X session in the FORCE AUTHORIZE state, meaning that the device is placed unconditionally in the authorized state, bypassing 802.1X authentication and allowing all traffic from the specified MAC address. The *deny* action creates an 802.1X session in the FORCE UNAUTHORIZE state, meaning that the device will never be authorized, even if it has the appropriate credentials.

The `<src-mac> <mask> | any` parameter specifies the source MAC address. You can enter a specific address value and a comparison mask, or the keyword **any** to filter on all MAC addresses. Specify the mask using f (ones) and zeros. For example, to match on the first two bytes of the address aabb.ccdd.eeff, use the mask ffff.0000.0000. The filter matches on all MAC addresses that contain aabb as the first two bytes and accepts any value for the remaining bytes of the MAC address. If you specify **any**, do not specify a mask. In this case, the filter matches on all MAC addresses. If no match is found, the implicit action is to authenticate the client.

The `<dest-mac> <mask> | any` parameter specifies the destination MAC address. The syntax rules are the same as those for the `<src-mac> <mask> | any` parameter. Note that the 802.1x Authentication filter (**dot1x auth-filter**) does not use the destination MAC address in the MAC address filter.

The `<filter-num>` command identifies the MAC address filter. The maximum number of supported MAC address filters is determined by the **mac-filter-sys** default or configured value.

The **dot1x auth-filter** `<filter-list>` command binds MAC address filters to a port.

The following rules apply when using the **dot1x auth-filter** command:

- When you add filters to or modify the **dot1x auth-filter**, the system clears all 802.1X sessions on the port. Consequently, all users that are logged in will need to be re-authenticated.
- The maximum number of filters that can be bound to a port is limited by the **mac-filter-port** default or configured value.
- The filters must be applied as a group. For example, if you want to apply four filters to an interface, they must all appear on the same command line.
- You cannot add or remove individual filters in the group. To add or remove a filter on an interface, apply the filter group again containing all the filters you want to apply to the port.

If you apply a filter group to a port that already has a filter group applied, the older filter group is replaced by the new filter group.

Locking a port to restrict addresses

Address-lock filters allow you to limit the number of devices that have access to a specific port. Access violations are reported as SNMP traps. This feature is disabled by default. A maximum of 2048 entries can be specified for access. The default address count is eight.

Lock address configuration notes

- Static trunk ports and link-aggregation configured ports do not support the lock-address option.
- The MAC port security feature is a more robust version of this feature. Refer to [“MAC port security configuration”](#) on page 1839.

Lock address command syntax

To enable address locking for port 2 and place a limit of 15 entries, enter a command such as the following.

```
Brocade(config)#lock e 2 addr 15
```

Syntax: `lock-address ethernet [<port> [addr-count <num>]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The *<num>* parameter is a value from 1 – 2048.

Monitoring MAC address movement

MAC address movement notification allows you to monitor the movement of MAC addresses that migrate from port to port. It enables you to distinguish between legitimate movement and malicious movement by allowing you to define malicious use as a threshold number of times a MAC address moves within a specific interval.

Malicious use typically involves many MAC address moves, while legitimate use usually involves a single move. Malicious movement is often the result of MAC address spoofing, in which a malicious user masquerades as a legitimate user by changing his own MAC address to that of a legitimate user. As a result, the MAC address moves back and forth between the ports where the legitimate and malicious users are connected. A legitimate use might be to spoof the MAC address of a failed device in order to continue access using a different device.

You can monitor MAC address movements in the following ways:

- Threshold-rate notifications allow you to configure the maximum number of movements over a specified interval for each MAC address before a notification is sent. For example you could define the malicious move rate as three moves every 30 seconds.

- Interval-history notifications are best suited for a statistical analysis of the number of MAC address movements for a configured time interval. For example, you may want to find out how many MAC addresses have moved in the system over a given interval or how many times a specific MAC address has moved during that interval. However, it is not possible to get this information for every MAC address if there are a lot of MAC addresses that moved during the interval. Consequently, the number of MAC addresses that can have a recorded history is limited.

NOTE

MAC address move notification does not detect MAC movements across an MCT cluster between MCT peers. It only detects MAC movements locally within a cluster MCT peer.

Configuring the MAC address movement threshold rate

To enable notification of MAC address moves, enter the **mac-movement notification threshold-rate** command at the global configuration level. This command enables a corresponding SNMP trap. Notification is triggered when a threshold number of MAC address moves occurs within a specified period for the same MAC address. This command sets the threshold level and the sampling interval.

Avoid threshold rates and sampling intervals that are too small. If you choose a small threshold and a sampling interval that is also small, an unnecessarily high number of traps could occur.

The following example enables notification of MAC address moves and sends an SNMP trap when any MAC address moves to a different port five times in a 10-second interval.

```
Brocade(config)# mac-movement notification threshold-rate 5 sampling-interval 10
```

To disable notification of MAC address moves and disable the SNMP trap, use the **no** form of the command, as shown in the following example.

```
Brocade(config)# no mac-movement notification threshold-rate 5 sampling-interval 10
```

Syntax: **[no] mac-movement notification threshold-rate** *<move-count>* **sampling-interval** *<interval>*

The *<move-count>* variable indicates the number of times a MAC address can move within the specified period until an SNMP trap is sent. It has no default value.

The *<interval>* variable specifies the sampling period in seconds. It has no default value.

Viewing the MAC address movement threshold rate configuration

To display the configuration of the MAC address movement threshold rate, enter the **show notification mac-movement threshold-rate** command at the privileged EXEC level. This command also displays ongoing statistics for the current sampling interval.

```
Brocade# show notification mac-movement threshold-rate
Threshold-Rate Mac Movement Notification is ENABLED
Configured Threshold-Rate : 5 moves
Configured Sampling-Interval : 30 seconds
Number of entries in the notification table : 100
MAC-Address      from-Port    to-Port          Last Move-Time    Vlan-id
-----
0010.9400.0022   7/1          7/2             5 minutes 30 seconds    10
0010.9400.0021   7/1          7/2             5 minutes 30 seconds    10
0010.9400.0020   7/1          7/2             5 minutes 30 seconds    10
0010.9400.001f   7/1          7/2             5 minutes 30 seconds    10
0010.9400.0024   7/1          7/2             5 minutes 30 seconds    10
0010.9400.001e   7/1          7/2             5 minutes 30 seconds    10
0010.9400.0023   7/1          7/2             5 minutes 30 seconds    10
0010.9400.001d   7/1          7/2             5 minutes 30 seconds    10
0010.9400.001c   7/1          7/2             5 minutes 30 seconds    10
(output truncated)
```

Syntax: show notification mac-movement threshold-rate

[Table 101](#) defines the fields in the output of the **show notification mac-movement threshold-rate** command.

TABLE 101 Field definitions for the show notification mac-movement threshold-rate command

Field	Description
Threshold-Rate Mac Movement Notification is	Specifies whether the MAC movement notification threshold rate is enabled.
Configured Threshold-Rate	The rate in MAC address moves per sampling interval after which a notification is issued. The range is from 1 through 50000.
Configured Sampling-Interval	The sampling interval in seconds over which the number of MAC address moves is measured. The range is from 1 through 86400, which is the number of seconds in a day.
Number of entries in the notification table	One entry for each time a MAC address notification threshold was reached.
MAC-Address	The MAC address that has moved to a different port.
from-Port	The port from which the MAC address moved.
to-Port	The port to which the MAC address moved.
Last Move-Time	The time since the last move occurred.
Vlan-id	The VLAN for the port where the MAC address movement was detected.

Configuring an interval for collecting MAC address move notifications

To configure an interval for collecting statistical data about MAC address moves, enter the **mac-movement notification interval-history** command at the privileged EXEC level. This command enables a corresponding SNMP trap. This history includes statistical information such as the number of MAC addresses that move over the specified period, the total number of MAC address moves, which MAC addresses have moved, and how many times a MAC address has moved.

The software places an upper limit on the number of MAC addresses for which MAC address-specific data is reported. This limit is necessary to do this because it is not possible to report on all MAC addresses when many move.

The following example configures a history interval of 10 seconds.

```
Brocade(config)# mac-movement notification interval-history 10
```

To disable the feature and the corresponding SNMP trap, enter the **no** version of the command, as shown in the following example.

```
Brocade(config)# no mac-movement notification interval-history 10
```

Syntax: [no] **mac-movement notification interval-history** <interval>

The <interval> variable represents the amount of time in seconds during which the MAC address movement notification data is collected. It has no default value.

Viewing MAC address movement statistics for the interval history

To display the collected history of MAC address movement notification, enter the **show notification mac-movement interval-history** command at the privileged EXEC level. This command displays how the history interval is configured in addition to the MAC address move data itself.

```
Brocade# show notification mac-movement interval-history
Interval-History Mac Movement Notification is ENABLED
Configured Interval : 30 seconds
Number of macs that moved in the interval : 100
Total number of moves in the interval: 98654
```

MAC-Address	from-Port	to-Port	Interval	Move-Count	Last Move-Time	Vlan-id
0010.9400.0052	7/1	7/2		1000	11 minutes 22 seconds	10
0010.9400.0051	7/1	7/2		1002	11 minutes 22 seconds	10
0010.9400.0050	7/1	7/2		1012	11 minutes 22 seconds	10
0010.9400.004f	7/1	7/2		1018	11 minutes 22 seconds	10
0010.9400.004e	7/1	7/2		1012	11 minutes 22 seconds	10

(output truncated)

[Table 102](#) defines the fields in the output of the **show notification mac-movement interval-history** command.

TABLE 102 Field definitions for the **show notification mac-movement interval-history** command

Field	Description
Interval-History Mac Movement Notification is	Specifies whether the interval-history data collection is enabled.
Configured Interval	The interval over which the MAC address movement statistics were collected.
Number of macs that moved in the interval	The number of MAC addresses that moved during the configured interval, regardless of how many times each address moved.
Total number of moves in the interval	The total number of MAC address moves over the configured interval.
MAC-Address	The MAC address that has moved to a different port.
from-Port	The port from which the MAC address moved.
to-Port	The port to which the MAC address moved.
Interval Move-Count	The number of times the MAC address has moved within the interval.
Last Move-Time	The time since the last move occurred.
Vlan-id	The VLAN for the port where the MAC address movement was detected.

Displaying and modifying system parameter default settings

Brocade devices have default table sizes for the system parameters shown in the following display outputs. The table sizes determine the maximum number of entries the tables can hold. You can adjust individual table sizes to accommodate your configuration needs.

The tables you can configure, as well as the default values and valid ranges for each table, differ depending on the Brocade device you are configuring. To display the adjustable tables on your Brocade device, use the **show default values** command. The following shows example outputs.

System default settings configuration considerations

- Changing the table size for a parameter reconfigures the device memory. Whenever you reconfigure the memory on a Brocade device, you must save the change to the startup-config file, then reload the software to place the change into effect.
- Configurable tables and their defaults and maximum values differ on Brocade IPv4 devices versus IPv6-capable devices.
- For more information about Layer 3 system parameter limits, refer to [“Displaying Layer 3 system parameter limits”](#) on page 1191.

Displaying system parameter default values

To display the configurable tables and their defaults and maximum values, enter the **show default values** command at any level of the CLI.

The following shows an example output of the **show default values** command on a FastIron Layer 2 device.

```
Brocade#show default values
sys log buffers:50          mac age time:300 sec          telnet sessions:5

System Parameters   Default   Maximum   Current
igmp-max-group-addr 4096      8192      1024
ip-filter-sys       2048      4096      4096
l3-vlan             32        1024      1024
mac                 32768     32768     32768
vlan                64        4095      4095
spanning-tree       32        255       255
mac-filter-port     32        256       256
mac-filter-sys      64        512       512
view                10        65535     65535
rmon-entries        1024      32768     32768
mld-max-group-addr  8192      32768     32768
igmp-snoop-mcache   512       8192      8192
mld-snoop-mcache    512       8192      8192
```

Displaying and modifying system parameter default settings

The following shows an example output on a FastIron IPV4 device running Layer 3 software.

```
Brocade#show default values
sys log buffers:50          mac age time:300 sec          telnet sessions:5

ip arp age:10 min          bootp relay max hops:4      ip ttl:64 hops
ip addr per intf:24

when multicast enabled :
igmp group memb.:260 sec    igmp query:125 sec          hardware drop: enabled

when ospf enabled :
ospf dead:40 sec           ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec

when bgp enabled :
bgp local pref.:100        bgp keep alive:60 sec      bgp hold:180 sec
bgp metric:10             bgp local as:1            bgp cluster id:0
bgp ext. distance:20       bgp int. distance:200     bgp local distance:200
```

System Parameters	Default	Maximum	Current
ip-arp	6000	64000	6000
ip-static-arp	512	6000	512
multicast-route	64	8192	64
dvmrp-route	2048	32000	2048
dvmrp-mcache	512	4096	512
pim-mcache	1024	4096	1024
igmp-max-group-addr	4096	8192	4096
ip-cache	10000	32768	10000
ip-filter-port	1015	1015	1015
ip-filter-sys	2048	8192	2048
l3-vlan	32	1024	32
ip-qos-session	1024	16000	1024
mac	16384	32768	16384
ip-route	80000	262144	80000
ip-static-route	64	2048	64
vlan	64	4095	64
spanning-tree	32	255	32
mac-filter-port	16	256	16
mac-filter-sys	32	512	32
ip-subnet-port	24	128	24
session-limit	65536	160000	65536
view	10	65535	10
virtual-interface	255	512	255
hw-ip-next-hop	2048	6144	2048
hw-logical-interface	4096	4096	4096
hw-ip-mcast-mll	1024	4096	1024
hw-traffic-condition	50	1024	50
rmon-entries	2048	32768	2048
mld-max-group-addr	8192	32768	8192
igmp-snoop-mcache	512	8192	512
mld-snoop-mcache	512	8192	512
msdp-sa-cache	4096	8192	4096

Displaying and modifying system parameter default settings

The following shows an example output on a FCX serving as a management host in an IPv6 network and running the Layer 3 software image.

```
Brocade#show default values
sys log buffers:50          mac age time:300 sec          telnet sessions:5

ip arp age:10 min          bootp relay max hops:4      ip ttl:64 hops
ip addr per intf:24

when multicast enabled :
igmp group memb.:260 sec   igmp query:125 sec          hardware drop: enabled

when ospf enabled :
ospf dead:40 sec           ospf hello:10 sec           ospf retrans:5 sec
ospf transit delay:1 sec

when bgp enabled :
bgp local pref.:100        bgp keep alive:60 sec       bgp hold:180 sec
bgp metric:10              bgp local as:1              bgp cluster id:0
bgp ext. distance:20       bgp int. distance:200      bgp local distance:200
```

System Parameters	Default	Maximum	Current
ip-arp	4000	64000	64000
ip-static-arp	512	6000	6000
multicast-route	64	8192	8192
pim-mcache	1024	4096	4096
igmp-max-group-addr	4096	8192	8192
ip-cache	10000	32768	32768
ip-filter-port	4093	4093	4093
ip-filter-sys	2048	4096	4096
l3-vlan	32	1024	1024
ip-qos-session	1024	16000	16000
mac	32768	32768	32768
ip-route	12000	16100	16100
ip-static-route	64	2048	2048
vlan	64	4095	4095
spanning-tree	32	255	255
mac-filter-port	16	256	256
mac-filter-sys	32	512	512
ip-subnet-port	24	128	128
session-limit	8192	16384	16384
view	10	65535	65535
virtual-interface	255	512	512
rmon-entries	1024	32768	32768
mld-max-group-addr	8192	32768	32768
igmp-snoop-mcache	512	8192	8192
mld-snoop-mcache	512	8192	8192
hw-ip-route-tcam	16384	16384	16384

Displaying and modifying system parameter default settings

The following shows an example output on a FastIron X Series IPV6 device running the Layer 3 software image.

```
Brocade#show default values
sys log buffers:50          mac age time:300 sec          telnet sessions:5

ip arp age:10 min          bootp relay max hops:4      ip ttl:64 hops
ip addr per intf:24

when multicast enabled :
igmp group memb.:260 sec   igmp query:125 sec          hardware drop: enabled

when ospf enabled :
ospf dead:40 sec           ospf hello:10 sec           ospf retrans:5 sec
ospf transit delay:1 sec

when bgp enabled :
bgp local pref.:100        bgp keep alive:60 sec       bgp hold:180 sec
bgp metric:10              bgp local as:1              bgp cluster id:0
bgp ext. distance:20       bgp int. distance:200      bgp local distance:200
```

System Parameters	Default	Maximum	Current
ip-arp	6000	64000	64000
ip-static-arp	512	6000	6000
multicast-route	64	8192	8192
dvmrp-route	2048	32000	32000
dvmrp-mcache	512	4096	4096
pim-mcache	1024	4096	4096
igmp-max-group-addr	4096	8192	8192
ip-cache	10000	32768	32768
ip-filter-port	1015	1015	1015
ip-filter-sys	2048	8192	4096
l3-vlan	32	1024	1024
ip-qos-session	1024	16000	16000
mac	16384	32768	32768
ip-route	262144	524288	524288
ip-static-route	64	2048	2048
vlan	64	4095	4095
spanning-tree	32	255	255
mac-filter-port	16	256	256
mac-filter-sys	32	512	512
ip-subnet-port	24	128	128
session-limit	65536	160000	65537
view	10	65535	65535
virtual-interface	255	512	512
hw-ip-next-hop	2048	6144	2481
hw-traffic-condition	50	1024	52
rmon-entries	2048	32768	32768
mld-max-group-addr	8192	32768	32768
igmp-snoop-mcache	512	8192	8192
mld-snoop-mcache	512	8192	8192
ip6-route	32768	65536	65536
ip6-static-route	64	512	512
ip6-cache	65536	131072	131072
msdp-sa-cache	4096	8192	8192
gre-tunnels	16	64	64

Table 103 defines the system parameters in the **show default values** command output.

TABLE 103 System parameters in show default values command

Parameter	Definition
dvmrp-mcache	PIM and DVMRP multicast cache flows stored in CAM
dvmrp-route	DVMRP routes
hw-ip-mcast-mll	Multicast output interfaces (clients)
hw-ip-next-hop	IP next hops and routes, including unicast next hops and multicast route entries
hw-logical-interface	Hardware logical interface pairs (physical port and VLAN pairs)
hw-traffic-conditioner	Traffic policies
ip-arp	ARP entries
ip-cache	IP forwarding cache entries
ip-filter-port	IP ACL entries per port
ip-filter-sys	IP ACL entries per system
ip-qos-session	Layer 4 session table entries
ip-route	Learned IP routes
ip-static-arp	Static IP ARP entries
ip-static-route	Static IP routes
ip-subnet-port	IP subnets per port
l3-vlan	Layer 3 VLANs
mac	MAC entries
mac-filter-port	MAC address filter entries per port
mac-filter-sys	MAC address filter entries per system
multicast-route	Multicast routes
pim-mcache	PIM multicast cache entries
rmon-entries	RMON control table entries
session-limit	Session entries
spanning-tree	Spanning tree instances
view	SNMP views
virtual-interface	Virtual routing interfaces
vlan	VLANs
mld-max-group-addr	MLD group limit
igmp-snoop-mcache	IGMP snooping cache entries
mld-snoop-mcache	MLD snooping cache entries

Modifying system parameter default values

Information for the configurable tables appears under the columns that are shown in bold type in the above examples. To simplify configuration, the command parameter you enter to configure the table is used for the table name. For example, to increase the capacity of the IP route table, enter the following commands.

```
Brocade(config)#system-max ip-route 120000
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
```

Syntax: **system-max ip-route** *<num>*

The *<num>* parameter specifies the maximum number of routes in the IP route table. The minimum value is 4096. The maximum value is 524288 (subject to route patterns for FSX). The default is 80000 IP routes.

NOTE

If you accidentally enter a value that is not within the valid range of values, the CLI will display the valid range for you.

To increase the number of IP subnet interfaces you can configure on each port on a device running Layer 3 code from 24 to 64, enter the following commands.

```
Brocade(config)#system-max ip-subnet-port 64
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
```

Syntax: **system-max ip-subnet-port** *<num>*

The *<num>* parameter specifies the maximum number of subnet addresses per port and can be from 24 – 128. The default is 24.

Dynamic buffer allocation for QoS priorities for FastIron X Series devices

NOTE

This section applies to FastIron X Series devices only. To configure dynamic buffer limits on FastIron WS, Brocade FCX Series, and ICX devices, refer to [“Dynamic buffer allocation for FCX, FWS, and ICX devices”](#) on page 589.

By default, Brocade IronWare software allocates a certain number of buffers to the outbound transmit queue for each port, based on QoS priority (traffic class). The buffers control the total number of packets permitted in the outbound transmit queue for the port. For each port, the Brocade device defines the maximum outbound transmit buffers, also called *queue depth limits*, as follows:

- **Total Transmit Queue Depth Limit** – The total maximum number of transmit buffers allocated for all outbound packets on a port. Packets are added to the port's outbound queue as long as the number of buffers currently in use is less than the total transmit queue depth limit. When this limit is reached, any new packets attempting to enter the port's transmit queue will be dropped until at least one buffer is freed.
- **Transmit Queue Depth Limit for a Given Traffic Class** – The maximum number of transmit buffers allocated for packets with a given traffic class (0 through 7) on a port. Packets with the specified traffic class are added to the port's outbound queue as long as the number of buffers currently in use for that traffic class is less than the transmit queue depth limit for the traffic class. When this limit is reached, any new packets with the specified traffic class attempting to enter the port's transmit queue will be dropped.

Except for the third-generation Interface modules, you can increase or decrease both of these queue depth limits per port. On the SX-FI48GPP, SX-FI-24GPP, SX-FI-24HF, SX-FI-2XG or SX-FI-8XG modules, to increase or decrease the queue depth limits for a port, you configure a buffer profile and then apply it to the port. A buffer profile can be tied to one or more ports. Therefore, if you change the configuration on one port, it will change the configuration on all ports associated with the same buffer profile.

Dynamic buffer allocation for QoS is useful in situations where applications have intermittent bursts of oversubscription. For example, by increasing the buffers on the egress port, the Brocade device will be able to forward oversubscribed packets instead of dropping them.

Default queue depth limits for FastIron X Series devices

[Table 104](#) defines the default maximum queue depth values per port, per traffic class. The Brocade device drops the packets that cause the port to exceed these limits.

NOTE

The SX-FI48GPP Interface module supports 48 tri-speed (10/100/1000) ports. When the ports are configured at lower speeds, for example, 100 Mbps or 10 Mbps, the maximum queue depth is less than 256 when egress congestion occurs at the front-end Network Processor (NP) of the SX-FI48GPP module. If egress congestion occurs at the back-end NP of the SX-FI48GPP module, the maximum queue depth is 4095. The limit for buffer sharing is 4088.

TABLE 104 Default maximum queue depth

Port type	Maximum queue depth per port, per priority 0 – 6	Maximum queue depth per port, per priority 7	Total maximum queue depth
1 Gbps port	96	224	896
10 Gbps port without jumbo enabled	400	704	3504
10 Gbps port with jumbo enabled	352	640	3104

Configuring the total transmit queue depth limit for FastIron X Series devices

NOTE

To configure the total transmit queue depth limit on a third-generation module, refer to [“Buffer profile configuration”](#) on page 587.

To set the total transmit queue depth limit on a port, enter a command such as the following.

```
Brocade(config)#qd 2 2049
```

This command sets the queue depth limit on port 2 to 2049. Packets are added to the port's outbound queue as long as the packets do not cause the port to exceed 2048 buffers. If the port reaches its queue depth limit of 2049, any new packets attempting to enter the port transmit queue will be dropped until at least one buffer is freed.

Syntax: `qd <port> <limit>`

Specify the *port* variable in one of the following formats:

- FSX, FSX 800, and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The *<limit>* variable can be a value from 0 through 4095. [Table 104](#) on page 585 lists the default values.

Configuring the transmit queue depth limit for a given traffic class on FastIron X Series devices

NOTE

To configure transmit queue depth limits for an SX-FI48GPP module, refer to “[Buffer profile configuration](#)” on page 587.

To set the transmit queue depth limit on a port for a given traffic class, first enter the transmit queue depth limit for the traffic class, and then specify the traffic class.

```
Brocade(config)#qd 2 200 7
```

This command sets the queue depth limit on port 2 to 200 for packets with a traffic class of 7. Packets with priority 7 are added to the outbound queue on port 2 as long as the packets do not exceed 199 buffers. When the port reaches its queue depth limit of 200, packets with the given traffic class will be dropped.

Syntax: `qd <port> <limit> <traffic-class>`

Specify the *port* variable in one of the following formats:

- FSX, FSX 800, and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The *<limit>* variable can be a value from 0 through 4095 and cannot exceed the total transmit queue depth limit configured for the port. [Table 104](#) on page 585 lists the default values.

The sum of the queue depth limits for individual traffic classes on a port does not need to equal the total queue depth limit for the port:

- If the sum of the individual traffic class queue depth limits exceeds the total port limit and the total port limit is reached, any buffer that gets released can be used by any traffic class queue that has not reached its individual limit.
- If the sum of the individual traffic class queue depth limits is less than the total port limit, the remaining buffers can be used only by packets with a priority of 7.

The *<traffic-class>* variable can be a value from 0 through 7, where 7 is the highest priority queue.

Removing buffer allocation limits on FastIron X Series devices

You can remove buffer allocation limits on all ports and all Traffic Classes globally. This permits all available buffers in a port region to be used in a first-come-first-serve basis by any of its ports, regardless of priority. This can be done using the following command.

```
Brocade(config)#buffer-sharing-full
```

Syntax: [no] **buffer-sharing-full**

The **buffer-sharing-full** command sets the total transmit queue depth limit and the transmit queue depth limits for each Traffic Class to 4095 for all ports of the device. The command overrides any existing individually configured queue depth limits.

ATTENTION

The **buffer-sharing-full** command should be used carefully. By entering this command, there is no limit on the number of buffers a port or a specific priority on a port can use. One port could potentially use up all the available buffers of its port region and cause starvation on other ports of the port region.

Buffer profile configuration

The following Interface modules support up to eight buffer profiles:

- SX-FI48GPP
- SX-FI-24GPP
- SX-FI-24HF
- SX-FI-2XG
- SX-FI-8XG

A buffer profile defines the total transmit queue depth limit for a port and the transmit queue depth limit for a given traffic class. On the listed supported Interface modules, each port is associated with a buffer profile. In contrast, Interface modules other than the SX-FI-48GPP, SX-FI-24GPP, SX-FI-24HF, SX-FI-2XG, and the SX-FI-8XG support the configuration of transmit queue depth limits per port.

By default, each port on an SX-FI48GPP, SX-FI-24GPP, SX-FI-24HF, SX-FI-2XG or SX-FI-8XG Interface module is associated with buffer profile ID 1. Profile 0 is reserved for buffer sharing. Default queue depth limits are provided in [“Default queue depth limits for FastIron X Series devices”](#) on page 585. To change the queue depth limit for a port on the supported Interface modules, configure a buffer profile ID that is different from the default or configured value, and apply the buffer profile to the port. When you change the queue depth limit on a port, the configuration will apply to all ports associated with the same buffer profile ID. For example, if ports 1/1 through 1/5 are associated with buffer profile ID 3 and you enter the **qd 1/1 1000** command, the queue depth limit of 1000 will apply to ports 1/1 through 1/5.

NOTE

In the case of SX-FI48GPP module, the queue depth configured using the **qd** command is not applicable when the port is set to speeds lower than 1Gbps. The **qd** command is applicable only when the port is set to speeds greater than 1Gbps.

Configuring a buffer profile and defining the queue depth limits

1. Create a buffer profile and assign it to a port. For example, to create buffer profile 2 and assign it to port 1/1, enter the following command.

```
Brocade(config)#qd 1/1 profile-id 2
```

Syntax: [no] **qd** <slotnum>/<portnum> **profile-id** <number>

The <number> variable specifies the buffer profile ID associated with the <slotnum>/<portnum>. The buffer profile ID can be a number from 1 through 7. Profile 0 is reserved for buffer sharing.

2. Configure the total transmit queue depth limit associated with the buffer profile. For example, to change the total transmit queue depth limit for buffer profile 2 to 1000, enter the following command.

```
Brocade(config)#qd 1/1 1000
```

Because port 1/1 is associated with buffer profile 2, this command sets the queue depth limit for buffer profile 2 to 1000. Packets are added to the outbound queue of a port as long as the packets do not cause the port to exceed 1000 buffers. If the port reaches its queue depth limit of 1000, any new packets attempting to enter the transmit queue will be dropped until at least one buffer is freed.

Syntax: [no] **qd** <slotnum>/<portnum> <limit>

The <limit> variable can be a value from 0 through 4095. The default is 4095.

3. If desired, configure the queue depth limit for a given traffic class. For example, to change the queue depth limit for buffer profile 2 to 300 for packets with a traffic class of 1, enter the following command.

```
Brocade(config)#qd 1/1 300 1
```

For ports that use buffer profile 2, packets with priority 1 are added to the outbound queue as long as the packets do not exceed 299 buffers. When the port reaches its queue depth limit of 300, packets with the given traffic class will be dropped.

Syntax: [no] **qd** <slotnum>/<portnum> <limit> <traffic-class>

The <limit> variable can be a value from 0 through 4095 and cannot exceed the total transmit queue depth limit configured in the previous step.

The sum of the queue depth limits for individual traffic classes on a port does not need to equal the total queue depth limit for the port:

- If the sum of the individual traffic class queue depth limits exceeds the total port limit and the total port limit is reached, any buffer that gets released can be used by any traffic class queue that has not reached its individual limit.
- If the sum of the individual traffic class queue depth limits is less than the total port limit, the remaining buffers can be used only by packets with a priority of 7.

The <traffic-class> variable can be a value from 0 through 7, where 7 is the highest priority queue.

4. Enter the **write memory** command to save the configuration.

Displaying the buffer profile configuration

To display the buffer profile configuration for an SX-FI48GPP Interface module, use the **show configuration** command.

The following example shows that buffer profile 2 and its configured queue depth values apply to ports 1/1 and 1/2. Although the profile configuration was changed for port 1/1 only, port 1/2 has also changed to match the configuration.

```
Brocade(config)#show configuration

qd 1/1 profile-id 2
qd 1/2 profile-id 2
!
qd 1/1 1000
qd 1/2 1000
qd 1/1 300 1
qd 1/2 300 1
```

Syntax: show configuration

Default queue depth limits for FastIron X Series devices

[Table 104](#) defines the default maximum queue depth values per port, per traffic class. The Craig Brocade device drops the packets that cause the port to exceed these limits.

NOTE

The SX-FI48GPP Interface module supports 48 tri-speed (10/100/1000) ports. When the ports are configured at lower speeds, for example 100 Mbps or 10 Mbps, the maximum queue depth is less than 256 when egress congestion occurs at the front-end Network Processor (NP) of the SX-FI48GPP module. If egress congestion occurs at the back-end NP of the SX-FI48GPP module, the maximum queue depth is 4095. The limit for buffer sharing is 4088.

Dynamic buffer allocation for FCX, FWS, and ICX devices

By default, the IronStack architecture allocates fixed buffers on a per-priority queue, per-packet processor basis. The buffers control the total number of packets that can be queued in the outbound transmit for the port. In instances of heavy traffic bursts to aggregation links, such as in stacking configurations or mixed-speed environments, momentary oversubscription of the buffers and descriptors may occur. A descriptor points to one or more packet buffers.

Brocade FastIron stackable devices provide the capability to allocate additional egress buffering and descriptors to handle momentary bursty traffic periods, especially when other priority queues may not be in use, or may not be experiencing heavy traffic. This allows users to allocate and fine-tune the depth of priority buffer queues for each packet processor.

Configuring buffer profiles

There are two different methods of allocating buffers and descriptors to the ports and its queues.

One method uses the **qd-descriptor** and **qd-buffer** CLI commands to allocate descriptors and buffers, respectively, to the port and its queues. This method is available on FWS, FCX, and ICX devices.

The other method uses user-configurable buffer profiles. This method allows you to define a template of buffer allocations to be used on a per-port per-queue basis on the devices. When applied, this buffer profile acts as if you created a series of the **qd** commands. This buffer profile is a simpler form of allocating descriptors and buffers to the port and its queues. This method is available on FCX and ICX devices.

Configuring buffer profiles with qd-descriptor and qd-buffer commands on FCX, FWS, and ICX devices

The 48-port Brocade stackable switch has two packet processors. The 24-port Brocade stackable switch has a single packet processor. For devices in an IronStack, each stack unit has the possibility of two packet processors, but the second processor for a 24-port stack unit cannot be configured. The number of actual available packet processors depends on the type and number of switches in the stack.

For example, for an 8-unit stack of 48 ports, the packet processor numbering scheme is as follows:

- Stack unit 1 - Packet processors 0 and 1
- Stack unit 2 - Packet processors 2 and 3
- Stack unit 3 - Packet processors 4 and 5
- Stack unit 4 - Packet processors 6 and 7
- Stack unit 5 - Packet processors 8 and 9
- Stack unit 6 - Packet processors 10 and 11
- Stack unit 7 - Packet processors 12 and 13
- Stack unit 8 - Packet processors 14 and 15

In this configuration, if stack unit 3 and stack unit 7 are 24-port devices, the odd-numbered packet processors 5 and 13 cannot be configured, and do not exist, although they are reserved.

Configuration steps for buffer profile with qd-descriptor and qd-buffer commands on FCX, FWS, and ICX

The descriptor and buffer allocation process occurs in four sequential steps using the **qd-buffer** and **qd-descriptor** commands.

NOTE

For FCX devices, when you reset buffer values for the 10 Gbps ports, the buffer values for the rear-panel 10 Gbps and 16 Gbps ports are also reset.

1. Configure the allowable port descriptors. Port descriptors set the limit for the ports. The minimum limit for the port descriptors is 16. The maximum limit of the port descriptors depends on the hardware device. Port descriptor limits of different platforms are listed in the section [“Buffer and descriptor maximum and default allocation values”](#) on page 596. Configure the allowable port descriptors by entering a command similar to the following.

```
Brocade# qd-descriptor 1 2 76
```

Syntax: [no] qd-descriptor <devicenum> <portypeval> <numdescriptors>

The <devicenum> variable refers to the device in the stacking unit. The device number starts from 0.

The <portypeval> variable is 1 for 1 Gbps ports and 2 for 10 Gbps ports.

The <numdescriptors> variable refers to the number of descriptors to allocate.

2. Configure the queue descriptors. The minimum limit for queue descriptors is 16. The system default queue descriptors for different platforms are listed in [“Buffer and descriptor maximum and default allocation values”](#) on page 596. Configure the queue descriptors for the queue by entering a command similar to the following.

```
Brocade# qd-descriptor 1 2 76 2
```

Syntax: [no] qd-descriptor <devicenum> <portypeval> <numdescriptors> <priorityqueue>

The <devicenum> variable refers to the device in the stacking unit. The device number starts from 0.

The <portypeval> variable is 1 for 1 Gbps ports and 2 for 10 Gbps ports.

The <numdescriptors> variable refers to the number of descriptors to allocate.

The *<priorityqueue>* variable refers to the specific queue of the port from 0 through 7.

3. Configure the port buffers. The minimum limit for port buffers is 16. The maximum limit for the port buffer depends on the hardware device. Port buffer limits of different platforms are listed in [“Buffer and descriptor maximum and default allocation values”](#) on page 596. Configure the allowable packet buffers by entering a command similar to the following.

```
Brocade# qd-buffer 1 2 76
```

Syntax: [no] qd-buffer *<devicenum>* *<portypeval>* *<numbuffers>*

The *<devicenum>* variable refers to the device in the stacking unit. The device number starts from 0.

The *<portypeval>* variable is 1 for 1 Gbps ports and 2 for 10 Gbps ports.

The *<numbuffers>* variable refers to the number of buffers to allocate.

4. Configure the queue buffers. The maximum limit of queue buffers depends on the hardware device. Queue buffer limits of different platforms are listed in [“Buffer and descriptor maximum and default allocation values”](#) on page 596. Configure the queue buffers by entering a command similar to the following.

```
Brocade# qd-buffer 1 2 76 2
```

Syntax: [no] qd-buffer *<devicenum>* *<portypeval>* *<numbuffers>* *<priorityqueue>*

The *<devicenum>* variable refers to the device in the stacking unit. The device number starts from 0.

The *<portypeval>* variable is 1 for 1 Gbps ports and 2 for 10 Gbps ports.

The *<numbuffers>* variable refers to the number of buffers to allocate.

The *<priorityqueue>* variable refers to the specific queue of the port from 0 through 7.

NOTE

In FWS devices the global buffer limit is 2016 instead of 2094.

Sample configuration for buffer profile with qd-descriptor and qd-buffer commands on FCX, FWS, and ICX

This sample configuration assumes a four-unit stack with the following topology. Note that there is no packet processor 3 or 7, because stack units 2 and 4 are 24-port devices.

- Stack unit 1, 48 ports - Packet processor numbers 0 and 1
- Stack unit 2, 24 ports - Packet processor number 2
- Stack unit 3, 48 ports - Packet processors 4 and 5
- Stack unit 4, 24 ports - Packet processor number 6

The following commands allocate available buffers to be used by priority 0 queues in the four-unit stack.

```
qd-descriptor 0 1 4095
qd-descriptor 1 1 4095
qd-descriptor 2 1 4095
qd-descriptor 4 1 4095
qd-descriptor 5 1 4095
qd-descriptor 6 1 4095
qd-descriptor 0 2 4095
qd-descriptor 1 2 4095
```

```

qd-descriptor 2 2 4095
qd-descriptor 4 2 4095
qd-descriptor 5 2 4095
qd-descriptor 6 2 4095
qd-descriptor 0 1 4095 0
qd-descriptor 1 1 4095 0
qd-descriptor 2 1 4095 0
qd-descriptor 4 1 4095 0
qd-descriptor 5 1 4095 0
qd-descriptor 6 1 4095 0
qd-descriptor 0 2 4095 0
qd-descriptor 1 2 4095 0
qd-descriptor 2 2 4095 0
qd-descriptor 4 2 4095 0
qd-descriptor 5 2 4095 0
qd-descriptor 6 2 4095 0
qd-buffer 0 1 4095
qd-buffer 1 1 4095
qd-buffer 2 1 4095
qd-buffer 4 1 4095
qd-buffer 5 1 4095
qd-buffer 6 1 4095
qd-buffer 0 2 4095
qd-buffer 1 2 4095
qd-buffer 2 2 4095
qd-buffer 4 2 4095
qd-buffer 5 2 4095
qd-buffer 6 2 4095
qd-buffer 0 1 4095 0
qd-buffer 1 1 4095 0
qd-buffer 2 1 4095 0
qd-buffer 4 1 4095 0
qd-buffer 5 1 4095 0
qd-buffer 6 1 4095 0
qd-buffer 0 2 4095 0
qd-buffer 1 2 4095 0
qd-buffer 2 2 4095 0
qd-buffer 4 2 4095 0
qd-buffer 5 2 4095 0
qd-buffer 6 2 4095 0

```

Configuring user-configurable buffer profiles on FCX and ICX devices

NOTE

The user-configurable buffer profile method is not available on FWS devices.

A buffer profile is a mechanism to arbitrarily allocate the egress buffers and descriptors limits to be applied to a port and its queues. Users can define a limit for a port and its queues by configuring the buffer profiles on the device. Therefore, user-configurable buffer profiles provide a template to allocate egress buffers and descriptors limits to the port and on its queues. This template is then applied to the device.

NOTE

Buffer profiles can be configured for 10 Gbps and 1 Gbps ports, but not for ICX 6610 40 Gbps ports. The 10 Gbps profile will apply to ICX 6430 and 6450 stacking ports, as well as FCX 16 Gbps stacking ports.

Configuring and applying a user-configurable buffer profile is a two-step process.

First, create a user-configurable buffer profile with the **qd-buffer-profile** CLI command. Define a name for the user-configurable buffer profile, assign buffer and descriptor limits at the port level, and then define buffer and descriptor limits per queue of that port.

Second, apply the user-configurable buffer profile you created to the device with the **buffer-profile** CLI command.

Configuring a user-configurable buffer profile

To configure a user-configurable buffer profile, complete the following steps.

1. Create a user-configurable buffer profile. For example, to create a user-configurable buffer profile named `profile1` enter the following command.

```
Brocade(config)# qd-buffer-profile profile1#
```

Syntax: **[no] qd-buffer-profile** *<user-profile-name>*

The *<user-profile-name>* variable specifies the name of the user-configurable buffer profile. The profile name can be up to 64 characters long.

2. Configure the port type for the user-configurable buffer profile.

The buffer profile port type can be configured for 1 Gbps ports, 10 Gbps ports, or for all the ports. The **port-type** option should be in line with the **qd-buffer** legacy command; that is, 1 for 1 Gbps ports, 2 for 10 Gbps ports, 0 for all the ports. The default port type is set to 1 Gbps. To configure a user-configurable profile for 10 Gbps ports, the 10 Gbps port type must be explicitly provided by the **port-type** option. Modifications to buffers and descriptors of a port and its queues take effect dynamically.

When the profile type is configured as all 1 Gbps and 10 Gbps ports, the default buffers and descriptors will be set according to the port type; that is, all 1 Gbps ports use 1 Gbps defaults and 10 Gbps ports use 10 Gbps defaults. If you configure a port and its queue with egress buffer and descriptor limits, then the configured limits are used for both 1 Gbps and 10 Gbps ports.

To configure the port type 1 for the profile named “profile1”, enter the following command at the profile configuration level.

```
Brocade(qd-profile-profile1)# profile-config port-type 1
```

Syntax: **[no] profile-config port-type** 0 | 1 | 2

- 0 – All 1 Gbps and 10 Gbps ports
- 1 – All 1 Gbps ports
- 2 – All 10 Gbps ports

NOTE

The **no** form of the command sets the profile port type to 1 Gbps.

Port type modification resets the profile to its default value. All the port and queue buffers and descriptors will be set to either 1 Gbps or 10 Gbps defaults as per the configuration, which means all the user configurations for the port and its queues will be lost.

NOTE

Port type modifications on an active profile are not allowed.

3. Configure the port buffers. Port buffer sets the maximum buffer limit for the ports. The maximum limit depends on the hardware device. Port buffer limits of different platforms are listed in the section [“Buffer and descriptor maximum and default allocation values”](#) on page 596. To configure the port buffers for the user-configurable buffer profile named “profile1”, enter the following command at the profile configuration level.

```
Brocade(qd-profile-profile1)# profile-config port-buffer 8000
```

Syntax: `profile-config port-buffer <decimal>`

The `<decimal>` variable refers to the number of buffers.

4. Configure the port descriptors. Port descriptors set the maximum descriptor limit for the ports. The maximum limit of port descriptors depends on the hardware device. Port descriptor limits of different platforms are listed in [“Buffer and descriptor maximum and default allocation values”](#) on page 596. To configure the port descriptors for the user-configurable buffer profile named “profile1”, enter the following command at the profile configuration level.

```
Brocade(qd-profile-profile1)# profile-config port-descriptor 8000
```

Syntax: `profile-config port-descriptor <decimal>`

The `<decimal>` variable refers to the number of descriptors.

5. Configure the queue buffers. Queue buffers set the maximum buffer limit for a specified queue. If a queue buffer limit is not set, then the default is used. The system default queue buffers for different platforms are listed in [“Buffer and descriptor maximum and default allocation values”](#) on page 596. To configure the queue buffers for the user-configurable buffer profile named “profile1”, enter the following command at the profile configuration level.

```
Brocade(qd-profile-profile1)# profile-config queue-buffers 2 600
```

Syntax: `profile-config queue-buffers <queue-num> <decimal>`

The `<queue-num>` variable refers to the number of the queue of a port from 0 through 7.

The `<decimal>` variable refers to the number of buffers.

6. Configure the queue descriptors. Queue descriptors set the maximum descriptor limit for the specified queue. If a queue descriptors limit is not set, then the default is used. The system default queue descriptors for different platforms are listed in [“Buffer and descriptor maximum and default allocation values”](#) on page 596. To configure the queue descriptors for the user-configurable buffer profile named “profile1”, enter the following command at the profile configuration level.

```
Brocade(qd-profile-profile1)#Profile-config queue-descriptors 2 600
```

Syntax: `profile-config queue-descriptors <queue-num> <decimal>`

The `<queue-num>` variable refers to the number of the queue of a port from 0 through 7.

The `<decimal>` variable refers to the number of descriptors.

Applying a user-configurable buffer profile on the device

Once configured, a user-configurable buffer profile must be applied on the device. To apply the user-configurable buffer profile named “profile1” to a device, enter the following command.

```
Brocade(config)# buffer-profile port-region 0 qd-buffer-profile profile1
```

Syntax: `[no] buffer-profile port-region <port-region> qd-buffer-profile <user-profile-name>`

The `<port-region>` variable is the device number on which the user-configurable buffer profile is applied.

The `<user-profile-name>` variable is the name of the user-configured profile.

Buffer and descriptor maximum and default allocation values

This section lists the maximum and default buffers and descriptors values of a port and its queues on each hardware platform. The following tables are included:

- [Table 105](#) describes FCX devices.
- [Table 106](#) describes FWS devices.
- [Table 107](#) describes ICX 6610 devices.
- [Table 108](#) describes ICX 6430 devices.
- [Table 109](#) describes ICX 6450 devices.

NOTE

Values in the following tables are for software traffic classes (TCs) or QoS priority (qosp) levels.

TABLE 105 Port buffers and descriptors values on FCX devices

	1 Gbps buffers and descriptors	10 Gbps buffers and descriptors
Port Limit	8096	8096
TC0	512	768
TC1	192	320
TC2	192	320
TC3	192	320
TC4	192	320
TC5	256	384
TC6	256	384
TC7	256	384

TABLE 106 Port buffers and descriptors values on FWS devices

	Descriptors	Buffers
Port Limit	3999	2904
TC0	30	176
TC1	30	176
TC2	30	176
TC3	30	176
TC4	30	176
TC5	30	176
TC6	30	176
TC7	60	176

TABLE 107 Port buffer and descriptors values on ICX 6610 devices

	1 Gbps buffers and descriptors	10 Gbps buffers and descriptors	40 Gbps buffers and descriptors
Port Limit	8096	8096	8096
TC0	128	160	256
TC1	32	48	64
TC2	32	48	64
TC3	32	48	64
TC4	32	48	64
TC5	64	96	144
TC6	64	96	144
TC7	64	96	144

TABLE 108 Port buffer and descriptors values on ICX 6430 devices¹

	1 Gbps buffers	10 Gbps buffers	1Gbps descriptors	10 Gbps descriptors
Port Limit	4032	NA	3854	NA
TC0	182	NA	182	NA
TC1	182	NA	182	NA
TC2	96	NA	96	NA
TC3	96	NA	96	NA
TC4	96	NA	96	NA
TC5	128	NA	128	NA
TC6	128	NA	128	NA
TC7	128	NA	128	NA

1. Values are the same for stand-alone and stacking systems.

TABLE 109 Port buffer and descriptors values on ICX 6450 devices¹

	1 Gbps buffers	10 Gbps buffers	1Gbps descriptors	10 Gbps descriptors
Port Limit	6143	6143	5902	5092
TC0	256	384	256	384
TC1	144	192	144	192
TC2	144	192	144	192
TC3	144	192	144	192
TC4	144	192	144	192
TC5	192	256	192	256
TC6	192	256	192	256
TC7	192	256	192	256

1. Values are the same for stand-alone and stacking systems.

Configuring values for the ICX 6430

Port buffer and descriptor values in [Table 108](#) are default values for software traffic classes. For the ICX 6430, traffic classes are mapped to shared hardware queues (refer to “[Queues for the ICX 6430 switch](#)” on page 1973. Refer to the following are considerations for configuring your own values and buffer profiles.

- User-defined values have precedence over default values. For example, if default values for TC2, TC3, and TC4 are 96 and you configure 80 for TC3, then the hardware TC1 will be 80 (software TC2, TC3, and TC4 are mapped to hardware TC1).
- The higher user-configured value has precedence. For example, if the software TC2 default value is 96 (which is also the value of TC3 and 4), if you configure 100 for TC2 and 120 for TC3, then the hardware TC1 value will be 120 (software TC2, TC3, and TC4 are mapped to hardware TC1).

Displaying the user-configurable buffer profile configuration on FCX and ICX devices

To display the specified user-configurable buffer profile configuration, use the **show qd-buffer-profile <user-profile-name>** command.

To display all the user-configurable buffer profiles configured on the device, use the **show qd-buffer-profile active-profile** command.

To display all the buffer profiles configured on the device, use the **show qd-buffer-profile all** command.

The following example shows that the user-configurable buffer profile “OneGigProfile” is configured for 1 Gbps ports with the number of buffers and descriptors allocated to each queue. Note that buffers and descriptors displayed in this example are not necessarily default values.

```
Brocade(config)# show qd-buffer-profile OneGigProfile
User Buffer Profile: OneGigProfile      Port-type: 1Gig
Total Buffers = 8096      Total Descriptors = 8096
Per Queue details:      Buffers      Descriptors
Traffic Class 0          50          38
Traffic Class 1          50          38
Traffic Class 2          50          38
Traffic Class 3          50          38
Traffic Class 4          50          38
Traffic Class 5          50          38
Traffic Class 6          132         132
Traffic Class 7          20          20
```

Syntax: **show qd-buffer <user-profile-name> | all**

[Table 110](#) defines the fields in the output of the **show qd-buffer all** command.

TABLE 110 Field definitions for the output of show qd-buffer-profile command

Field	Description
User Buffer Profile	The name of the user-configurable buffer profile
Port-type	The type of the port: 1 Gbps or 10 Gbps or All
Total Buffers	The total number of buffers allocated to the port
Total Descriptors	The total number of descriptors allocated to the port

TABLE 110 Field definitions for the output of show qd-buffer-profile command

Field	Description
Per Queue details	The names of the queues
Buffers	The total number of buffers allocated to the queue
Descriptors	The total number of descriptors allocated to the queue

Configuring buffer sharing on FCX and ICX devices

Network congestion can be caused by various reasons such as port shaping, flow control received on the link due to congestion on the peer, or oversubscription of the egress line rate.

To support priority queuing, FCX and ICX devices support a configurable amount of guaranteed buffers and descriptors per (port, queue) pair. For information about guaranteed buffers and descriptors, refer to [“Configuring buffer profiles”](#) on page 590.

In addition, FCX and ICX devices support configurable shared buffer pools, which help absorb traffic bursts without packet loss. For a given (port, queue) pair, if its buffer usage exceeds the guaranteed limit, it will start using buffers in the sharing pool. The shared buffers are apportioned among the 1 Gbps, 10 Gbps, 16 Gbps, and stacking ports.

Buffer sharing is always enabled. You can configure buffer sharing by changing the buffer sharing level. To configure the buffer sharing level, enter a command similar to the following.

```
Brocade(config)# qd-share-level 2
```

Syntax: `qd-share-level <level>`

The `<level>` variable is the buffer sharing level, which is a decimal value. The range of valid values for FCX is from 1 to 8. The range of valid values for an ICX 6610 is from 2 through 8.

For descriptions of the buffer sharing levels for FCX devices, refer to [“FCX buffer sharing levels”](#) on page 600.

For descriptions of the buffer sharing levels for ICX 6610 devices, refer to [“ICX 6610 buffer sharing levels”](#) on page 600.

For descriptions of the buffer sharing levels for ICX 6430 and 6450 devices, refer to [“ICX 6430 and ICX 6450 buffer sharing levels”](#) on page 601.

FCX buffer sharing levels

The FCX buffer sharing level configures the shared buffers on the device. [Table 111](#) defines the FCX buffer sharing level settings. For information about configuring buffer sharing, refer to [“Configuring buffer sharing on FCX and ICX devices”](#) on page 599.

If you configure buffers at the port or queue level (using **qd** commands or buffer profiles), the buffer sharing level automatically changes to 1. You can change it manually.

TABLE 111 FCX buffer sharing level definitions

Buffer sharing level	Shared buffer limit (in buffers)	Shared buffer total (in kilobytes)
1	256	64
2	1024	250
3	1536	375
4	2048	500
5 (default)	2560	625
6	3072	750
7	3584	875
8	4096	1000

ICX 6610 buffer sharing levels

The ICX 6610 buffer sharing level configures the shared buffers on the device. The ICX 6610 shared buffers are divided into pools. Each of the following pools defines the buffer allocation for a set of traffic class (TC) queues:

- Pool 0 contains TCs 0 and 1.
- Pool 1 contains TCs 2, 3, and 4.
- Pool 2 contains TCs 5 and 6.
- Pool 3 contains TC 7.

For example, the buffers allocated to Pool 0 are shared between TCs 0 and 1. [Table 112](#) defines the ICX 6610 buffer sharing level settings. Note that only the values of Pool 0 change. For information about configuring buffer sharing, refer to [“Configuring buffer sharing on FCX and ICX devices”](#) on page 599.

TABLE 112 ICX 6610 buffer sharing level definitions

Buffer sharing level	Shared buffer limit				Shared buffer total (in kilobytes)	Pool 0 sharing buffers (in kilobytes)
	Pool 0 – TC 0, 1	Pool 1 – TC 2, 3, 4	Pool 2 – TC 5, 6	Pool 3 – TC 7		
1	Not supported					
2	128	128	192	192	250	64
3	256	128	192	192	375	125
4	512	128	192	192	500	250

TABLE 112 ICX 6610 buffer sharing level definitions (Continued)

Buffer sharing level	Shared buffer limit				Shared buffer total (in kilobytes)	Pool 0 sharing buffers (in kilobytes)
	Pool 0 – TC 0, 1	Pool 1 – TC 2, 3, 4	Pool 2 – TC 5, 6	Pool 3 – TC 7		
5 (default)	768	128	192	192	625	375
6	1024	128	192	192	750	500
7	1280	128	192	192	875	625
8	1536	128	192	192	1000	750

ICX 6430 and ICX 6450 buffer sharing levels

The ICX 6430 and 6450 buffer sharing level configures the shared buffers on the device.

[Table 111](#) defines the FCX buffer sharing level settings. For information about configuring buffer sharing, refer to “[Configuring buffer sharing on FCX and ICX devices](#)” on page 599.

If you configure buffers at the port or queue level (using **qd** commands or buffer profiles), the buffer sharing level automatically changes to 1. You can change it manually.

TABLE 113 ICX 6430 and ICX 6450 buffer sharing level definitions

Buffer sharing level	Shared buffer limit (in buffers)	Shared buffer total (in kilobytes)
1	256	64
2 (default)	1024	250
3	1536	375
4	2048	500
5	2560	625
6	3072	750
7	3584	875
8	4096	1000

Displaying buffer sharing information

To display information about buffer sharing, enter the **show qd-share-level** command. Following is an example for an FCX device.

```
Brocade# show qd-share-level
Sharing level: 1-64KB, 2-250KB, 3-375KB, 4-500KB, 5-625KB (default), 6-750KB,
7-875KB, 8-1000KB
Current qdsharing level 5
Device 0 Sharing buffers in use 0
Device 1 Sharing buffers in use 0
Device 1 Sharing buffers in use 0
Device 2 Sharing buffers in use 0
```

Following is an example for ICX 6610 devices.

```
ICX6610-48 Router# show qd-share-level
Sharing level: 1-64KB, 2-250KB, 3-375KB, 4-500KB, 5-625KB (default), 6-750KB,
7-875KB, 8-1000KB
Current qd sharing level 5
Sharing pools to Traffic Class (TC) map:
    Pool 0: TC 0,1  Pool 1: TC 2,3,4  Pool 2: TC 5,6  Pool 3: TC 7
Device 0 Sharing pool 0 buffers in use 0
Device 0 Sharing pool 1 buffers in use 0
Device 0 Sharing pool 2 buffers in use 0
Device 0 Sharing pool 3 buffers in use 0
Device 1 Sharing pool 0 buffers in use 0
Device 1 Sharing pool 1 buffers in use 0
Device 1 Sharing pool 2 buffers in use 0
Device 1 Sharing pool 3 buffers in use 0
Device 2 Sharing pool 0 buffers in use 0
Device 2 Sharing pool 1 buffers in use 0
Device 2 Sharing pool 2 buffers in use 0
Device 2 Sharing pool 3 buffers in use 0
```

Syntax: show qd-share-level

The command output displays the following information:

- Definitions of the buffer sharing levels
- The current buffer sharing level
- Mapping of traffic classes to sharing pools (ICX 6610 devices only)
- Buffer usage information

Removing buffer allocation limits on FCX, FWS, and ICX

You can remove buffer allocation limits on all ports and all traffic classes globally. This permits all available buffers in a port region to be used on a first-come, first-served basis by any of its ports, regardless of priority. This can be done using the following command.

```
Brocade(config)# buffer-sharing-full
```

Syntax: [no] buffer-sharing-full

The command overrides any existing configured queue depth limits and buffer allocation.

ATTENTION

Use the **buffer-sharing-full** command carefully. By entering this command, there is no limit to the number of buffers a port or a specific priority on a port can use. One port could potentially use up all the available buffers of its port region and cause starvation on other ports of the port region.

To prevent traffic loss during temporary network bursts, it is recommended that you use guaranteed (port, queue) buffers allocation or shared buffer allocation to adjust queue depth, rather than enabling the **buffer-sharing-full** command.

NOTE

The **buffer-sharing-full** command can create unpredictable behavior during traffic congestion or a blocking scenario, compromising network stability (by losing control packets), QoS, and stacking.

Buffer profiles for VoIP on FastIron stackable devices

NOTE

Configuring buffer profiles for VoIP traffic is not supported on FastIron X Series devices.

Default buffer settings are currently optimized for 1 GbE-to-1 GbE traffic. Configuring VoIP buffer profiles adds buffer profiles for 1 GbE-to-100 Mbit traffic, simplifying configuration and improving performance.

VoIP profiles allows you to configure a pre-defined set of buffers and descriptors for the priority 0 and 7. These profiles support VoIP traffic that uses priority 7, with 10 Mbps or 100 Mbps uplink ports and 1000 Mbps downlink ports.

NOTE

In previous software versions, you could manually configure buffers and descriptors using **qd** commands. Buffer profiles for VoIP cannot coexist with **qd** commands. You may use buffer profiles for VoIP or **qd** command, but not both at the same time.

Configuring buffer profiles for VoIP

To configure predefined buffers, enter a command similar to the following.

```
Brocade#buffer-profile port-region 0 voip downlink 100 uplink 1000
```

Syntax: [no] buffer-profile port-region <num> voip downlink 100 uplink 1000

NOTE

The port-region num can be either 0 (ports 0/1/1 to 0/1/24) or 1 (ports 0/1/25 to 0/1/48).

NOTE

Only FGS and FLS models support the **buffer-profile port-region <num> voip downlink 100 uplink 1000** command. FCX and ICX models do support this command.

Deleting buffer profiles for VoIP

To delete an existing buffer profile configuration, use the no form of the command.

```
Brocade#no buffer-profile port-region 0 voip downlink 100 uplink 1000
```

Syntax: [no] buffer-profile port-region <num> voip downlink 100 uplink 1000

Remote Fault Notification on 1Gbps fiber connections

NOTE

Remote fault notification (RFN) is only available for 1 Gbps Ethernet Fiber ports. It is not available for 10/100 ports and Gbps Ethernet Copper ports. This feature is not available on ICX 6430 and ICX 6450 devices.

For fiber-optic connections, you can optionally configure a transmit port to notify the receive port on the remote device whenever the transmit port becomes disabled.

When you enable this feature, the transmit port notifies the remote port whenever the fiber cable is either physically disconnected or has failed. When this occurs and the feature is enabled, the device disables the link and turns OFF both LEDs associated with the ports.

By default, RFN is enabled.

You can configure RFN as follows:

- Globally, on the entire device
- On a trunk group
- On an individual interface

Enabling and disabling remote fault notification

RFN is ON by default. To disable RFN, use the following command.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#gig-default neg-off
```

To re-enable RFN, use the following command.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#gig-default auto-gig
```

Syntax: `gig-default neg-off | auto-gig`

For more information about the parameters supported with the **gig-default** command, see [“Changing the Gbps fiber negotiation mode”](#) on page 59.

Link Fault Signaling for 10Gbps Ethernet devices

Link Fault Signaling (LFS) is a physical layer protocol that enables communication on a link between two 10 Gbps Ethernet devices. When configured on a Brocade 10 Gbps Ethernet port, the port can detect and report fault conditions on transmit and receive ports. Brocade recommends enabling LFS on both ends of a link.

LFS is disabled by default on all Brocade FastIron devices.

NOTE

Enable LFS on any device prior to connecting that device with FastIron platforms. FastIron platforms have LFS enabled by default and it cannot be disabled; any connecting device must have LFS currently enabled to ensure interoperability.

When LFS is enabled on an interface, the following Syslog messages are generated when the link goes up or down, or when the TX or RX fiber is removed from one or both sides of the link that has LFS enabled.

```
Interface ethernet1/1, state down - link down
Interface ethernet1/1, state up
```

When a link fault occurs, the Link and Activity LEDs turn OFF.

The Link and Activity LEDs turn ON when there is traffic traversing the link after the fiber is installed.

Enabling Link Fault Signaling

To enable Link Fault Signaling (LFS) between two 10 Gbps Ethernet devices, enter commands such as the following on both ends of the link.

```
Brocade(config)#interface e 1/1
Brocade(config-if-e1000-1/1)#link-fault-signal
```

Syntax: [no] link-fault-signal

Use the **no** form of the command to disable LFS.

LFS is OFF by default.

Viewing the status of LFS-enabled links

The status of an LFS-enabled link is shown in the output of the **show interface** and **show interface brief** commands, as shown in the following examples.

```
Brocade#show interface e 10/1
10GigabitEthernet10/1 is down (remote fault), line protocol is down
  Hardware is 10GigabitEthernet, address is 0012.f227.79d8 (bia 0012.f227.79d8)
  Configured speed 10Gbit, actual unknown, configured duplex fdx, actual unknown
  Member of L2 VLAN ID 1, port is untagged, port state is BLOCKING
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Fault Signaling is Enabled, Link Error Dampening is Disabled
  STP configured to ON, priority is level0
  Flow Control is disabled
  mirror disabled, monitor disabled

some lines omitted for brevity...
```

The bold text in the above output shows that the LFS-enabled link (port 10/1) is down because of an error on the remote port, as indicated by **remote fault**.

Syntax: show interface ethernet <port>

Specify the *port* variable in one of the following formats:

- FSX, FSX 800, and FSX 1600 chassis devices – *slotnum/portnum*
- FESX compact switches – *portnum*

```
Brocade#show interfaces brief
Port  Link      State  Dupl  Speed  Trunk  Tag  Pvid  Pri  MAC              Name
10/1  Err-LFS  None   None   None   None   No   1     0   0012.f227.79d8
```

The bold text in the above output indicates that there is an error on the LFS-enabled link on port 10/1 and the link is down.

Syntax: show interfaces brief

Jumbo frame support

Ethernet traffic moves in units called frames. The maximum size of frames is called the Maximum Transmission Unit (MTU). When a network device receives a frame larger than its MTU, the data is either fragmented or dropped. Historically, Ethernet has a maximum frame size of 1500 bytes, so most devices use 1500 as their default MTU.

Jumbo frames are Ethernet frames with more than 1,500 bytes MTU. Conventionally, jumbo frames can carry up to 9,000 bytes MTU. Brocade FastIron devices support Layer 2 jumbo frames on 10/100, 100/100/1000, and 10GbE ports.

Metro Features

Table 114 lists the individual Brocade FastIron switches and the metro features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 114 Supported metro features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Topology groups	Yes	Yes	Yes	Yes	Yes
Metro Ring Protocol 1 (MRP 1)	Yes	Yes	Yes	Yes	Yes
Metro Ring Protocol 2 (MRP 2)	Yes	Yes	Yes	Yes	Yes
Extended MRP ring IDs from 1 – 1023	Yes	No	Yes	Yes	Yes
Virtual Switch Redundancy Protocol (VSRP)	Yes	Yes	Yes	Yes	Yes
VSRP-Aware security features	Yes	Yes	Yes	Yes	Yes
VSRP and MRP signaling	Yes	Yes	Yes	Yes	Yes
VSRP Fast Start	Yes	Yes	Yes	Yes	Yes
VSRP timer scaling	Yes	Yes	Yes	Yes	Yes

Topology groups

A topology group is a named set of VLANs that share a Layer 2 topology. Topology groups simplify configuration and enhance scalability of Layer 2 protocols by allowing you to run a single instance of a Layer 2 protocol on multiple VLANs.

You can use topology groups with the following Layer 2 protocols:

- STP
- MRP
- VSRP
- 802.1W

Topology groups simplify Layer 2 configuration and provide scalability by enabling you to use the same instance of a Layer 2 protocol for multiple VLANs. For example, if a Brocade device is deployed in a Metro network and provides forwarding for two MRP rings that each contain 128 VLANs, you can configure a topology group for each ring. If a link failure in a ring causes a topology change, the change is applied to all the VLANs in the ring topology group. Without topology groups, you would need to configure a separate ring for each VLAN.

Master VLAN and member VLANs

Each topology group contains a master VLAN and can contain one or more member VLANs and VLAN groups:

- **Master VLAN** – The master VLAN contains the configuration information for the Layer 2 protocol. For example, if you plan to use the topology group for MRP, the topology group master VLAN contains the ring configuration information.
- **Member VLANs** – The member VLANs are additional VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the member VLANs. A change to the master VLAN Layer 2 protocol configuration or Layer 2 topology affects all the member VLANs. Member VLANs do not independently run a Layer 2 protocol.
- **Member VLAN groups** – A VLAN group is a named set of VLANs. The VLANs within a VLAN group have the same ports and use the same values for other VLAN parameters.

When a Layer 2 topology change occurs on a port in the master VLAN, the same change is applied to that port in all the member VLANs that contain the port. For example, if you configure a topology group whose master VLAN contains ports 1/1 and 1/2, a Layer 2 state change on port 1/1 applies to port 1/1 in all the member VLANs that contain that port. However, the state change does not affect port 1/1 in VLANs that are not members of the topology group.

Control ports and free ports

A port that is in a topology group can be a control port or a free port:

- **Control port** – A control port is a port in the master VLAN, and is therefore controlled by the Layer 2 protocol configured in the master VLAN. The same port in all the member VLANs is controlled by the master VLAN Layer 2 protocol. Each member VLAN must contain all of the control ports and can contain additional ports.
- **Free port** – A free port is not controlled by the master VLAN Layer 2 protocol. The master VLAN can contain free ports. (In this case, the Layer 2 protocol is disabled on those ports.) In addition, any ports in the member VLANs that are not also in the master VLAN are free ports.

NOTE

Since free ports are not controlled by the master port Layer 2 protocol, they are assumed to always be in the Forwarding state.

Topology group configuration considerations

- You must configure the master VLAN and member VLANs or member VLAN groups before you configure the topology group.
- You can configure up to 256 topology groups. Each group can control up to 4096 VLANs. A VLAN cannot be controlled by more than one topology group.
- The topology group must contain a master VLAN and can also contain individual member VLANs, VLAN groups, or a combination of individual member VLANs and VLAN groups.
- If you add a new master VLAN to a topology group that already has a master VLAN, the new master VLAN replaces the older master VLAN. All member VLANs and VLAN groups follow the Layer 2 protocol settings of the new master VLAN.

- If you remove the master VLAN (by entering **no master-vlan <vlan-id>**), the software selects the new master VLAN from member VLANs. A new candidate master VLAN will be in configured order to a member VLAN so that the first added member VLAN will be a new candidate master VLAN. Once you save and reload, a member-vlan with the youngest VLAN ID will be the new candidate master. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.
- Once you add a VLAN as a member of a topology group, all the Layer 2 protocol information on the VLAN is deleted.

Configuring a topology group

To configure a topology group, enter commands such as the following.

```
Brocade(config)#topology-group 2
Brocade(config-topo-group-2)#master-vlan 2
Brocade(config-topo-group-2)#member-vlan 3
Brocade(config-topo-group-2)#member-vlan 4
Brocade(config-topo-group-2)#member-vlan 5
Brocade(config-topo-group-2)#member-group 2
```

These commands create topology group 2 and add the following:

- Master VLAN 2
- Member VLANs 2, 3, and 4
- Member VLAN group 2

Syntax: **[no] topology-group <group-id>**

The **<group-id>** parameter specifies the topology group ID and can be from 1 – 256.

Syntax: **[no] master-vlan <vlan-id>**

This command adds the master VLAN. The VLAN must already be configured. Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

NOTE

If you remove the master VLAN (by entering **no master-vlan <vlan-id>**), the software selects the new master VLAN from member VLANs. For example, if you remove master VLAN 2 from the example above, the CLI converts member VLAN 3 into the new master VLAN. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.

NOTE

If you add a new master VLAN to a topology group that already has a master VLAN, the new master VLAN replaces the older master VLAN. All member VLANs and VLAN groups follow the Layer 2 protocol settings of the new master VLAN.

Syntax: **[no] member-vlan <vlan-id>**

The **<vlan-id>** parameter specifies a VLAN ID. The VLAN must already be configured.

Syntax: **[no] member-group <num>**

The **<num>** specifies a VLAN group ID. The VLAN group must already be configured.

NOTE

Once you add a VLAN or VLAN group as a member of a topology group, all the Layer 2 protocol configuration information for the VLAN or group is deleted. For example, if STP is configured on a VLAN and you add the VLAN to a topology group, the STP configuration is removed from the VLAN. Once you add the VLAN to a topology group, the VLAN uses the Layer 2 protocol settings of the master VLAN.

If you remove a member VLAN or VLAN group from a topology group, you will need to reconfigure the Layer 2 protocol information in the VLAN or VLAN group.

Displaying topology group information

The following sections show how to display STP information and topology group information for VLANs.

Displaying STP information

To display STP information for a VLAN, enter a command such as the following.

```
Brocade#show span vlan 4
VLAN 4 BPDU cam_index is 14344 and the Master DMA Are(HEX) 18 1A
STP instance owned by VLAN 2
```

This example shows STP information for VLAN 4. The line shown in bold type indicates that the VLAN STP configuration is controlled by VLAN 2. This information indicates that VLAN 4 is a member of a topology group and VLAN 2 is the master VLAN in that topology group.

Displaying topology group information

To display topology group information, enter the following command.

```
Brocade#show topology-group

Topology Group 3
=====
master-vlan 2
member-vlan none

Common control ports          L2 protocol
ethernet 1/1                  MRP
ethernet 1/2                  MRP
ethernet 1/5                  VSRP
ethernet 2/22                 VSRP
Per vlan free ports
ethernet 2/3                  Vlan 2
ethernet 2/4                  Vlan 2
ethernet 2/11                 Vlan 2
ethernet 2/12                 Vlan 2
```

Syntax: show topology-group [<group-id>]

This display shows the following information.

TABLE 115 CLI display of topology group information

Field	Description
master-vlan	The master VLAN for the topology group. The settings for STP, MRP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
member-vlan	The member VLANs in the topology group.
Common control ports	The master VLAN ports that are configured with Layer 2 protocol information. The Layer 2 protocol configuration and state of these ports in the master VLAN applies to the same port numbers in all the member VLANs.
L2 protocol	The Layer 2 protocol configured on the control ports. The Layer 2 protocol can be one of the following: <ul style="list-style-type: none"> • MRP • STP • VSRP
Per vlan free ports	The ports that are not controlled by the Layer 2 protocol information in the master VLAN.

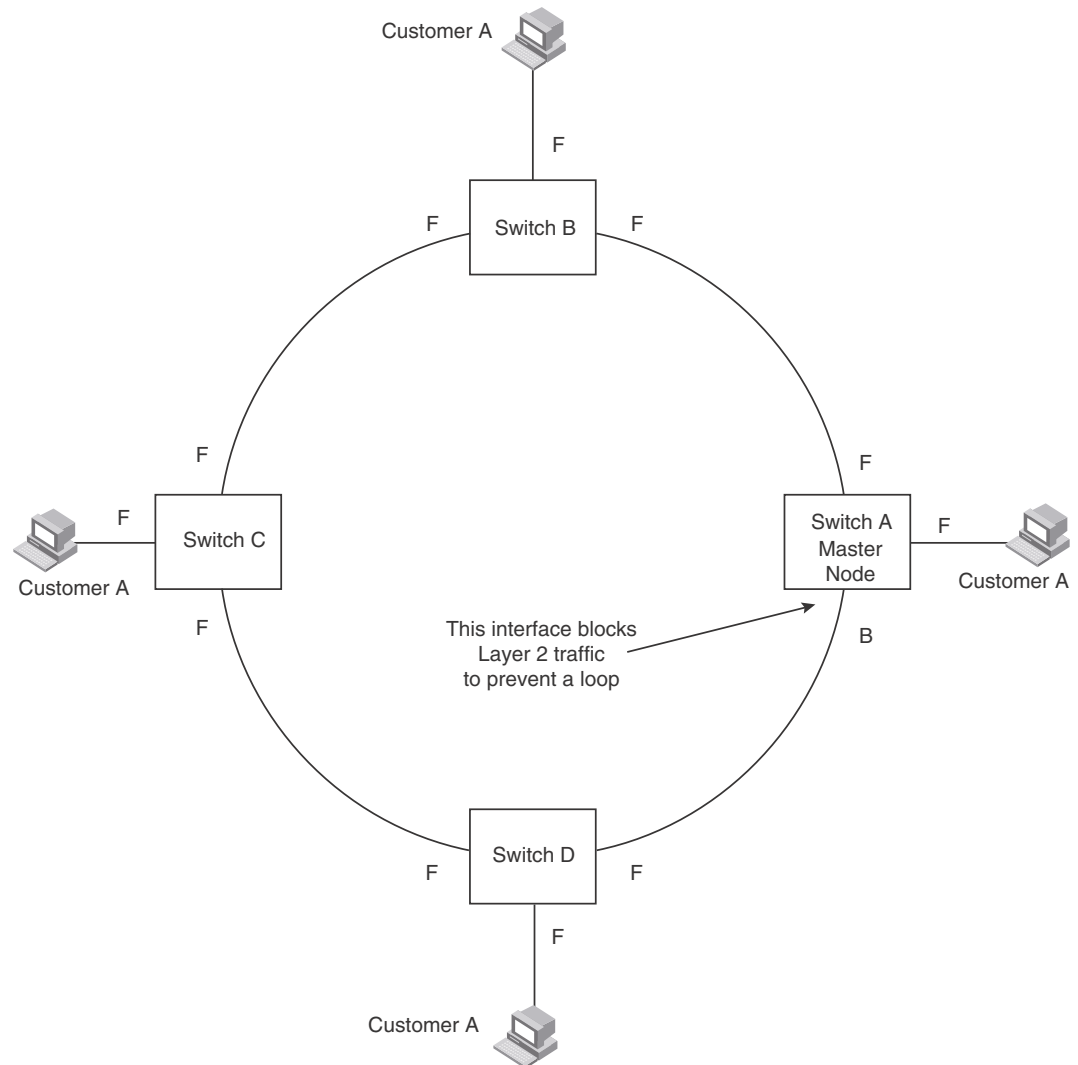
Metro Ring Protocol

Metro Ring Protocol (MRP) is a Brocade proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in Layer 2 ring topologies. It is an alternative to STP and is especially useful in Metropolitan Area Networks (MANs) where using STP has the following drawbacks:

- STP allows a maximum of seven nodes. Metro rings can easily contain more nodes than this.
- STP has a slow reconvergence time, taking many seconds or even minutes. MRP can detect and heal a break in the ring in sub-second time.

Figure 40 shows an example of an MRP metro ring.

FIGURE 40 Metro ring – normal state



The ring in this example consists of four MRP nodes (Brocade switches). Each node has two interfaces with the ring. Each node also is connected to a separate customer network. The nodes forward Layer 2 traffic to and from the customer networks through the ring. The ring interfaces are all in one port-based VLAN. Each customer interface can be in the same VLAN as the ring or in a separate VLAN.

One node is configured as the master node of the MRP ring. One of the two interfaces on the master node is configured as the primary interface; the other is the secondary interface. The primary interface originates Ring Health Packets (RHPs), which are used to monitor the health of the ring. An RHP is forwarded on the ring to the next interface until it reaches the secondary interface of the master node. The secondary interface blocks the packet to prevent a Layer 2 loops.

Metro Ring Protocol configuration notes

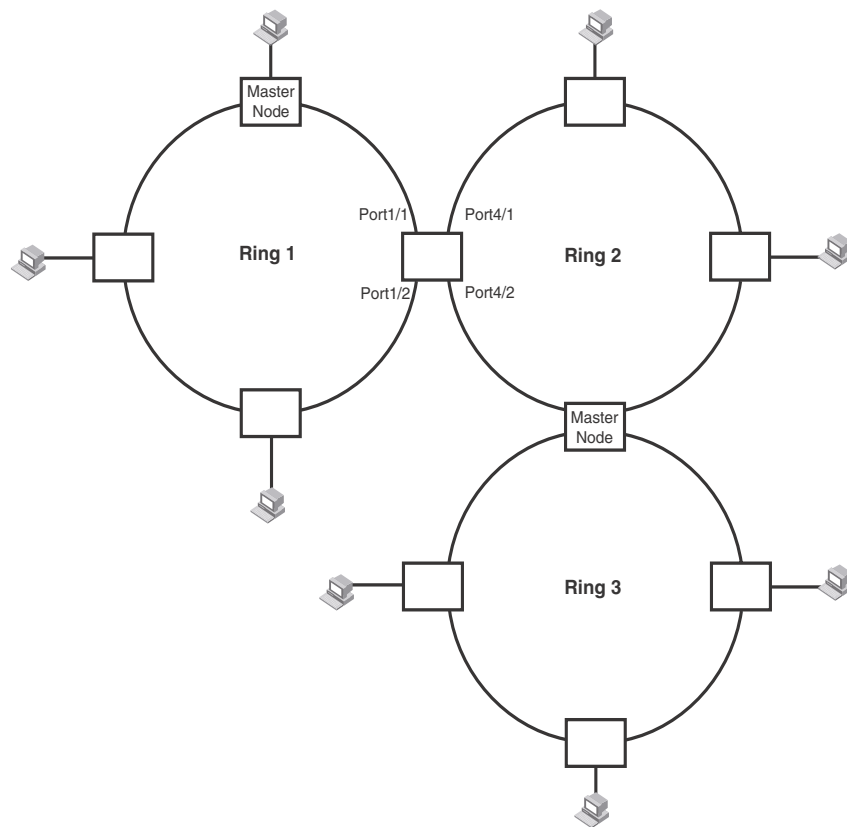
- When you configure Metro Ring Protocol (MRP), Brocade recommends that you disable one of the ring interfaces before beginning the ring configuration. Disabling an interface prevents a Layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once MRP is configured and enabled on all the nodes, you can re-enable the interface.
- The above configurations can be configured as MRP masters or MRP members (for different rings).
- Brocade does not recommend configuring more than 15 MRP instances on FCX devices. Also, due to hardware limitations on this platforms, configuring 40 or more MRP instances may cause errors.
- If you configure MRP on a device running Layer 3 software, then restart the device running Layer 2 software, the MRP configuration gets deleted.

MRP rings without shared interfaces (MRP Phase 1)

MRP Phase 1 allows you to configure multiple MRP rings, as shown in [Figure 41](#), but the rings cannot share the same link. For example, you cannot configure ring 1 and ring 2 to each have interfaces 1/1 and 1/2.

Also, when you configure an MRP ring, any node on the ring can be designated as the master node for the ring. A master node can be the master node of more than one ring. (Refer to [Figure 41](#).) Each ring is an independent ring and RHP packets are processed within each ring.

FIGURE 41 Metro ring – multiple rings



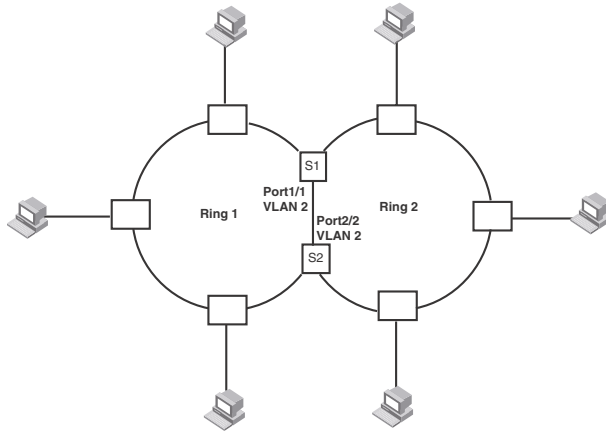
In this example, two nodes are each configured with two MRP rings. Any node in a ring can be the master for its ring. A node also can be the master for more than one ring.

MRP rings with shared interfaces (MRP Phase 2)

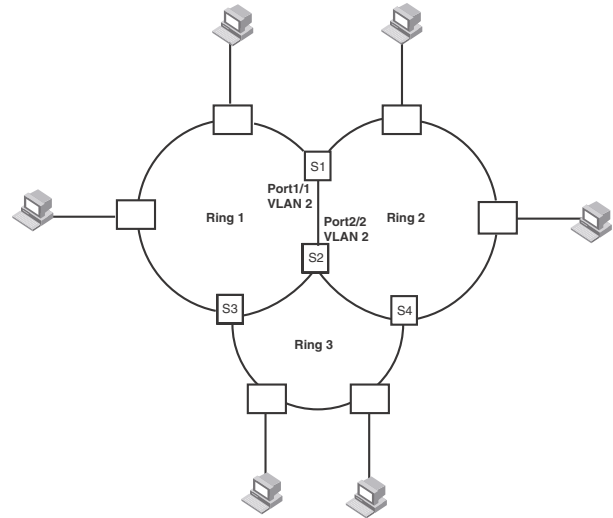
With MRP Phase 2, MRP rings can be configured to share the same interfaces as long as the interfaces belong to the same VLAN. [Figure 42](#) shows examples of multiple MRP rings that share the same interface.

FIGURE 42 Examples of multiple rings sharing the same interface - MRP Phase 2

Example 1



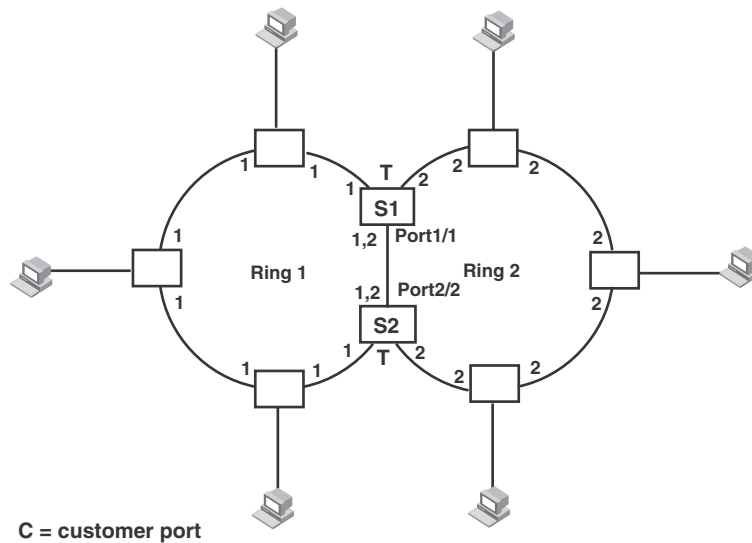
Example 2



On each node that will participate in the ring, you specify the ring ID and the interfaces that will be used for ring traffic. In a multiple ring configuration, a ring ID determines its priority. The lower the ring ID, the higher the priority of a ring.

A ring ID is also used to identify the interfaces that belong to a ring.

FIGURE 43 Interface IDs and types



For example, in [Figure 43](#), the ID of all interfaces on all nodes on Ring 1 is 1 and all interfaces on all nodes on Ring 2 is 2. Port 1/1 on node S1 and Port 2/2 on S2 have the IDs of 1 and 2 since the interfaces are shared by Rings 1 and 2.

The ring ID is also used to determine an interface priority. Generally, a ring ID is also the ring priority and the priority of all interfaces on that ring. However, if the interface is shared by two or more rings, then the highest priority (lowest ID) becomes the priority of the interface. For example, in [Figure 43](#), all interfaces on Ring 1, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 1. Likewise, all interfaces on Ring 2, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 2. Port 1/1 on S1 and Port 2/2 on S2 have a priority of 1 since 1 is the highest priority (lowest ID) of the rings that share the interface.

If a node has interfaces that have different IDs, the interfaces that belong to the ring with the highest priority become regular ports. Those interfaces that do not belong to the ring with the highest priority become tunnel ports. In [Figure 43](#), nodes S1 and S2 have interfaces that belong to Rings 1 and 2. Those interfaces with a priority of 1 are regular ports. The interfaces with a priority of 2 are the tunnel ports since they belong to Ring 2, which has a lower priority than Ring 1.

Selection of master node

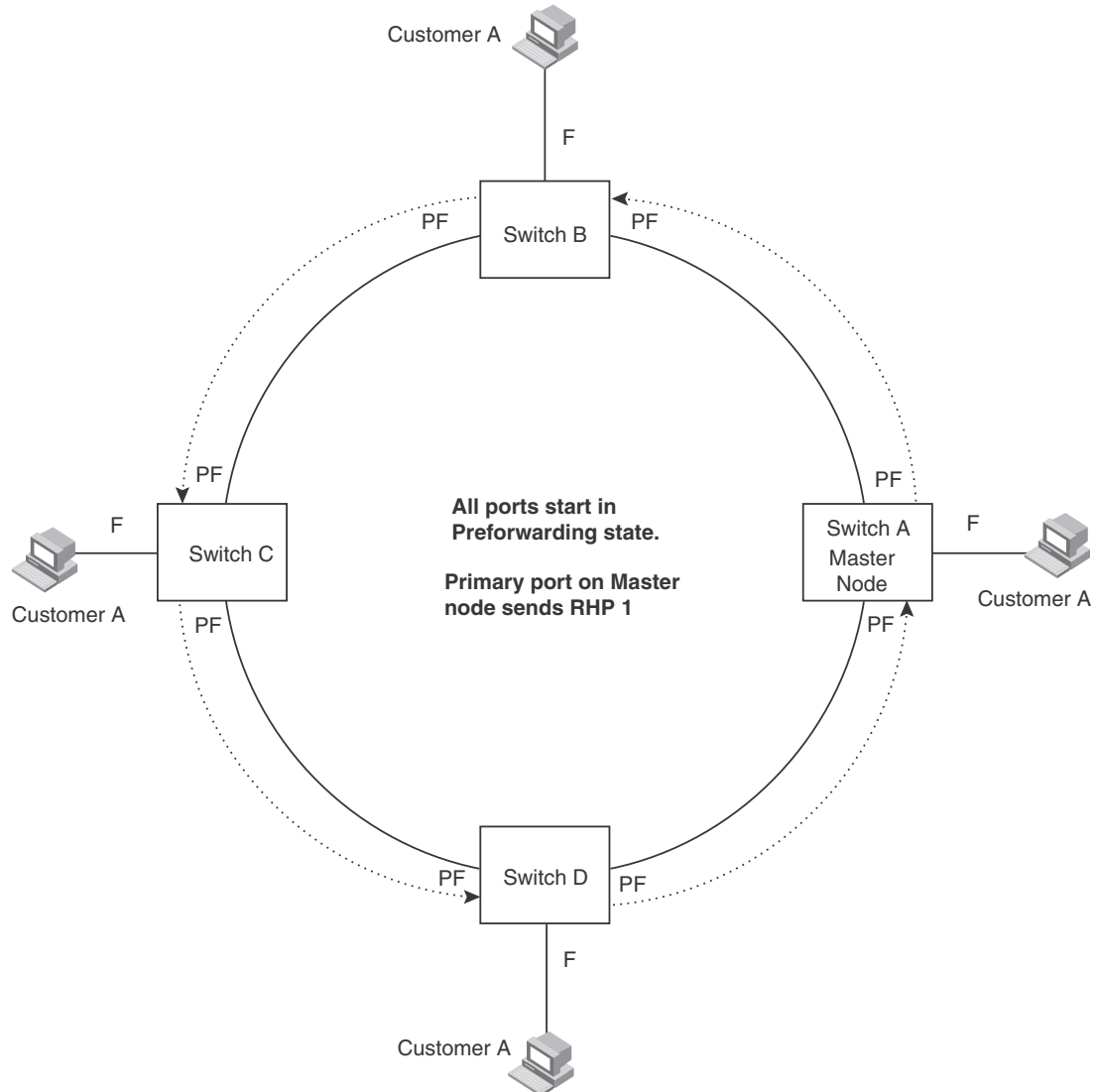
Allowing MRP rings to share interfaces limits the nodes that can be designated as the master node. Any node on an MRP ring that does not have a shared interface can be designated as the ring master node. However, if all nodes on the ring have shared interfaces, nodes that do not have tunnel ports can be designated as the master node of that ring. If none of the nodes meet these criteria, you must change the rings' priorities by reconfiguring the rings' ID.

In [Figure 43](#), any of the nodes on Ring 1, even S1 or S2, can be a master node since none of its interfaces are tunnel ports. However in Ring 2, neither S1 nor S2 can be a master node since these nodes contain tunnel ports.

Ring initialization

The ring shown in [Figure 40](#) shows the port states in a fully initialized ring without any broken links. [Figure 44](#) shows the initial state of the ring, when MRP is first enabled on the ring switches. All ring interfaces on the master node and member nodes begin in the Preforwarding state (PF).

FIGURE 44 Metro ring – initial state



MRP uses Ring Health Packets (RHPs) to monitor the health of the ring. An RHP is an MRP protocol packet. The source address is the MAC address of the master node and the destination MAC address is a protocol address for MRP. The Master node generates RHPs and sends them on the ring. The state of a ring port depends on the RHPs.

RHP processing in MRP Phase 1

A ring interface can have one of the following MRP states:

- **Preforwarding (PF)** – The interface can forward RHPs but cannot forward data. All ring ports begin in this state when you enable MRP.

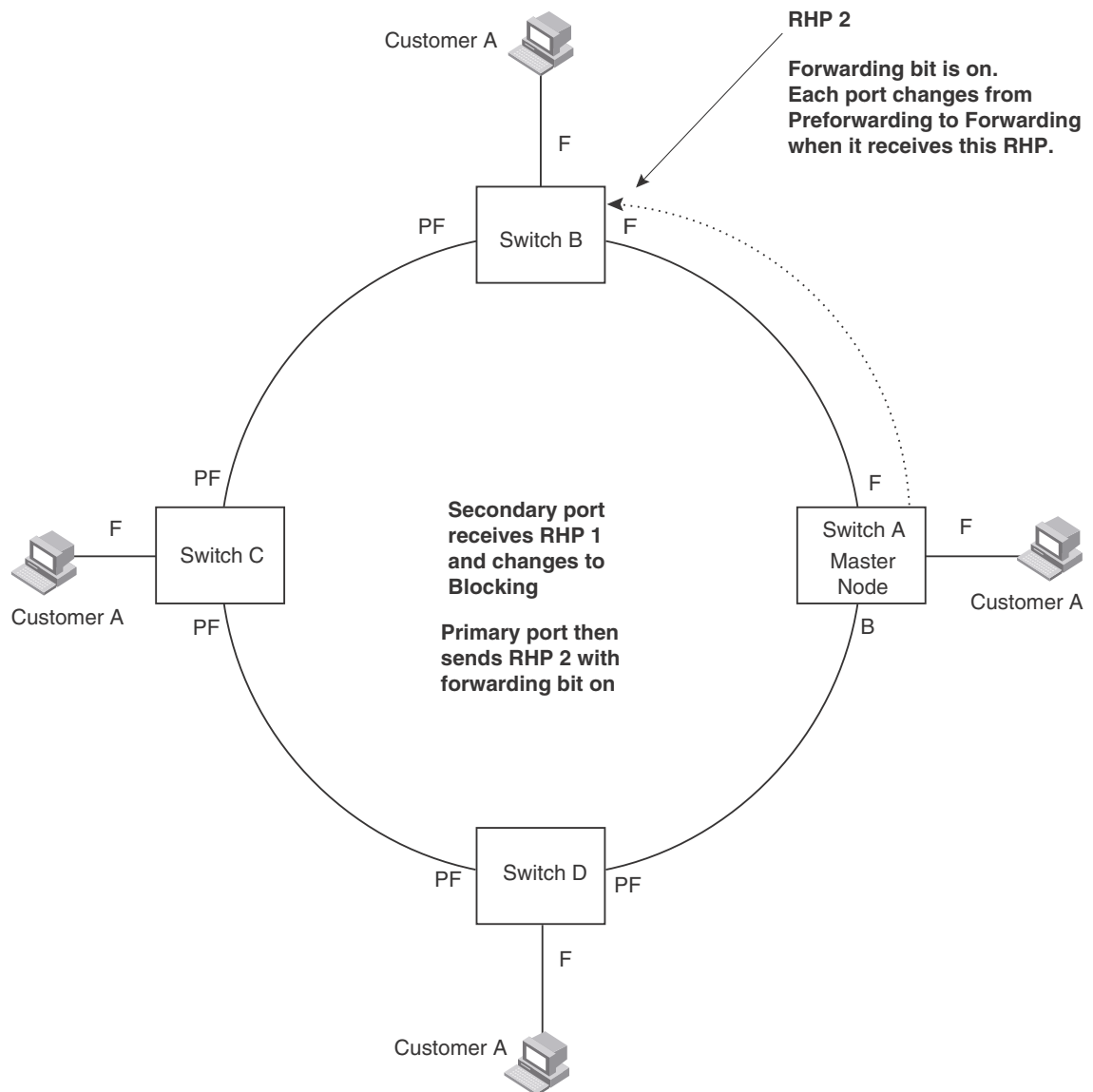
- **Forwarding (F)** – The interface can forward data as well as RHPs. An interface changes from Preforwarding to Forwarding when the port preforwarding time expires. This occurs if the port does not receive an RHP from the Master, or if the forwarding bit in the RHPs received by the port is off. This indicates a break in the ring. The port heals the ring by changing its state to Forwarding. The preforwarding time is the number of milliseconds the port will remain in the Preforwarding state before changing to the Forwarding state, even without receiving an RHP.
- **Blocking (B)** – The interface cannot forward data. Only the secondary interface on the Master node can be Blocking.

When MRP is enabled, all ports begin in the Preforwarding state. The primary interface on the Master node, although it is in the Preforwarding state like the other ports, immediately sends an RHP onto the ring. The secondary port on the Master node listens for the RHP.

- If the secondary port receives the RHP, all links in the ring are up and the port changes its state to Blocking. The primary port then sends another MRP with its forwarding bit set on. As each of the member ports receives the RHP, the ports change their state to Forwarding. Typically, this occurs in sub-second time. The ring very quickly enters the fully initialized state.
- If the secondary port does not receive the RHP by the time the preforwarding time expires, a break has occurred in the ring. The port changes its state to Forwarding. The member ports also change their states from Preforwarding to Forwarding as their preforwarding timers expire. The ring is not intact, but data can still travel among the nodes using the links that are up.

Figure 45 shows an example.

FIGURE 45 Metro ring – from preforwarding to forwarding

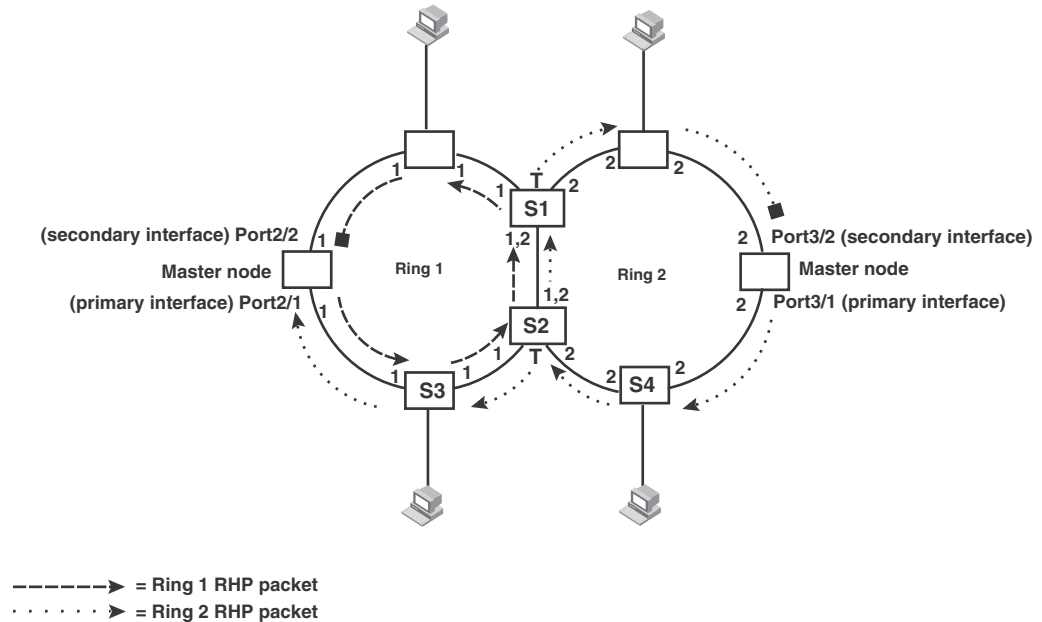


Each RHP also has a sequence number. MRP can use the sequence number to determine the round-trip time for RHPs in the ring. Refer to [“Metro Ring Protocol diagnostics”](#) on page 628.

RHP processing in MRP Phase 2

[Figure 46](#) shows an example of how RHP packets are processed normally in MRP rings with shared interfaces.

FIGURE 46 Flow of RHP packets on MRP rings with shared interfaces



Port 2/1 on Ring 1 master node is the primary interface of the master node. The primary interface forwards an RHP packet on the ring. Since all the interfaces on Ring 1 are regular ports, the RHP packet is forwarded to all the interfaces until it reaches Port 2/2, the secondary interface of the master node. Port 2/2 then blocks the packet to complete the process.

On Ring 2, Port 3/1, is the primary interface of the master node. It sends an RHP packet on the ring. Since all ports on S4 are regular ports, the RHP packet is forwarded on those interfaces. When the packet reaches S2, the receiving interface is a tunnel port. The port compares the packet priority to its priority. Since the packet priority is the same as the tunnel port priority, the packet is forwarded up the link shared by Rings 1 and 2.

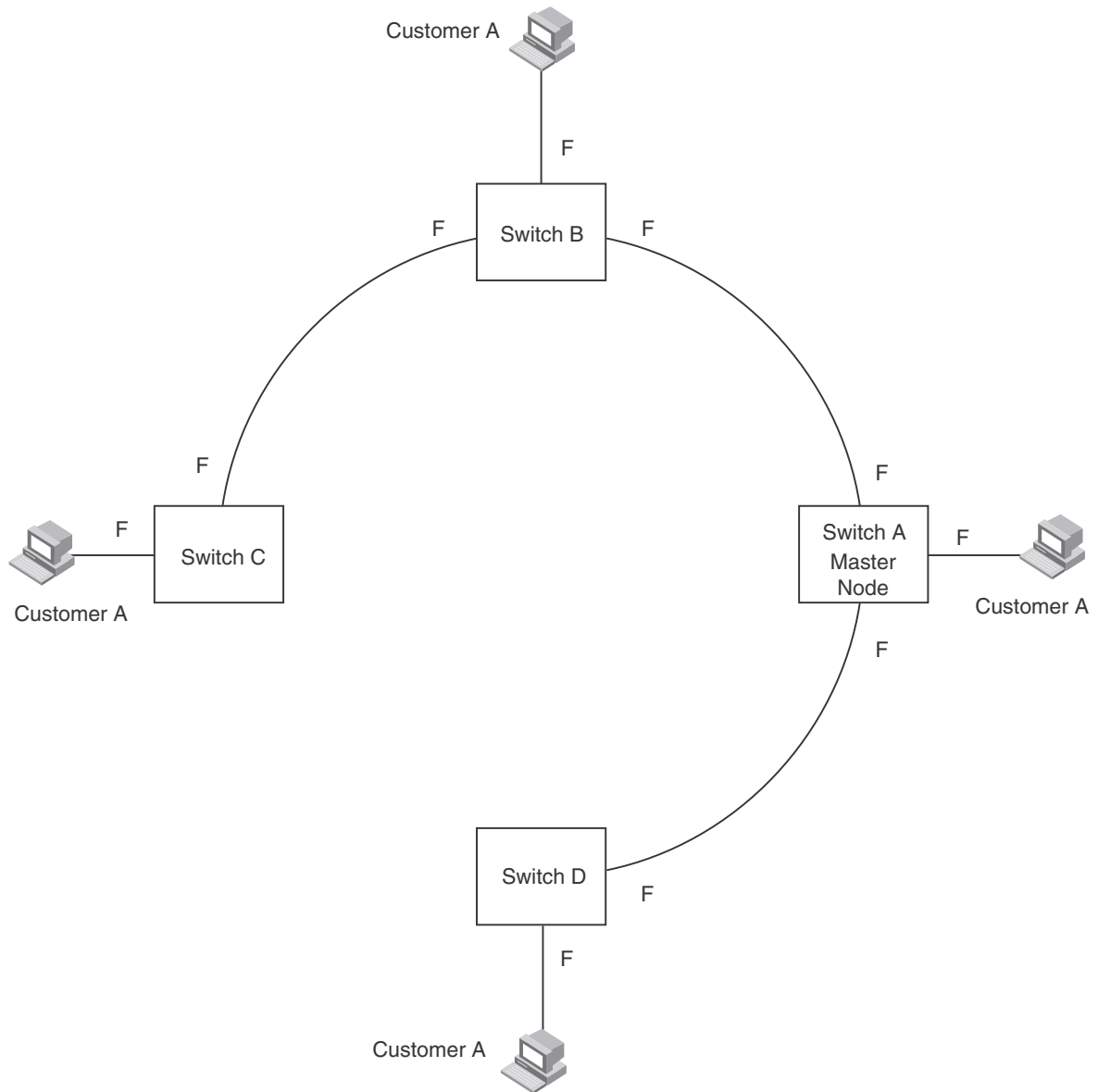
When the RHP packet reaches the interface on node S2 shared by Rings 1 and 2, the packet is forwarded since its priority is less than the interface priority. The packet continues to be forwarded to node S1 until it reaches the tunnel port on S1. That tunnel port determines that the RHP packet priority is equal to the port priority and forwards the packet. The RHP packet is forwarded to the remaining interfaces on Ring 2 until it reaches port 3/2, the secondary interface of the master node. Port 3/2 then blocks the packet to prevent a loop.

When the RHP packet from Ring 2 reached S2, it was also forwarded from S2 to S3 on Ring 1 since the port on S2 has a higher priority than the RHP packet. The packets is forwarded around Ring 1 until it reaches port 2/2, Ring 1 the secondary port. The RHP packet is then blocked by that port.

How ring breaks are detected and healed

Figure 47 shows ring interface states following a link break. MRP quickly heals the ring and preserves connectivity among the customer networks.

FIGURE 47 Metro ring – ring break



If a break in the ring occurs, MRP heals the ring by changing the states of some of the ring interfaces:

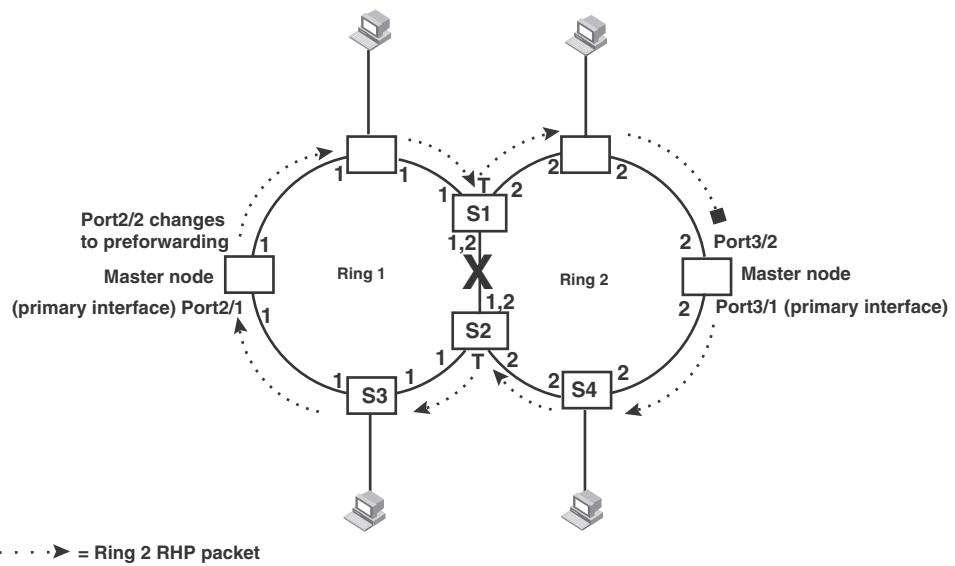
- **Blocking interface** – The Blocking interface on the Master node has a dead timer. If the dead time expires before the interface receives one of its ring RHPs, the interface changes state to Preforwarding. Once the secondary interface changes state to Preforwarding:
 - If the interface receives an RHP, the interface changes back to the Blocking state and resets the dead timer.
 - If the interface does not receive an RHP for its ring before the Preforwarding time expires, the interface changes to the Forwarding state, as shown in [Figure 47](#).
- **Forwarding interfaces** – Each member interface remains in the Forwarding state.

When the broken link is repaired, the link interfaces come up in the Preforwarding state, which allows RHPs to travel through the restored interfaces and reach the secondary interface on the Master node:

- If an RHP reaches the Master node secondary interface, the ring is intact. The secondary interface changes to Blocking. The Master node sets the forwarding bit on in the next RHP. When the restored interfaces receive this RHP, they immediately change state to Forwarding.
- If an RHP does not reach the Master node secondary interface, the ring is still broken. The Master node does not send an RHP with the forwarding bit on. In this case, the restored interfaces remain in the Preforwarding state until the preforwarding timer expires, then change to the Forwarding state.

If the link between **shared interfaces** breaks ([Figure 48](#)), the secondary interface on Ring 1 master node changes to a preforwarding state. The RHP packet sent by port 3/1 on Ring 2 is forwarded through the interfaces on S4, then to S2. The packet is then forwarded through S2 to S3, but not from S2 to S1 since the link between the two nodes is not available. When the packet reaches Ring 1 master node, the packet is forwarded through the secondary interface since it is currently in a preforwarding state. A secondary interface in preforwarding mode ignores any RHP packet that is not from its ring. The secondary interface changes to blocking mode only when the RHP packet forwarded by its primary interface is returned.

The packet then continues around Ring 1, through the interfaces on S1 to Ring 2 until it reaches Ring 2 master node. Port 3/2, the secondary interface on Ring 2 changes to blocking mode since it received its own packet, then blocks the packet to prevent a loop.

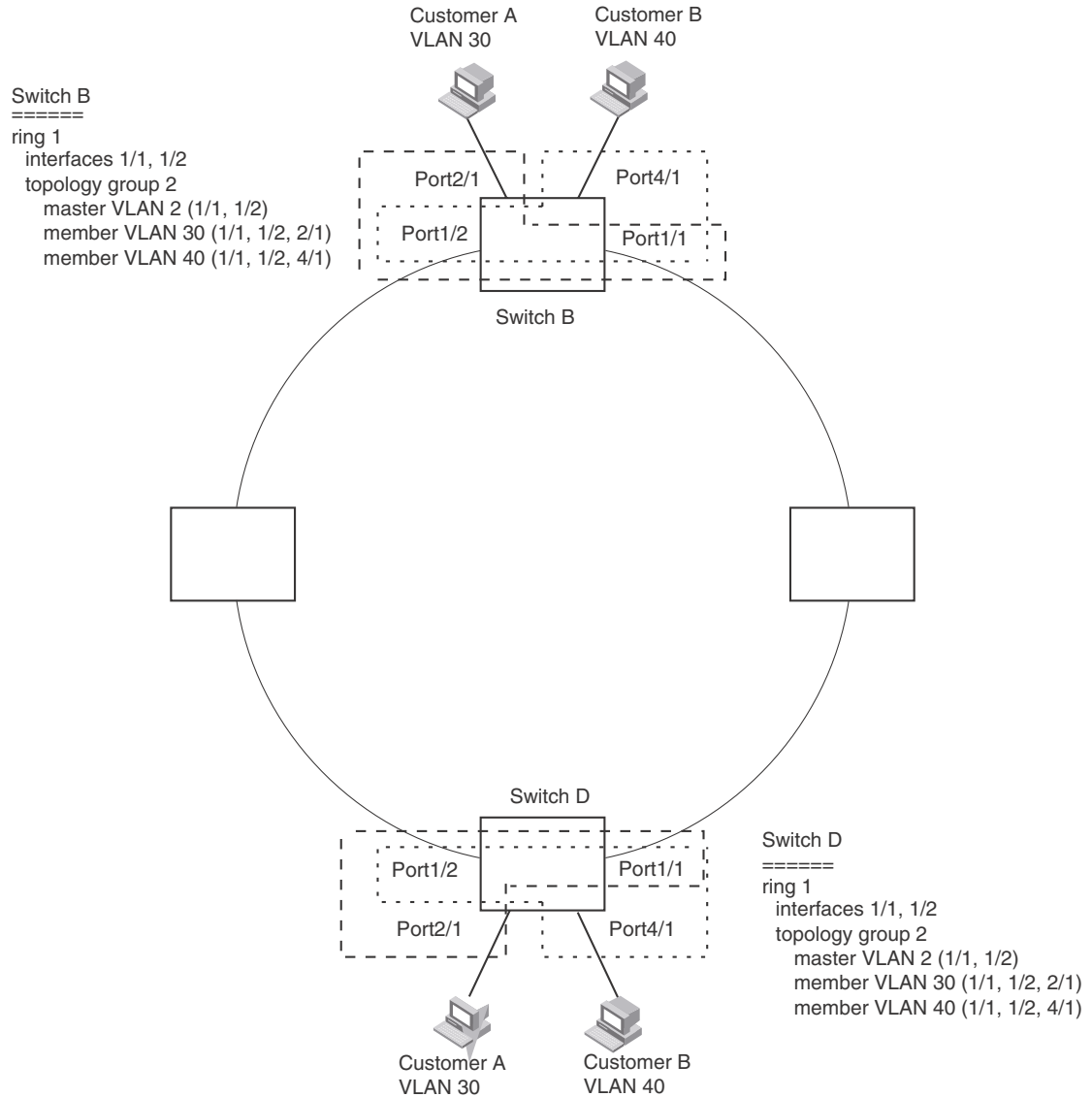
FIGURE 48 Flow of RHP packets when a link for shared interfaces breaks

RHP packets follow this flow until the link is restored; then the RHP packet returns to its normal flow as shown in [Figure 46](#).

Master VLANs and customer VLANs

All the ring ports must be in the same VLAN. Placing the ring ports in the same VLAN provides Layer 2 connectivity for a given customer across the ring. [Figure 49](#) shows an example.

FIGURE 49 Metro ring – ring VLAN and customer VLANs



Notice that each customer has their own VLAN. Customer A has VLAN 30 and Customer B has VLAN 40. Customer A host attached to Switch D can reach the Customer A host attached to Switch B at Layer 2 through the ring. Since Customer A and Customer B are on different VLANs, they will not receive each other traffic.

You can configure MRP separately on each customer VLAN. However, this is impractical if you have many customers. To simplify configuration when you have a lot of customers (and therefore a lot of VLANs), you can use a topology group.

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as MRP. A topology group contains a master VLAN and member VLANs. The master VLAN contains all the configuration parameters for the Layer 2 protocol (STP, MRP, or VSRP). The member VLANs use the Layer 2 configuration of the master VLAN.

In [Figure 49](#), VLAN 2 is the master VLAN and contains the MRP configuration parameters for ring 1. VLAN 30 and VLAN 40, the customer VLANs, are member VLANs in the topology group. Since a topology group is used, a single instance of MRP provides redundancy and loop prevention for both the customer VLANs.

If you use a topology group:

- The master VLAN must contain the ring interfaces. The ports must be tagged, since they will be shared by multiple VLANs.
- The member VLAN for a customer must contain the two ring interfaces and the interfaces for the customer. Since these interfaces are shared with the master VLAN, they must be tagged. Do not add another customer interfaces to the VLAN.

For more information about topology groups, refer to [“Topology groups”](#) on page 607.

Refer to [“MRP CLI example”](#) on page 631 for the configuration commands required to implement the MRP configuration shown in [Figure 49](#).

Metro Ring Protocol configuration

To configure Metro Ring Protocol (MRP), perform the following tasks. You need to perform the first task on only one of the nodes. Perform the remaining tasks on all the nodes.

NOTE

There are no new commands or parameters to configure MRP with shared interfaces (MRP Phase 2).

- Disable one of the ring interfaces. This prevents a Layer 2 loop from occurring while you are configuring the devices for MRP.
- Add an MRP ring to a port-based VLAN. When you add a ring, the CLI changes to the configuration level for the ring, where you can perform the following tasks.
 - Optionally, specify a name for the ring.
 - On the master node only, enable the device to be the master for the ring. Each ring can have only one master node.
 - Specify the MRP interfaces. Each device has two interfaces to an MRP ring.
 - Optionally, change the hello time and the preforwarding time. These parameters control how quickly failover occurs following a change in the state of a link in the ring.
 - Enable the ring.
- Optionally, add the ring VLAN to a topology group to add more VLANs to the ring. If you use a topology group, make sure you configure MRP on the group master VLAN. Refer to [“Topology groups”](#) on page 607.
- Re-enable the interface you disabled to prevent a Layer 2 loop. Once MRP is enabled, MRP will prevent the Layer 2 loop.
- On FCX devices, when configuring MRP-1 or MRP-2 rings on a VLAN, using the **metro-rings** command in addition to the **metro-ring** command is highly recommended. Since these devices do not support mac-range filtering, the **metro-rings** command greatly reduces the number of FDB entries.

Adding an MRP ring to a VLAN

To add an MRP ring to a VLAN, enter commands such as the following.

NOTE

If you plan to use a topology group to add VLANs to the ring, make sure you configure MRP on the topology group master VLAN.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#metro-ring 1
Brocade(config-vlan-2-mrp-1)#name CustomerA
Brocade(config-vlan-2-mrp-1)#master
Brocade(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)#enable
```

These commands configure an MRP ring on VLAN 2. The ring ID is 1, the ring name is CustomerA, and this node (this Brocade device) is the master for the ring. The ring interfaces are 1/1 and 1/2. Interface 1/1 is the primary interface and 1/2 is the secondary interface. The primary interface will initiate RHPs by default. The ring takes effect in VLAN 2.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#metro-ring 1
Brocade(config-vlan-2-mrp-1)#name CustomerA
Brocade(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)#enable
Brocade(config-vlan-2-mrp-1)#metro-ring 2
Brocade(config-vlan-2-mrp-2)#name CustomerB
Brocade(config-vlan-2-mrp-2)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-2)#enable
```

Syntax: [no] metro-ring <ring id>

The <ring-id> parameter specifies the ring ID. In the FWS, the ring ID can be a value from 1 – 255. Otherwise, the <ring-id> can be from 1 – 1023; ID 256 is reserved for VSRP.

On FWS, FCX, and ICX devices, enter the **metro-rings** in addition to the **metro-ring** command as shown below.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#metro-rings 1 2
Brocade(config-vlan-2)#metro-ring 1
Brocade(config-vlan-2-mrp-1)#name CustomerA
Brocade(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)#enable
Brocade(config-vlan-2-mrp-1)#metro-ring 2
Brocade(config-vlan-2-mrp-2)#name CustomerB
Brocade(config-vlan-2-mrp-2)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-2)#enable
```

Syntax: [no] metro-rings <ring id> <ring id> . . .

The <ring id> variables identify the metro rings you want to configure on the VLAN.

Syntax: [no] name <string>

The <string> parameter specifies a name for the ring. The name is optional, but it can be up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] master

Configures this node as the master node for the ring. Enter this command only on one node in the ring. The node is a member (non-master) node by default.

Syntax: `[no] ring-interface ethernet <primary-if> ethernet <secondary-if>`

The **ethernet** <primary-if> parameter specifies the primary interface. On the master node, the primary interface is the one that originates RHPs. Ring control traffic and Layer 2 data traffic will flow in the outward direction from this interface by default. On member nodes, the direction of traffic flow depends on the traffic direction selected by the master node. Therefore, on a member node, the order in which you enter the interfaces does not matter.

The **ethernet** <secondary-if> parameter specifies the secondary interface.

NOTE

To take advantage of every interface in a Metro network, you can configure another MRP ring and either configure a different Master node for the ring or reverse the configuration of the primary and secondary interfaces on the Master node. Configuring multiple rings enables you to use all the ports in the ring. The same port can forward traffic one ring while blocking traffic for another ring.

Syntax: `[no] enable`

The **enable** command enables the ring.

Changing the hello and preforwarding times

You also can change the RHP hello time and preforwarding time. To do so, enter commands such as the following.

```
Brocade(config-vlan-2-mrp-1)#hello-time 200
Brocade(config-vlan-2-mrp-1)#preforwarding-time 400
```

These commands change the hello time to 200 ms and change the preforwarding time to 400 ms.

Syntax: `[no] hello-time <ms>`

Syntax: `[no] preforwarding-time <ms>`

The <ms> specifies the number of milliseconds. For the hello time, you can specify from 100 – 1000 (one second). The default hello time is 100 ms. The preforwarding time can be from 200 – 5000 ms, but must be at least twice the value of the hello time and must be a multiple of the hello time. The default preforwarding time is 300 ms. A change to the hello time or preforwarding time takes effect as soon as you enter the command.

Configuration notes for changing the hello and preforwarding times

- The preforwarding time must be at least twice the value of the hello time and must be a multiple of the hello time.
- If UDLD is also enabled on the device, Brocade recommends that you set the MRP preforwarding time slightly higher than the default of 300 ms; for example, to 400 or 500 ms.
- You can use MRP ring diagnostics to determine whether you need to change the hello time and preforwarding time. Refer to [“Metro Ring Protocol diagnostics”](#).

Metro Ring Protocol diagnostics

The Metro Ring Protocol (MRP) diagnostics feature calculates how long it takes for RHP packets to travel through the ring. When you enable MRP diagnostics, the software tracks RHP packets according to their sequence numbers and calculates how long it takes an RHP packet to travel one time through the entire ring. When you display the diagnostics, the CLI shows the average round-trip time for the RHP packets sent since you enabled diagnostics. The calculated results have a granularity of 1 microsecond.

Enabling MRP diagnostics

To enable MRP diagnostics for a ring, enter the following command on the Master node, at the configuration level for the ring.

```
Brocade(config-vlan-2-mrp-1)#diagnostics
```

Syntax: [no] diagnostics

NOTE

This command is valid only on the master node.

Displaying MRP diagnostics

To display MRP diagnostics results, enter the following command on the Master node.

```
Brocade#show metro 1 diag
```

```
Metro Ring 1 - CustomerA
=====
diagnostics results
```

Ring id	Diag state	RHP average time(microsec)	Recommended hello time(ms)	Recommended Prefwing time(ms)
2	enabled	125	100	300

Diag frame sent	Diag frame lost
1230	0

Syntax: show metro <ring-id> diag

This display shows the following information.

TABLE 116 CLI display of MRP ring diagnostic information

Field	Description
Ring id	The ring ID.
Diag state	The state of ring diagnostics.
RHP average time	The average round-trip time for an RHP packet on the ring. The calculated time has a granularity of 1 microsecond.
Recommended hello time	The hello time recommended by the software based on the RHP average round-trip time.
Recommended Prefwing time	The preforwarding time recommended by the software based on the RHP average round-trip time.

TABLE 116 CLI display of MRP ring diagnostic information (Continued)

Field	Description
Diag frame sent	The number of diagnostic RHPs sent for the test.
Diag frame lost	The number of diagnostic RHPs lost during the test.

If the recommended hello time and preforwarding time are different from the actual settings and you want to change them, refer to [“Metro Ring Protocol configuration”](#) on page 625.

Displaying MRP information

You can display the following MRP information:

- Topology group configuration information
- Ring configuration information and statistics

Displaying topology group information

To display topology group information, enter the following command.

Syntax: `show topology-group [<group-id>]`

Refer to [“Displaying topology group information”](#) on page 610 for more information.

Displaying ring information

To display ring information, enter the following command.

```
Brocade#show metro
```

```

Metro Ring 1
=====
Ring      State      Ring      Master      Topo      Hello      Prefwing
id        enabled   role      vlan        group     time(ms)   time(ms)
2         enabled   member    2           not conf  100        300

Ring interfaces      Interface role      Forwarding state      Active interface
Interface Type
ethernet 1/1         primary             disabled              none
Regular
ethernet 1/2         secondary           forwarding            ethernet 2          Tunnel

RHPs sent      RHPs rcvd      TC RHPs rcvd      State changes
3              0              0                 4
```

Syntax: `show metro [<ring-id>]`

This display shows the following information.

TABLE 117 CLI display of MRP ring information

Field	Description
Ring id	The ring ID
State	The state of MRP. The state can be one of the following: <ul style="list-style-type: none"> • enabled – MRP is enabled • disabled – MRP is disabled
Ring role	Whether this node is the master for the ring. The role can be one of the following: <ul style="list-style-type: none"> • master • member
Master vlan	The ID of the master VLAN in the topology group used by this ring. If a topology group is used by MRP, the master VLAN controls the MRP settings for all VLANs in the topology group. NOTE: The topology group ID is 0 if the MRP VLAN is not the master VLAN in a topology group. Using a topology group for MRP configuration is optional.
Topo group	The topology group ID.
Hello time	The interval, in milliseconds, at which the Forwarding port on the ring master node sends Ring Hello Packets (RHPs).
Prefwing time	The number of milliseconds an MRP interface that has entered the Preforwarding state will wait before changing to the Forwarding state. If a member port in the Preforwarding state does not receive an RHP within the Preforwarding time (Prefwing time), the port assumes that a topology change has occurred and changes to the Forwarding state. The secondary port on the Master node changes to Blocking if it receives an RHP, but changes to Forwarding if the port does not receive an RHP before the preforwarding time expires. NOTE: A member node Preforwarding interface also changes from Preforwarding to Forwarding if it receives an RHP whose forwarding bit is on.
Ring interfaces	The device two interfaces with the ring. NOTE: If the interfaces are trunk groups, only the primary ports of the groups are listed.
Interface role	The interface role can be one of the following: <ul style="list-style-type: none"> • primary <ul style="list-style-type: none"> • Master node – The interface generates RHPs. • Member node – The interface forwards RHPs received on the other interface (the secondary interface). • secondary – The interface does not generate RHPs. <ul style="list-style-type: none"> • Master node – The interface listens for RHPs. • Member node – The interface receives RHPs.
Forwarding state	Whether MRP Forwarding is enabled on the interface. The forwarding state can be one of the following: <ul style="list-style-type: none"> • blocking – The interface is blocking Layer 2 data traffic and RHPs • disabled – The interface is down • forwarding – The interface is forwarding Layer 2 data traffic and RHPs • preforwarding – The interface is listening for RHPs but is blocking Layer 2 data traffic

TABLE 117 CLI display of MRP ring information (Continued)

Field	Description
Active interface	The physical interfaces that are sending and receiving RHPs. NOTE: If a port is disabled, its state is shown as “disabled”. NOTE: If an interface is a trunk group, only the primary port of the group is listed.
Interface Type	Shows if the interface is a regular port or a tunnel port.
RHPs sent	The number of RHPs sent on the interface. NOTE: This field applies only to the master node. On non-master nodes, this field contains 0. This is because the RHPs are forwarded in hardware on the non-master nodes.
RHPs rcvd	The number of RHPs received on the interface. NOTE: On most Brocade devices, this field applies only to the master node. On non-master nodes, this field contains 0. This is because the RHPs are forwarded in hardware on the non-master nodes. However, on the FastIron devices, the RHP received counter on non-master MRP nodes increment. This is because, on FastIron devices, the CPU receives a copy of the RHPs forwarded in hardware.
TC RHPs rcvd	The number of Topology Change RHPs received on the interface. A Topology Change RHP indicates that the ring topology has changed.
State changes	The number of MRP interface state changes that have occurred. The state can be one of the states listed in the Forwarding state field.
Interface Type	Shows if the interface is a regular port or a tunnel port.

MRP CLI example

The following examples show the CLI commands required to implement the MRP configuration shown in [Figure 49](#) on page 623.

NOTE

For simplicity, the figure shows the VLANs on only two switches. The CLI examples implement the ring on all four switches.

MRP commands on Switch A (master node)

The following commands configure a VLAN for the ring. The ring VLAN must contain both of the node interfaces with the ring. Add these interfaces as tagged interfaces, since the interfaces also must be in each of the customer VLANs configured on the node.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-2)#metro-ring 1
Brocade(config-vlan-2-mrp-1)#name "Metro A"
Brocade(config-vlan-2-mrp-1)#master
Brocade(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)#enable
Brocade(config-vlan-2-mrp-1)#exit
Brocade(config-vlan-2)#exit
```

The following commands configure the customer VLANs. The customer VLANs must contain both the ring interfaces as well as the customer interfaces.

```
Brocade(config)#vlan 30
Brocade(config-vlan-30)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-30)#tag ethernet 2/1
Brocade(config-vlan-30)#exit
Brocade(config)#vlan 40
Brocade(config-vlan-40)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-40)#tag ethernet 4/1
Brocade(config-vlan-40)#exit
```

The following commands configure topology group 1 on VLAN 2. The master VLAN is the one that contains the MRP configuration. The member VLANs use the MRP parameters of the master VLAN. The control interfaces (the ones shared by the master VLAN and member VLAN) also share MRP state.

```
Brocade(config)#topology-group 1
Brocade(config-topo-group-1)#master-vlan 2
Brocade(config-topo-group-1)#member-vlan 30
Brocade(config-topo-group-1)#member-vlan 40
```

MRP commands on Switch B

The commands for configuring Switches B, C, and D are similar to the commands for configuring Switch A, with two differences: the nodes are not configured to be the ring master. Omitting the **master** command is required for non-master nodes.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-2)#metro-ring 1
Brocade(config-vlan-2-mrp-1)#name "Metro A"
Brocade(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)#enable
Brocade(config-vlan-2)#exit
Brocade(config)#vlan 30
Brocade(config-vlan-30)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-30)#tag ethernet 2/1
Brocade(config-vlan-30)#exit
Brocade(config)#vlan 40
Brocade(config-vlan-40)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-40)#tag ethernet 4/1
Brocade(config-vlan-40)#exit
Brocade(config)#topology-group 1
Brocade(config-topo-group-1)#master-vlan 2
Brocade(config-topo-group-1)#member-vlan 30
Brocade(config-topo-group-1)#member-vlan 40
```

MRP commands on Switch C

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-2)#metro-ring 1
Brocade(config-vlan-2-mrp-1)#name "Metro A"
Brocade(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)#enable
Brocade(config-vlan-2)#exit
```

```

Brocade(config)#vlan 30
Brocade(config-vlan-30)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-30)#tag ethernet 2/1
Brocade(config-vlan-30)#exit
Brocade(config)#vlan 40
Brocade(config-vlan-40)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-40)#tag ethernet 4/1
Brocade(config-vlan-40)#exit
Brocade(config)#topology-group 1
Brocade(config-topo-group-1)#master-vlan 2
Brocade(config-topo-group-1)#member-vlan 30
Brocade(config-topo-group-1)#member-vlan 40

```

MRP commands on Switch D

```

Brocade(config)#vlan 2
Brocade(config-vlan-2)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-2)#metro-ring 1
Brocade(config-vlan-2-mrp-1)#name "Metro A"
Brocade(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)#enable
Brocade(config-vlan-2)#exit
Brocade(config)#vlan 30
Brocade(config-vlan-30)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-30)#tag ethernet 2/1
Brocade(config-vlan-30)#exit
Brocade(config)#vlan 40
Brocade(config-vlan-40)#tag ethernet 1/1 to 1/2
Brocade(config-vlan-40)#tag ethernet 4/1
Brocade(config-vlan-40)#exit
Brocade(config)#topology-group 1
Brocade(config-topo-group-1)#master-vlan 2
Brocade(config-topo-group-1)#member-vlan 30
Brocade(config-topo-group-1)#member-vlan 40

```

VSRP

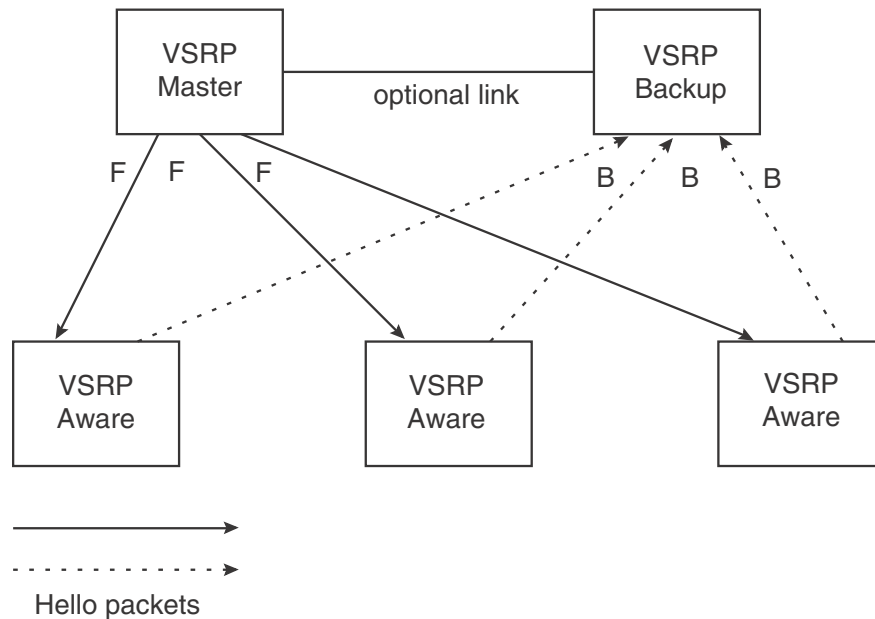
Virtual Switch Redundancy Protocol (VSRP) is a Brocade proprietary protocol that provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Based on the Brocade Virtual Router Redundancy Protocol Extended (VRRP-E), VSRP provides one or more backups for a Layer 2 Switch or Layer 3 Switch. If the active Layer 2 Switch or Layer 3 Switch becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network.

The FastIron family of switches support full VSRP as well as **VSRP-awareness**. A Brocade device that is not itself configured for VSRP but is connected to a Brocade device that is configured for VSRP, is **VSRP aware**.

You can use VSRP for Layer 2, Layer 3, or for both layers. On Layer 3 Switches, Layer 2 and Layer 3 share the same VSRP configuration information. On Layer 2 Switches, VSRP applies only to Layer 2.

Figure 50 shows an example of a VSRP configuration.

FIGURE 50 VSRP mesh – redundant paths for Layer 2 and Layer 3 traffic



In this example, two Brocade devices are configured as redundant paths for VRID 1. On each of the devices, a Virtual Router ID (VRID) is configured on a port-based VLAN. Since VSRP is primarily a Layer 2 redundancy protocol, the VRID applies to the entire VLAN. However, you can selectively remove individual ports from the VRID if needed.

Following Master election (described below), one of the Brocade devices becomes the Master for the VRID and sets the state of all the VLAN ports to Forwarding. The other device is a Backup and sets all the ports in its VRID VLAN to Blocking.

If a failover occurs, the Backup becomes the new Master and changes all its VRID ports to the Forwarding state.

Other Brocade devices can use the redundant paths provided by the VSRP devices. In this example, three Brocade devices use the redundant paths. A Brocade device that is not itself configured for VSRP but is connected to a Brocade device that is configured for VSRP, is **VSRP aware**. In this example, the three Brocade devices connected to the VSRP devices are VSRP aware. A Brocade device that is VSRP aware can failover its link to the new Master in sub-second time, by changing the MAC address associated with the redundant path.

When you configure VSRP, make sure each of the non-VSRP Brocade devices connected to the VSRP devices has a separate link to each of the VSRP devices.

VSRP configuration notes and feature limitations

- VSRP and 802.1Q-n-Q tagging are not supported together on the same device.
- VSRP and Super Aggregated VLANs are not supported together on the same device.
- When VSRP or VSRP-aware is configured on a VLAN, the VLAN will support IGMP snooping version 2 only. IGMP version 3 will not be supported on the VLAN.

Layer 2 and Layer 3 redundancy

You can configure VSRP to provide redundancy for Layer 2 only or also for Layer 3:

- **Layer 2 only** – The Layer 2 links are backed up but specific IP addresses are not backed up.
- **Layer 2 and Layer 3** – The Layer 2 links are backed up and a specific IP address is also backed up. Layer 3 VSRP is the same as VRRP-E. However, using VSRP provides redundancy at both layers at the same time.

Layer 2 Switches support Layer 2 VSRP only. Layer 3 Switches support Layer 2 and Layer 3 redundancy. You can configure a Layer 3 Switch for either Layer 2 only or Layer 2 and Layer 3. To configure for Layer 3, specify the IP address you are backing up.

NOTE

If you want to provide Layer 3 redundancy only, disable VSRP and use VRRP-E.

Master election and failover

Each VSRP device advertises its VSRP priority in Hello messages. During Master election, the VSRP device with the highest priority for a given VRID becomes the Master for that VRID. After Master election, the Master sends Hello messages at regular intervals to inform the Backups that the Master is healthy.

If there is a tie for highest VSRP priority, the tie is resolved as follows:

- **Layer 2 Switches** – The Layer 2 Switch with the higher management IP address becomes the Master.
 - Switches with management IP addresses are preferred over switches without management IP addresses.
 - If neither of the switches has a management IP address, then the switch with the higher MAC address becomes the Master. (VSRP compares the MAC addresses of the ports configured for the VRID, not the base MAC addresses of the switches.)
- **Layer 3 Switches** – The Layer 3 Switch whose virtual routing interface has a higher IP address becomes the master.

VSRP failover

Each Backup listens for Hello messages from the Master. The Hello messages indicate that the Master is still available. If the Backups stop receiving Hello messages from the Master, the election process occurs again and the Backup with the highest priority becomes the new Master.

Each Backup waits for a specific period of time, the Dead Interval, to receive a new Hello message from the Master. If the Backup does not receive a Hello message from the Master by the time the Dead Interval expires, the Backup sends a Hello message of its own, which includes the Backup's VSRP priority, to advertise the Backup's intent to become the Master. If there are multiple Backups for the VRID, each Backup sends a Hello message.

When a Backup sends a Hello message announcing its intent to become the Master, the Backup also starts a hold-down timer. During the hold-down time, the Backup listens for a Hello message with a higher priority than its own.

- If the Backup receives a Hello message with a higher priority than its own, the Backup resets its Dead Interval and returns to normal Backup status.

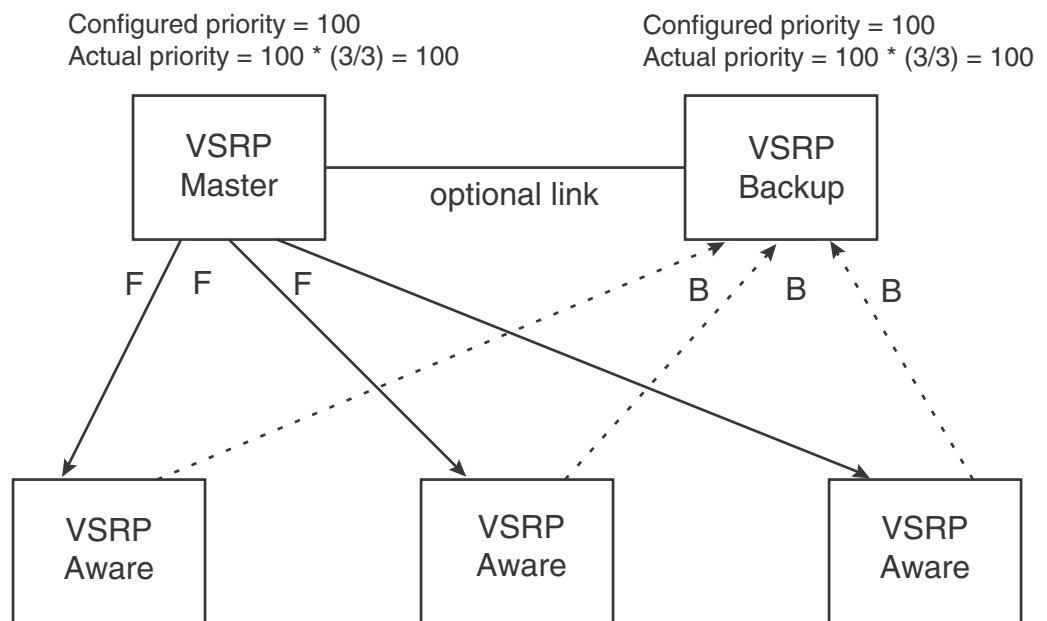
- If the Backup does not receive a Hello message with a higher priority than its own by the time the hold-down timer expires, the Backup becomes the new Master and starts forwarding Layer 2 traffic on all ports.

If you increase the timer scale value, each timer value is divided by the scale value. To achieve sub-second failover times, you can change the scale to a value up to 10. This shortens all the VSRP timers to 10 percent of their configured values.

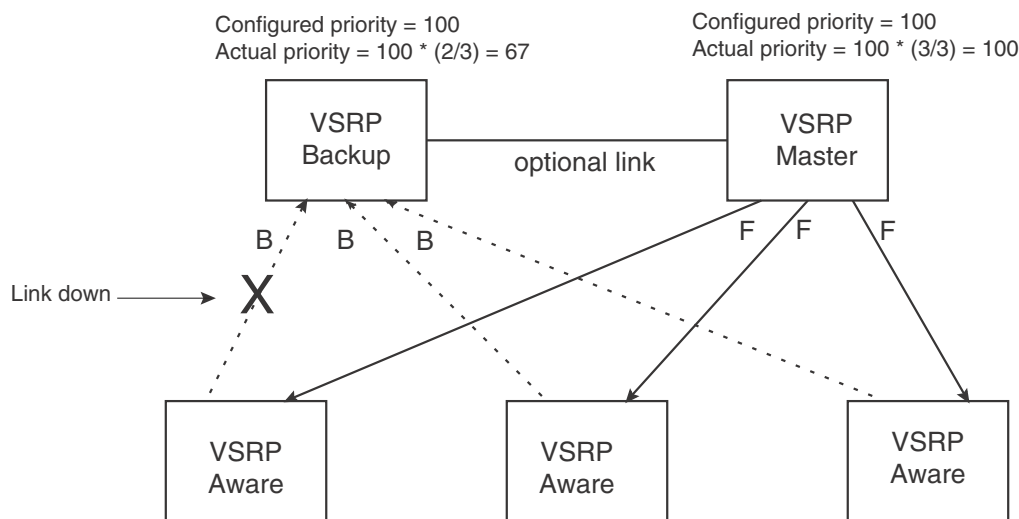
VSRP priority calculation

Each VSRP device has a VSRP priority for each VRID and its VLAN. The VRID is used during Master election for the VRID. By default, a device VSRP priority is the value configured on the device (which is 100 by default). However, to ensure that a Backup with a high number of up ports for a given VRID is elected, the device reduces the priority if a port in the VRID VLAN goes down. For example, if two Backups each have a configured priority of 100, and have three ports in VRID 1 in VLAN 10, each Backup begins with an equal priority, 100. This is shown in [Figure 51](#)

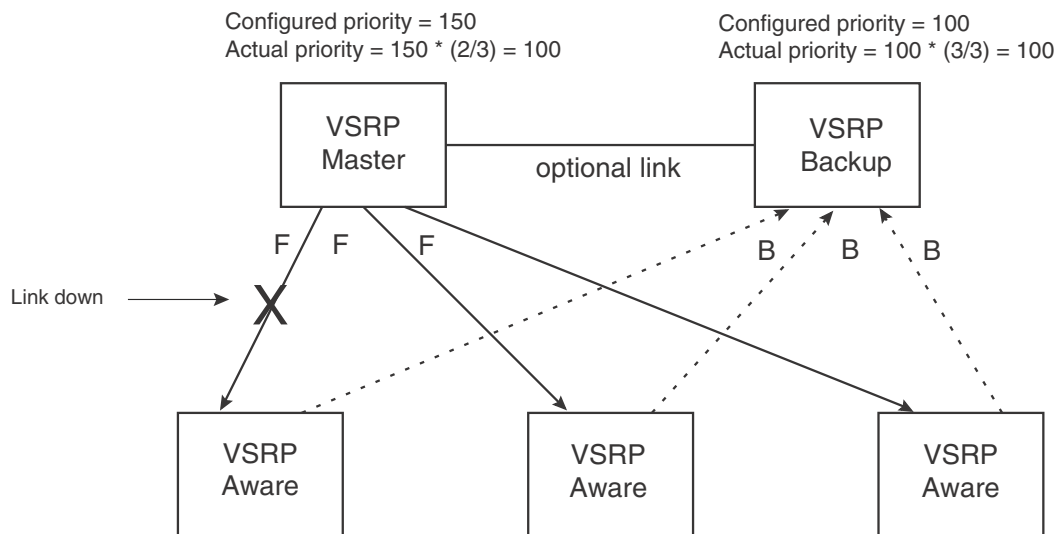
FIGURE 51 VSRP priority



However, if one of the VRID ports goes down on one of the Backups, that Backup priority is reduced. If the Master priority is reduced enough to make the priority lower than a Backup priority, the VRID fails over to the Backup. [Figure 52](#) shows an example.

FIGURE 52 VSRP priority recalculation

You can reduce the sensitivity of a VSRP device to failover by increasing its configured VSRP priority. For example, you can increase the configured priority of the VSRP device on the left in [Figure 52](#) to 150. In this case, failure of a single link does not cause failover. The link failure caused the priority to be reduced to 100, which is still equal to the priority of the other device. This is shown in [Figure 53](#).

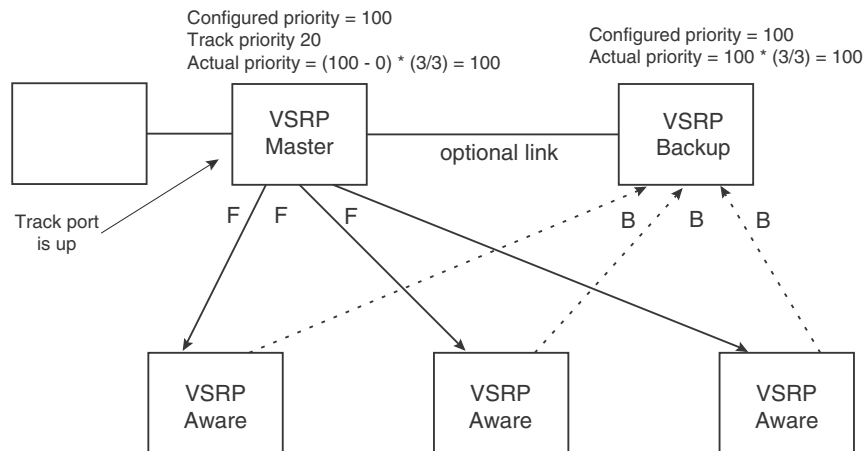
FIGURE 53 VSRP priority bias

Track ports

Optionally, you can configure track ports to be included during VSRP priority calculation. In VSRP, a **track port** is a port that is not a member of the VRID VLAN, but whose state is nonetheless considered when the priority is calculated. Typically, a track port represents the exit side of traffic received on the VRID ports. By default, no track ports are configured.

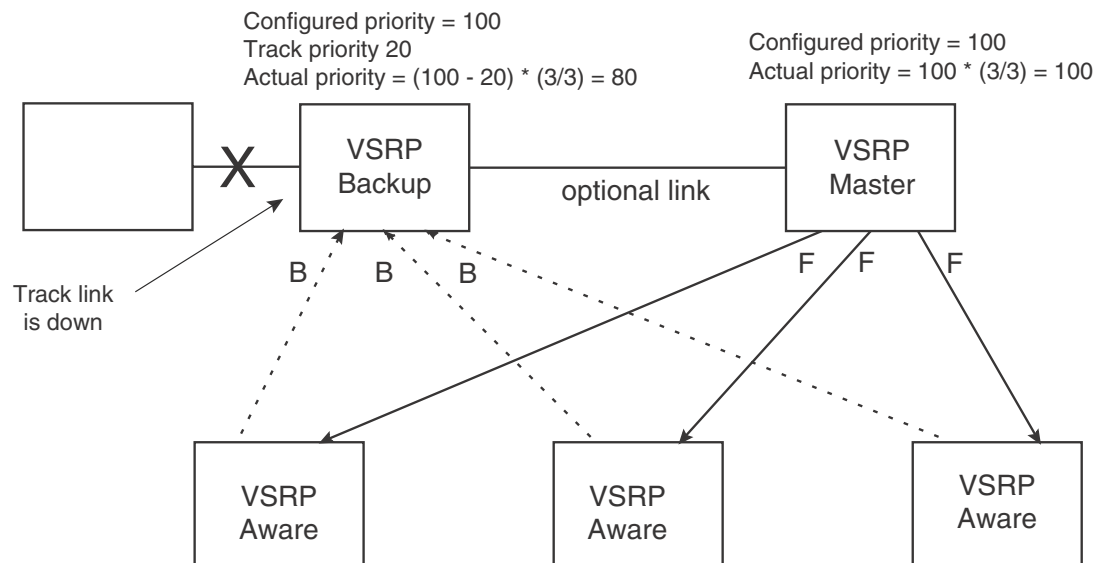
When you configure a track port, you assign a priority value to the port. If the port goes down, VSRP subtracts the track port priority value from the configured VSRP priority. For example, if the you configure a track port with priority 20 and the configured VSRP priority is 100, the software subtracts 20 from 100 if the track port goes down, resulting in a VSRP priority of 80. The new priority value is used when calculating the VSRP priority. [Figure 54](#) shows an example.

FIGURE 54 Track port priority



In [Figure 54](#), the track port is up. Since the port is up, the track priority does not affect the VSRP priority calculation. If the track port goes down, the track priority does affect VSRP priority calculation, as shown in [Figure 55](#).

FIGURE 55 Track port priority subtracted during priority calculation



MAC address failover on VSRP-aware devices

VSRP-aware devices maintain a record of each VRID and its VLAN. When the device has received a Hello message for a VRID in a given VLAN, the device creates a record for that VRID and VLAN and includes the port number in the record. Each subsequent time the device receives a Hello message for the same VRID and VLAN, the device checks the port number:

- If the port number is the same as the port that previously received a Hello message, the VSRP-aware device assumes that the message came from the same VSRP Master that sent the previous message.
- If the port number does not match, the VSRP-aware device assumes that a VSRP failover has occurred to a new Master, and moves the MAC addresses learned on the previous port to the new port.

The VRID records age out if unused. This can occur if the VSRP-aware device becomes disconnected from the Master. The VSRP-aware device will wait for a Hello message for the period of time equal to the following.

$$\text{VRID Age} = \text{Dead Interval} + \text{Hold-down Interval} + (3 \times \text{Hello Interval})$$

The values for these timers are determined by the VSRP device sending the Hello messages. If the Master uses the default timer values, the age time for VRID records on the VSRP-aware devices is as follows.

$$3 + 3 + (3 \times 1) = 9 \text{ seconds}$$

In this case, if the VSRP-aware device does not receive a new Hello message for a VRID in a given VLAN, on any port, the device assumes the connection to the Master is unavailable and removes the VRID record.

VSRP interval timers

The VSRP Hello interval, Dead interval, Backup Hello interval, and Hold-down interval timers are individually configurable. You also can easily change all the timers at the same time while preserving the ratios among their values. To do so, change the timer scale. The **timer scale** is a value used by the software to calculate the timers. The software divides a timer value by the timer scale value. By default, the scale is 1. This means the VSRP timer values are the same as the values in the configuration.

VSRP-aware security features

This feature protects against unauthorized VSRP hello packets by enabling you to configure VSRP-aware security parameters. Without VSRP-aware security, a VSRP-aware device passively learns the authentication method conveyed by the received VSRP hello packet. The VSRP-aware device then stores the authentication method until it ages out with the aware entry.

The VSRP-aware security feature enables you to perform the following:

- Define the specific authentication parameters that a VSRP-aware device will use on a VSRP backup switch. The authentication parameters that you define will not age out.
- Define a list of ports that have authentic VSRP backup switch connections. For ports included in the list, the VSRP-aware switch will process VSRP hello packets using the VSRP-aware security configuration. Conversely, for ports not included in the list, the VSRP-aware switch will not use the VSRP-aware security configuration.

If VSRP hello packets do not meet the acceptance criteria, the VSRP-aware device forwards the packets normally, without any VSRP-aware security processing.

To configure VSRP-Aware Security features, refer to “[Configuring security features on a VSRP-aware device](#)” on page 645.

VSRP parameters

[Table 118](#) lists the VSRP parameters.

TABLE 118 VSRP parameters

Parameter	Description	Default	For more information
Protocol	VSRP state NOTE: On a Layer 3 Switch, you must disable VSRP to use VRRP-E or VRRP.	Enabled	page 643
Virtual Router ID (VRID)	The ID of the virtual switch you are creating by configuring multiple devices as redundant links. You must configure the same VRID on each device that you want to use to back up the links.	None	page 642
Timer scale	The value used by the software to calculate all VSRP timers. Increasing the timer scale value decreases the length of all the VSRP timers equally, without changing the ratio of one timer to another.	1	page 644
Interface parameters			
Authentication type	The type of authentication the VSRP devices use to validate VSRP packets. On Layer 3 Switches, the authentication type must match the authentication type the VRID port uses with other routing protocols such as OSPF. <ul style="list-style-type: none"> No authentication – The interfaces do not use authentication. Simple – The interface uses a simple text-string as a password in packets sent on the interface. If the interface uses simple password authentication, the VRID configured on the interface must use the same authentication type and the same password. NOTE: MD5 is not supported.	No authentication	page 644
VSRP-Aware Security Parameters			
VSRP-Aware Authentication type	The type of authentication the VSRP-aware devices will use on a VSRP backup switch: <ul style="list-style-type: none"> No authentication – The device does not accept incoming packets that have authentication strings. Simple – The device uses a simple text-string as the authentication string for accepting incoming packets. 	Not configured	page 645
VRID parameters			
VSRP device type	Whether the device is a VSRP Backup for the VRID. All VSRP devices for a given VRID are Backups.	Not configured	page 642

TABLE 118 VSRP parameters (Continued)

Parameter	Description	Default	For more information
VSRP ports	The ports in the VRID VLAN that you want to use as VRID interfaces. You can selectively exclude individual ports from VSRP while allowing them to remain in the VLAN.	All ports in the VRID VLAN	page 646
VRID IP address	<p>A gateway address you are backing up. Configuring an IP address provides VRRP-E Layer 3 redundancy in addition to VSRP Layer 2 redundancy.</p> <p>The VRID IP address must be in the same subnet as a real IP address configured on the VSRP interface, but cannot be the same as a real IP address configured on the interface.</p> <p>NOTE: This parameter is valid only on Layer 3 Switches.</p>	None	page 646
Backup priority	<p>A numeric value that determines a Backup preferability for becoming the Master for the VRID. During negotiation, the device with the highest priority becomes the Master.</p> <p>In VSRP, all devices are Backups and have the same priority by default.</p> <p>If two or more Backups are tied with the highest priority, the Backup with the highest IP address becomes the Master for the VRID.</p>	100 for all Backups	page 646
Preference of timer source	<p>When you save a Backup configuration, the software can save the configured VSRP timer values or the VSRP timer values received from the Master.</p> <p>Saving the current timer values instead of the configured ones helps ensure consistent timer usage for all the VRID devices.</p> <p>NOTE: The Backup always gets its timer scale value from the Master.</p>	Configured timer values are saved	page 647
Time-to-Live (TTL)	The maximum number of hops a VSRP Hello packet can traverse before being dropped. You can specify from 1 – 255.	2	page 647
Hello interval	<p>The amount of time between Hello messages from the Master to the Backups for a given VRID.</p> <p>The interval can be from 1 – 84 seconds.</p>	One second	page 648
Dead interval	<p>The amount of time a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active.</p> <p>If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID.</p>	Three times the Hello Interval	page 648
Backup Hello state and interval	<p>The amount of time between Hello messages from a Backup to the Master.</p> <p>The message interval can be from 60 – 3600 seconds.</p> <p>You must enable the Backup to send the messages. The messages are disabled by default on Backups. The current Master sends Hello messages by default.</p>	Disabled 60 seconds when enabled	page 648

TABLE 118 VSRP parameters (Continued)

Parameter	Description	Default	For more information
Hold-down interval	The amount of time a Backup that has sent a Hello packet announcing its intent to become Master waits before beginning to forward traffic for the VRID. The hold-down interval prevents Layer 2 loops from occurring during VSRP rapid failover. The interval can from 1 – 84 seconds.	3 seconds	page 649
Track priority	A VSRP priority value assigned to the tracked ports. If a tracked port link goes down, the VRID port VSRP priority is reduced by the amount of the tracked port priority.	5	page 649
Track port	A track port is a port or virtual routing interface that is outside the VRID but whose link state is tracked by the VRID. Typically, the tracked interface represents the other side of VRID traffic flow through the device. If the link for a tracked interface goes down, the VSRP priority of the VRID interface is changed, causing the devices to renegotiate for Master.	None	page 650
Backup preempt mode	Prevents a Backup with a higher VSRP priority from taking control of the VRID from another Backup that has a lower priority but has already assumed control of the VRID.	Enabled	page 650
VRID active state	The active state of the VSRP VRID.	Disabled	page 642
RIP parameters			
Suppression of RIP advertisements	A Layer 3 Switch that is running RIP normally advertises routes to a backed up VRID even when the Layer 3 Switch is not currently the active Layer 3 Switch for the VRID. Suppression of these advertisements helps ensure that other Layer 3 Switches do not receive invalid route paths for the VRID. NOTE: This parameter is valid only on Layer 3 Switches.	Disabled (routes are advertised)	page 650

Configuring basic VSRP parameters

To configure VSRP, perform the following required tasks:

- Configure a port-based VLAN containing the ports for which you want to provide VSRP service.

NOTE

If you already have a port-based VLAN but only want to use VSRP on a sub-set of the VLANs ports, you can selectively remove ports from VSRP service in the VLAN. Refer to [“Removing a port from the VRID VLAN”](#) on page 646.

- Configure a VRID:
 - Specify that the device is a backup. Since VSRP, like VRRP-E, does not have an “owner”, all VSRP devices are backups. The active device for a VRID is elected based on the VRID priority, which is configurable.

- Activate the VRID.

The following example shows a simple VSRP configuration.

```
Brocade(config)#vlan 200
Brocade(config-vlan-200)#tag ethernet 1/1 to 1/8
Brocade(config-vlan-200)#vsrp vrid 1
Brocade(config-vlan-200-vrid-1)#backup
Brocade(config-vlan-200-vrid-1)#activate
```

Syntax: [no] vsrp vrid <num>

The <num> parameter specifies the VRID and can be from 1 – 255.

Syntax: [no] backup [priority <value>] [track-priority <value>]

This command is required. In VSRP, all devices on which a VRID are configured are Backups. The Master is then elected based on the VSRP priority of each device. There is no “owner” device as there is in VRRP.

For information about the command optional parameters, refer to the following:

- [“Changing the backup priority”](#) on page 646
- [“Changing the default track priority setting”](#) on page 649

Syntax: [no] activate

or

Syntax: enable | disable

Configuring optional VSRP parameters

The following sections describe how to configure optional VSRP parameters.

Disabling or re-enabling VSRP

VSRP is enabled by default on Layer 2 Switches and Layer 3 Switches. On a Layer 3 Switch, if you want to use VRRP or VRRP-E for Layer 3 redundancy instead of VSRP, you need to disable VSRP first. To do so, enter the following command at the global CONFIG level.

```
Brocade(config)#no router vsrp
router vsrp is disabled. All vsrp config data will be lost when writing to flash
```

To re-enable the protocol, enter the following command.

```
Brocade(config)#router vsrp
```

Syntax: [no] router vsrp

Since VRRP and VRRP-E do not apply to Layer 2 Switches, there is no need to disable VSRP and there is no command to do so. The protocol is always enabled.

Changing the timer scale

To achieve sub-second failover times, you can shorten the duration of all scale timers for VSRP, VRRP, and VRRP-E by adjusting the timer scale. The **timer scale** is a value used by the software to calculate the timers. By default, the scale value is 1. If you increase the timer scale, each timer value is divided by the scale value. Using the timer scale to adjust timer values enables you to easily change all the timers while preserving the ratios among their values. Here is an example.

Timer	Timer scale	Timer value
Hello interval	1	1 second
	2	0.5 seconds
Dead interval	1	3 seconds
	2	1.5 seconds
Backup Hello interval	1	60 seconds
	2	30 seconds
Hold-down interval	1	3 seconds
	2	1.5 second

NOTE

The Backups always use the value of the timer scale received from the Master, and the value from the Master will be written in the configuration file.

To change the timer scale, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# scale-timer 2
```

This command changes the scale to 2. All VSRP, VRRP, and VRRP-E timer values will be divided by 2.

Syntax: [no] **scale-timer** <num>

The <num> parameter specifies the multiplier. You can specify a timer scale from 1 – 10.

Configuring authentication

If the interfaces on which you configure the VRID use authentication, the VSRP packets on those interfaces also must use the same authentication. VSRP supports the following authentication types:

- **No authentication** – The interfaces do not use authentication.
- **Simple** – The interfaces use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the VRID configured on the interfaces must use the same authentication type and the same password.

To configure a simple password, enter a command such as the following at the VLAN configuration level.

```
Brocade(config-vlan-10)#vsrp auth-type simple-text-auth ourpword
```

This command configures the simple text password “ourpword”.

Syntax: `[no] vsrp auth-type no-auth | simple-text-auth <auth-data>`

The **auth-type no-auth** parameter indicates that the VRID and the interface it is configured on do not use authentication.

The **auth-type simple-text-auth <auth-data>** parameter indicates that the VRID and the interface it is configured on use a simple text password for authentication. The **<auth-data>** value is the password, and can be up to eight characters. If you use this parameter, make sure all interfaces on all the devices supporting this VRID are configured for simple password authentication and use the same password.

Configuring security features on a VSRP-aware device

This section shows how to configure security features on a VSRP-aware device. For an overview of this feature, refer to [“VSRP-aware security features”](#) on page 639.

Specifying an authentication string for VSRP hello packets

The following configuration defines **pri-key** as the authentication string for accepting incoming VSRP hello packets. In this example, the VSRP-aware device will accept all incoming packets that have this authorization string.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#vsrp-aware vrid 3 simple-text-auth pri-key
```

Syntax: `vsrp-aware vrid <vrid number> simple text auth <string>`

Specifying no authentication for VSRP hello packets

The following configuration specifies no authentication as the preferred VSRP-aware security method. In this case, the VSRP device will not accept incoming packets that have authentication strings.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#vsrp-aware vrid 2 no-auth
```

Syntax: `vsrp-aware vrid <vrid number> no-auth`

The following configuration specifies no authentication for VSRP hello packets received on ports 1/1, 1/2, 1/3, and 1/4 in VRID 4. For these ports, the VSRP device will not accept incoming packets that have authentication strings.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#vsrp-aware vrid 4 no-auth port-list ethe 1/1 to 1/4
```

Syntax: `vsrp-aware vrid <vrid number> no-auth port-list <port range>`

<vrid number> is a valid VRID (from 1 to 255).

no-auth specifies no authentication as the preferred VSRP-aware security method. The VSRP device will not accept incoming packets that have authentication strings.

simple-text-auth <string> specifies the authentication string for accepting VSRP hello packets, where **<string>** can be up to 8 characters.

port-list <port range> specifies the range of ports to include in the configuration.

Removing a port from the VRID VLAN

By default, all the ports on which you configure a VRID are interfaces for the VRID. You can remove a port from the VRID while allowing it to remain in the VLAN.

Removing a port is useful in the following cases:

- There is no risk of a loop occurring, such as when the port is attached directly to an end host.
- You plan to use a port in an MRP ring.

To remove a port from a VRID, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#no include-port ethernet 1/2
```

Syntax: [no] include-port ethernet [<slotnum>]/<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies the port you are removing from the VRID. The port remains in the VLAN but its forwarding state is not controlled by VSRP. If you are configuring a chassis device, specify the slot number as well as the port number (<slotnum>/<portnum>).

Configuring a VRID IP address

If you are configuring a Layer 3 Switch for VSRP, you can specify an IP address to back up. When you specify an IP address, VSRP provides redundancy for the address. This is useful if you want to back up the gateway address used by hosts attached to the VSRP Backups.

VSRP does not require you to specify an IP address. If you do not specify an address, VSRP provides Layer 2 redundancy. If you do specify an address, VSRP provides Layer 2 and Layer 3 redundancy.

The Layer 3 redundancy support is the same as VRRP-E support. For information, refer to [Chapter 39, “VRRP and VRRP-E”](#).

NOTE

The VRID IP address must be in the same subnet as a real IP address configured on the VSRP interface, but cannot be the same as a real IP address configured on the interface.

NOTE

Failover applies to both Layer 2 and Layer 3.

To specify an IP address to back up, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#ip-address 10.10.10.1
```

Syntax: [no] ip-address <ip-addr>

Changing the backup priority

When you enter the backup command to configure the device as a VSRP Backup for the VRID, you also can change the backup priority and the track priority:

- The backup priority is used for election of the Master. The VSRP Backup with the highest priority value for the VRID is elected as the Master for that VRID. The default priority is 100. If two or more Backups are tied with the highest priority, the Backup with the highest IP address becomes the Master for the VRID.
- The track priority is used with the track port feature. Refer to [“VSRP priority calculation”](#) on page 636 and [“Changing the default track priority setting”](#) on page 649.

To change the backup priority, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#backup priority 75
```

Syntax: [no] backup [priority <value>] [track-priority <value>]

The **priority <value>** parameter specifies the backup priority for this interface and VRID. Specify a value as follows:

- For VRRP, specify a value from 3 – 254. The default is 100.
- For VSRP and VRRP-E, specify a value from 6 – 255. The default is 100.

For a description of the **track-priority <value>** parameter, refer to [“Changing the default track priority setting”](#) on page 649.

Saving the timer values received from the master

The Hello messages sent by a VRID master contain the VRID values for the following VSRP timers:

- Hello interval
- Dead interval
- Backup Hello interval
- Hold-down interval

The Backups always use the value of the timers received from the Master.

To configure a Backup to save the VSRP timer values received from the Master instead of the timer values configured on the Backup, enter the following command. Saving the current timer values instead of the configured ones helps ensure consistent timer usage for all the VRID devices.

```
Brocade(config-vlan-200-vrid-1)#save-current-values
```

Syntax: [no] save-current-values

Changing the TTL setting

A VSRP Hello packet time to live (TTL) specifies how many hops the packet can traverse before being dropped. A hop can be a Layer 3 Switch or a Layer 2 Switch. You can specify from 1 – 255. The default TTL is 2. When a VSRP device (Master or Backup) sends a VSRP Hello packet, the device subtracts one from the TTL. Thus, if the TTL is 2, the device that originates the Hello packet sends it out with a TTL of 1. Each subsequent device that receives the packet also subtracts one from the packet TTL. When the packet has a TTL of 1, the receiving device subtracts 1 and then drops the packet because the TTL is zero.

NOTE

An MRP ring is considered to be a single hop, regardless of the number of nodes in the ring.

To change the TTL for a VRID, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#initial-ttl 5
```

Syntax: [no] initial-ttl <num>

The <num> parameter specifies the TTL and can be from 1 – 255. The default TTL is 2.

Changing the hello interval setting

The Master periodically sends Hello messages to the Backups. To change the Hello interval, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#hello-interval 10
```

Syntax: [no] hello-interval <num>

The <num> parameter specifies the interval and can be from 1 – 84 seconds. The default is 1 second.

NOTE

The default Dead interval is three times the Hello interval plus one-half second. Generally, if you change the Hello interval, you also should change the Dead interval on the Backups.

NOTE

If you change the timer scale, the change affects the actual number of seconds.

Changing the dead interval setting

The Dead interval is the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is dead. The default is 3 seconds. This is three times the default Hello interval.

To change the Dead interval, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#dead-interval 30
```

Syntax: [no] dead-interval <num>

The <num> parameter specifies the interval and can be from 1 – 84 seconds. The default is 3 seconds.

NOTE

If you change the timer scale, the change affects the actual number of seconds.

Changing the backup hello state and interval setting

By default, Backups do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#advertise backup
```

Syntax: [no] advertise backup

When a Backup is enabled to send Hello messages, the Backup sends a Hello message to the Master every 60 seconds by default. You can change the interval to be up to 3600 seconds.

To change the Backup Hello interval, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#backup-hello-interval 180
```

Syntax: [no] **backup-hello-interval** <num>

The <num> parameter specifies the message interval and can be from 60 – 3600 seconds. The default is 60 seconds.

NOTE

If you change the timer scale, the change affects the actual number of seconds.

Changing the hold-down interval setting

The hold-down interval prevents Layer 2 loops from occurring during failover, by delaying the new Master from forwarding traffic long enough to ensure that the failed Master is really unavailable.

To change the Hold-down interval, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#hold-down-interval 4
```

Syntax: [no] **hold-down-interval** <num>

The <num> parameter specifies the hold-down interval and can be from 1 – 84 seconds. The default is 3 seconds.

NOTE

If you change the timer scale, the change affects the actual number of seconds.

Changing the default track priority setting

When you configure a VRID to track the link state of other interfaces, if one of the tracked interface goes down, the software changes the VSRP priority of the VRID interface.

The software reduces the VRID priority by the amount of the priority of the tracked interface that went down. For example, if the VSRP interface priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VSRP interface priority to 40. If another tracked interface goes down, the software reduces the VRID priority again, by the amount of the tracked interface track priority.

The default track priority for all track ports is 5. You can change the default track priority or override the default for an individual track port.

- To change the default track priority, use the **backup priority <value> track-priority <value>** command, described below.
- To override the default track priority for a specific track port, use the **track-port** command. Refer to [“Specifying a track port setting”](#) on page 650.

To change the track priority, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#backup priority 100 track-priority 2
```

Syntax: [no] backup [priority <value>] [track-priority <value>]

Specifying a track port setting

You can configure the VRID on one interface to track the link state of another interface on the device. This capability is useful for tracking the state of the exit interface for the path for which the VRID is providing redundancy. Refer to “[VSRP priority calculation](#)” on page 636.

To configure a VRID to track an interface, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#track-port ethernet 2/4
```

Syntax: [no] track-port ethernet [<slotnum>/]<portnum> | ve <num> [priority <num>]

The **priority <num>** parameter changes the VSRP priority of the interface. If this interface goes down, the VRID VSRP priority is reduced by the amount of the track port priority you specify here.

NOTE

The priority <num> option changes the priority of the specified interface, overriding the default track port priority. To change the default track port priority, use the **backup track-priority <num>** command.

Disabling or re-enabling backup pre-emption setting

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the VRID. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, because the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

To disable preemption on a Backup, enter a command such as the following at the configuration level for the VRID.

```
Brocade(config-vlan-200-vrid-1)#non-preempt-mode
```

Syntax: [no] non-preempt-mode

Suppressing RIP advertisement from backups

Normally, for Layer 3 a VSRP Backup includes route information for a backed up IP address in RIP advertisements. As a result, other Layer 3 Switches receive multiple paths for the backed up interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master.

You can prevent the Backups from advertising route information for the backed up interface by enabling suppression of the advertisements.

NOTE

This parameter applies only if you specified an IP address to back up and is valid only on Layer 3 Switches.

To suppress RIP advertisements, enter the following commands.

```
Brocade(config)#router rip
Brocade(config-rip-router)#use-vrrp-path
```

Syntax: [no] use-vrrp-path

VSRP-aware interoperability

The **vsrp-aware tc-vlan-flush** command should be used in network configurations in which the Brocade switch operates as the VSRP-Aware device connecting to a FastIron platforms configured as a VSRP Master.

The command is available at the VLAN level, and is issued per a specific VRID, as shown here for VRID 11.

```
Brocade(config-vlan-10)#vsrp-aware vrid 11 tc-vlan-flush
```

Syntax: vsrp-aware vrid <num> tc-vlan-flush

When this command is enabled, MAC addresses will be flushed at the VLAN level, instead of at the port level. MAC addresses will be flushed for every topology change (TC) received on the VSRP-aware ports.

When you configure the **vsrp-aware tc-vlan-flush** command on a VSRP-aware device, and the device receives VSRP hello packets from the VSRP master, VSRP authentication is automatically configured. However, if the VSRP-aware device does not receive VSRP hello packets from the VSRP master when the **vsrp-aware tc-vlan-flush** command is configured, you must manually configure VSRP authentication. For more information on configuring VSRP authentication, refer to [“Configuring authentication”](#) on page 644.

When this command is enabled, the results of the **show vsrp-aware vlan** command resemble the following.

```
Brocade(config-vlan-10)#vsrp-aware vrid 11 tc-vlan-flush
Brocade(config-vlan-10)#show vsrp aware vlan 10
Aware Port Listing
  VLAN ID VRID Last Port Auth Type      Mac-Flush Age
    10      11 N/A   no-auth Configured Enabled  00:00:00.0
```

Displaying VSRP information

You can display the following VSRP information:

- Configuration information and current parameter values for a VRID or VLAN
- The interfaces on a VSRP-aware device that are active for the VRID

Displaying VRID information

To display VSRP information, enter the following command.

```
Brocade#show vsrp vrid 1
Total number of VSRP routers defined: 2
VLAN 200
auth-type no authentication
VRID 1
  State      Administrative-status  Advertise-backup  Preempt-mode  save-current
  standby    enabled                  disabled          true          false

  Parameter      Configured Current      Unit
  priority        100      80      (100-0)*(4.0/5.0)
  hello-interval    1        1      sec/1
  dead-interval     3        3      sec/1
  hold-interval     3        3      sec/1
  initial-ttl       2        2      hops

  next hello sent in 00:00:00.8
  Member ports:      ethe 1/1 to 1/5
  Operational ports: ethe 1/1 to 1/4
  Forwarding ports:  ethe 1/1 to 1/4
```

Syntax: `show vsrp [vrid <num> | vlan <vlan-id>]`

This display shows the following information when you use the **vrid <num>** or **vlan <vlan-id>** parameter. For information about the display when you use the **aware** parameter, refer to [“Displaying the active interfaces for a VRID”](#) on page 654.

TABLE 119 CLI display of VSRP VRID or VLAN information

Field	Description
Total number of VSRP routers defined	The total number of VRIDs configured on this device.
VLAN	The VLAN on which VSRP is configured.
auth-type	The authentication type in effect on the ports in the VSRP VLAN.
VRID parameters	
VRID	The VRID for which the following information is displayed.
state	<p>This device VSRP state for the VRID. The state can be one of the following:</p> <ul style="list-style-type: none"> initialize – The VRID is not enabled (activated). If the state remains “initialize” after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. <p>NOTE: If the state is “initialize” and the mode is incomplete, make sure you have specified the IP address for the VRID.</p> <ul style="list-style-type: none"> standby – This device is a Backup for the VRID. master – This device is the Master for the VRID.
Administrative-status	<p>The administrative status of the VRID. The administrative status can be one of the following:</p> <ul style="list-style-type: none"> disabled – The VRID is configured on the interface but VSRP or VRRP-E has not been activated on the interface. enabled – VSRP has been activated on the interface.

TABLE 119 CLI display of VSRP VRID or VLAN information (Continued)

Field	Description
Advertise-backup	Whether the device is enabled to send VSRP Hello messages when it is a Backup. This field can have one of the following values: <ul style="list-style-type: none"> disabled – The device does not send Hello messages when it is a Backup. enabled – The device does send Hello messages when it is a Backup.
Preempt-mode	Whether the device can be pre-empted by a device with a higher VSRP priority after this device becomes the Master. This field can have one of the following values: <ul style="list-style-type: none"> disabled – The device cannot be pre-empted. enabled – The device can be pre-empted.
save-current	The source of VSRP timer values preferred when you save the configuration. This field can have one of the following values: <ul style="list-style-type: none"> false – The timer values configured on this device are saved. true – The timer values most recently received from the Master are saved instead of the locally configured values.
NOTE: For the following fields: <ul style="list-style-type: none"> Configured – indicates the parameter value configured on this device. Current – indicates the parameter value received from the Master. Unit – indicates the formula used for calculating the VSRP priority and the timer scales in effect for the VSRP timers. A timer true value is the value listed in the Configured or Current field divided by the scale value. 	
priority	The device preferability for becoming the Master for the VRID. During negotiation, the Backup with the highest priority becomes the Master. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.
hello-interval	The number of seconds between Hello messages from the Master to the Backups for a given VRID.
dead-interval	The configured value for the dead interval. The dead interval is the number of seconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. NOTE: If the value is 0, then you have not configured this parameter.
hold-interval	The number of seconds a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID. If the Backup receives a Hello message with a higher priority than its own before the hold-down interval expires, the Backup remains in the Backup state and does not become the new Master.
initial-ttl	The number of hops a Hello message can traverse after leaving the device before the Hello message is dropped. NOTE: An MRP ring counts as one hop, regardless of the number of nodes in the ring.
next hello sent in	The amount of time until the Master dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this Layer 3 Switch itself will become the Master. NOTE: This field applies only when this device is a Backup.
Member ports	The ports in the VRID.

TABLE 119 CLI display of VSRP VRID or VLAN information (Continued)

Field	Description
Operational ports	The member ports that are currently up.
Forwarding ports	The member ports that are currently in the Forwarding state. Ports that are forwarding on the Master are listed. Ports on the Standby, which are in the Blocking state, are not listed.

Displaying the active interfaces for a VRID

On a VSRP-aware device, you can display VLAN and port information for the connections to the VSRP devices (Master and Backups).

To display the active VRID interfaces, enter the following command on the VSRP-aware device.

```
Brocade#show vsrp aware
Aware port listing
VLAN ID  VRID  Last Port
100      1      3/2
200      2      4/1
```

Syntax: `show vsrp aware`

This display shows the following information when you use the **aware** parameter. For information about the display when you use the **vrid <num>** or **vlan <vlan-id>** parameter, refer to [“Displaying VRID information”](#) on page 652.

TABLE 120 CLI display of VSRP-aware information

Field	Description
VLAN ID	The VLAN that contains the VSRP-aware device connection with the VSRP Master and Backups.
VRID	The VRID.
Last Port	The most recent active port connection to the VRID. This is the port connected to the current Master. If a failover occurs, the VSRP-aware device changes the port to the port connected to the new Master. The VSRP-aware device uses this port to send and receive data through the backed up node.

VSRP fast start

VSRP fast start allows non-Brocade or non-VSRP aware devices that are connected to a Brocade device that is the VSRP Master to quickly switchover to the new Master when a VSRP failover occurs

This feature causes the port on a VSRP Master to restart when a VSRP failover occurs. When the port shuts down at the start of the restart, ports on the non-VSRP aware devices that are connected to the VSRP Master flush the MAC address they have learned for the VSRP master. After a specified time, the port on the previous VSRP Master (which now becomes the Backup) returns back online. Ports on the non-VSRP aware devices switch over to the new Master and learn its MAC address.

Configuring VSRP fast start

The VSRP fast start feature can be enabled on a VSRP-configured Brocade device, either on the VLAN to which the VRID of the VSRP-configured device belongs (globally) or on a port that belongs to the VRID.

To globally configure a VSRP-configured device to shut down its ports when a failover occurs, then restart after five seconds, enter the following command.

```
Brocade(config)#vlan 100
Brocade(config-vlan-100)#vsrp vrid 1
Brocade(config-vlan-100-vrid-1)#restart-ports 5
```

Syntax: [no] restart-ports <seconds>

This command shuts down all the ports that belong to the VLAN when a failover occurs. All the ports will have the specified VRID.

To configure a single port on a VSRP-configured device to shut down when a failover occurs, then restart after a period of time, enter the following command.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e1000-1/1)#restart-vsrp-port 5
```

Syntax: [no] restart-vsrp-port <seconds>

In both commands, the <seconds> parameter instructs the VSRP Master to shut down its port for the specified number of seconds before it starts back up. Enter a value between 1 – 120 seconds. The default is 1 second.

Displaying ports that Have the VSRP fast start feature enabled

The **show vsrp vrid** command shows the ports on which the VSRP fast start feature is enabled.

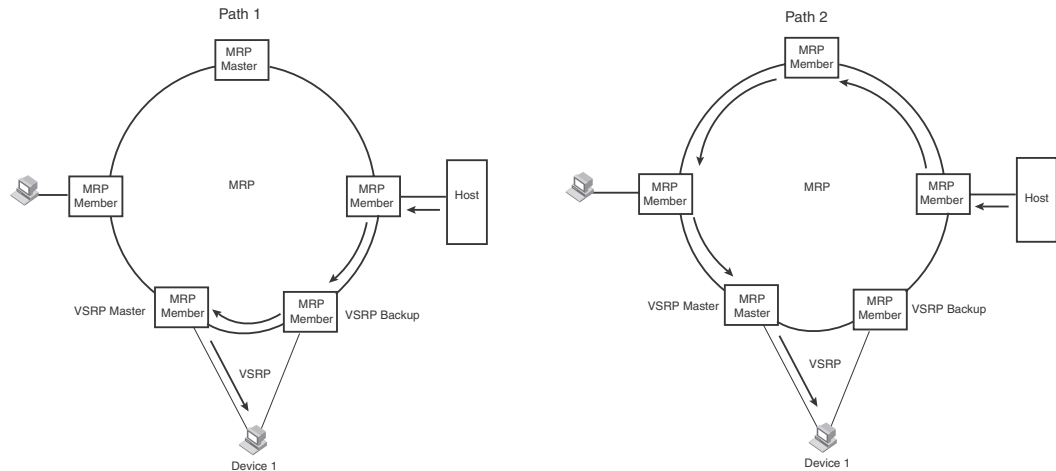
```
Brocade#show vsrp vrid 100
VLAN 100
  auth-type no authentication
  VRID 100
  =====
  State      Administrative-status  Advertise-backup  Preempt-mode  save-current
  master     enabled                  disabled          true          false
  Parameter  Configured Current      Unit/Formula
  priority   100          50          (100-0)*(2.0/4.0)
  hello-interval 1          1          sec/1
  dead-interval 3          3          sec/1
  hold-interval 3          3          sec/1
  initial-ttl 2          2          hops
  next hello sent in 00:00:00.3
  Member ports:   ethe 2/5 to 2/8
  Operational ports: ethe 2/5 ethe 2/8
  Forwarding ports: ethe 2/5 ethe 2/8
  Restart ports:   2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

The "Restart ports:" line lists the ports that have the VSRP fast start enabled, and the downtime for each port. Refer to [Table 119](#) on page 652 to interpret the remaining information on the display.

VSRP and MRP signaling

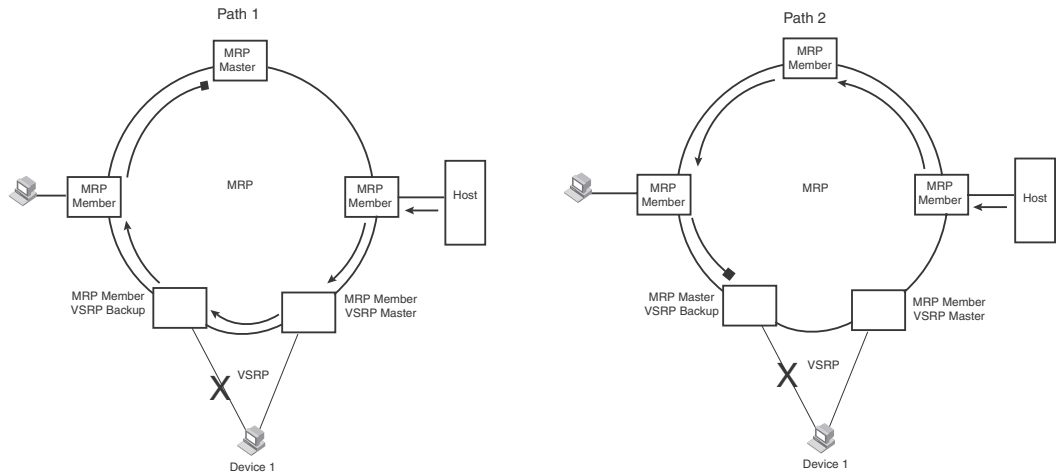
A device may connect to an MRP ring through VSRP to provide a redundant path between the device and the MRP ring. VSRP and MRP signaling ensures rapid failover by flushing MAC addresses appropriately. The host on the MRP ring learns the MAC addresses of all devices on the MRP ring and VSRP link. From these MAC addresses, the host creates a MAC database (table), which is used to establish a data path from the host to a VSRP-linked device. [Figure 56](#) below shows two possible data paths from the host to Device 1.

FIGURE 56 Two data paths from host on an MRP ring to a VSRP-linked device



If a VSRP failover from master to backup occurs, VSRP needs to inform MRP of the topology change; otherwise, data from the host continues along the obsolete learned path and never reach the VSRP-linked device, as shown in [Figure 57](#).

FIGURE 57 VSRP on MRP rings that failed over

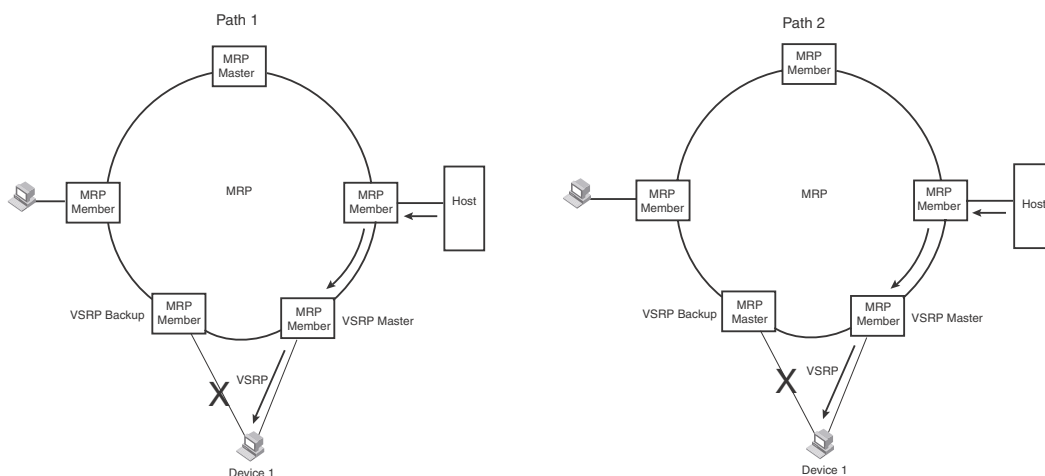


A signaling process for the interaction between VSRP and MRP ensures that MRP is informed of the topology change and achieves convergence rapidly. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all MRP instances impacted by the failover. Then each MRP instance does the following:

- The MRP node sends out an MRP PDU with the mac-flush flag set three times on the MRP ring.
- The MRP node that receives this MRP PDU empties all the MAC entries from its interfaces that participate on the MRP ring.
- The MRP node then forwards the MRP PDU with the mac-flush flag set to the next MRP node that is in forwarding state.

The process continues until the Master MRP node secondary (blocking) interface blocks the packet. Once the MAC address entries have been flushed, the MAC table can be rebuilt for the new path from the host to the VSRP-linked device ([Figure 58](#)).

FIGURE 58 New path established



There are no CLI commands used to configure this process.

VSRP

Power over Ethernet

Table 121 lists the individual Brocade FastIron switches and the Power over Ethernet (PoE) features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where noted.

TABLE 121 Supported PoE features

Feature	FESX FSX 800 FSX 1600 PoE interface modules ¹	FWS PoE models only	FCX PoE+ models only	ICX 6610 PoE models only	ICX 6430 ICX 6450 PoE models only
PoE+ (802.3at)	Yes (FSX 800 and FSX 1600 with SX-FI48GPP module only)	No	Yes	Yes	Yes
PoE (802.3af)	Yes	Yes	Yes	Yes	Yes
Detection of PoE power requirements advertised through CDP	Yes	Yes	Yes	Yes	Yes
Maximum power level for a PoE power consuming device	Yes	Yes	Yes	Yes	Yes
Power class for PoE power consuming device	Yes	Yes	Yes	Yes	Yes
Maximum power budget per PoE interface module	Yes	No	No	No	Yes
In-line power priority for a PoE port	Yes	Yes	Yes	Yes	Yes
PoE firmware upgrade via CLI	Yes	No	Yes	Yes	Yes

1. Supported on PoE-enabled interface modules installed in a chassis with PoE power supply.

Power over Ethernet overview

This section provides an overview of the requirements for delivering power over the LAN, as defined by the Institute of Electrical and Electronics Engineers Inc. (IEEE) in the 802.3af (PoE) and 802.3at (PoE+) specifications.

Brocade PoE devices provide Power over Ethernet, compliant with the standards described in the IEEE 802.3af specification for delivering inline power. Brocade PoE+ devices are compliant with both the 802.3af and 802.3at specifications. The 802.3af specification defined the original standard for delivering power over existing network cabling infrastructure, enabling multicast-enabled full streaming audio and video applications for converged services, such as, Voice over IP (VoIP), Wireless Local Area Access (WLAN) points, IP surveillance cameras, and other

IP technology devices. The 802.3at specification expands the standards to support higher power levels for more demanding powered devices, such as video IP phones, pan-tilt-zoom cameras and high-power outdoor antennas for wireless access points. Except where noted, this document will use the term PoE to refer to both PoE and PoE+.

[Table 121](#) lists the FastIron devices and modules that support PoE, PoE+, or both.

PoE technology eliminates the need for an electrical outlet and dedicated UPS near IP powered devices. With power sourcing equipment such as a Brocade FastIron PoE device, power is consolidated and centralized in the wiring closets, improving the reliability and resiliency of the network. Because PoE can provide Power over Ethernet cable, power is continuous, even in the event of a power failure.

Power over Ethernet terms used in this chapter

The following terms are introduced in this chapter:

- Power-sourcing device or Power-sourcing equipment (PSE) - This is the source of the power, or the device that integrates the power onto the network. Power sourcing devices and equipment have embedded PoE technology. The Brocade FastIron PoE device is a power sourcing device.
- IP powered device (PD) or power-consuming device - This is the Ethernet device that requires power and is situated on the end of the cable opposite the power sourcing equipment.

Methods for delivering Power over Ethernet

There are two methods for delivering Power over Ethernet (PoE), as defined in the 802.3af and 802.3at specifications:

- Endspan - Power is supplied through the Ethernet ports on a power sourcing device. With the Endspan solution, power can be carried over the two data pairs (Alternative A) or the two spare pairs (Alternative B).
- Midspan - Power is supplied by an intermediate power sourcing device placed between the switch and the PD. With the Midspan solution, power is carried over the two spare pairs (Alternative B).

With both methods, power is transferred over four conductors, between the two pairs. 802.3af- and 802.3at-compliant PDs are able to accept power from either set of pairs.

Brocade PoE devices use the Endspan method, compliant with the 802.3af and 802.3at standards.

The Endspan and Midspan methods are described in more detail in the following sections.

NOTE

All 802.3af- and 802.3at-compliant power consuming devices are required to support both application methods defined in the 802.3af and 802.3at specification.

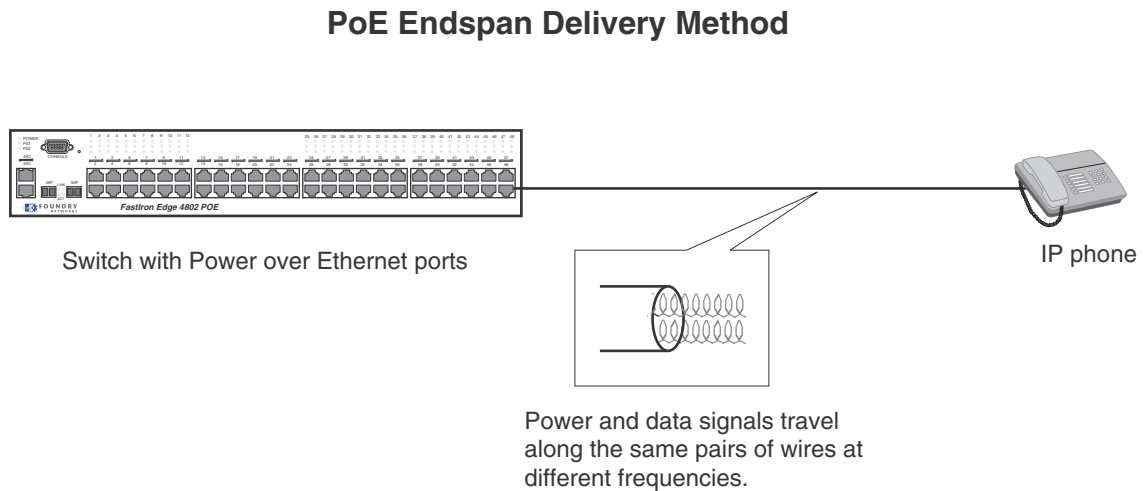
PoE endspan method

The PoE Endspan method uses the Ethernet switch ports on power sourcing equipment, such as a Brocade FastIron PoE switch, which has embedded PoE technology to deliver power over the network.

With the Endspan solution, there are two supported methods of delivering power. In Alternative A, four wires deliver data and power over the network. Specifically, power is carried over the live wire pairs that deliver data, as illustrated in [Figure 59](#). In Alternative B, the four wires of the spare pairs are used to deliver power over the network. Brocade PoE devices support Alternative A.

The Endspan method is illustrated in [Figure 59](#).

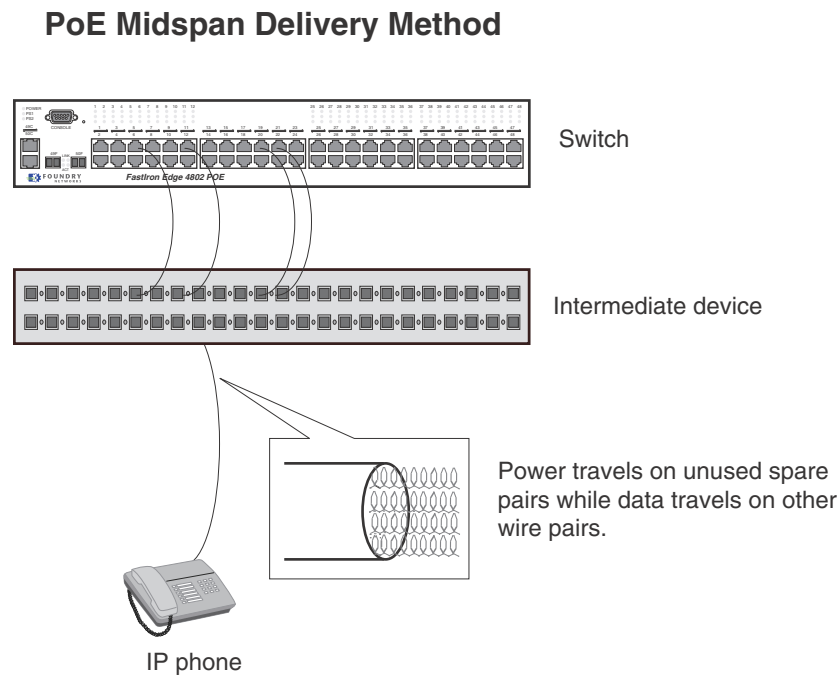
FIGURE 59 PoE Endspan delivery method



PoE midspan method

The PoE Midspan method uses an intermediate device, usually a PD, to inject power into the network. The intermediate device is positioned between the switch and the PD and delivers power over the network using the spare pairs of wires (Alternative B). The intermediate device has multiple channels (typically 6 to 24), and each of the channels has data input and a data-plus-power RJ-45 output connector.

The Midspan method is illustrated in [Figure 60](#).

FIGURE 60 PoE Midspan delivery method

PoE autodiscovery

PoE autodiscovery is a detection mechanism that identifies whether or not an installed device is 802.3af- or 802.3at-compatible. When you plug a device into an Ethernet port that is capable of providing inline power, the autodiscovery mechanism detects whether or not the device requires power and how much power is needed. The autodiscovery mechanism also has a disconnect protection mechanism that shuts down the power once a PD has been disconnected from the network or when a faulty PD has been detected. This feature enables safe installation and prevents high-voltage damage to equipment.

PoE autodiscovery is achieved by periodically transmitting current or test voltages that can detect when a PD is attached to the network. When an 802.3af- or 802.3at-compatible device is plugged into a PoE or PoE+ port, the PD reflects test voltage back to the power sourcing device (the Brocade device), ultimately causing the power to be switched on. Devices not compatible with 802.3af do not reflect test voltage back to the power sourcing device.

Power class

A power class determines the amount of power a PD receives from a PSE. When a valid PD is detected, the Brocade PoE device performs power classification by inducing a specific voltage and measuring the current consumption of the PD. Depending on the measured current, the appropriate class is assigned to the PD. PDs that do not support classification are assigned a class of 0 (zero). [Table 122](#) shows the different power classes and their respective power consumption needs.

TABLE 122 Power classes for PDs

Class	Usage	Power (watts) from Power Sourcing Device	
		Standard PoE	PoE+
0	default	15.4	30
1	optional	4	4
2	optional	7	7
3	optional	15.4	15.4
4	optional	15.4	30

Power specifications

The 802.3af (PoE) standard limits power to 15.4 watts (44 to 50 volts) from the power sourcing device, in compliance with safety standards and existing wiring limitations. Though limited by the 802.3af standard, 15.4 watts of power was ample for most PDs, which consumed an average of 5 to 12 watts of power (IP phones, wireless LAN access points, and network surveillance cameras each consume an average of 3.5 to 9 watts of power). The newer 802.3at (PoE+) standard nearly doubles the power, providing 30 watts (52 or 54 volts) from the power sourcing device.

The PoE power supply provides power to the PoE circuitry block, and ultimately to PoE power-consuming devices. The number of PoE power-consuming devices that one PoE power supply can support depends on the number of watts required by each power-consuming device. Each PoE power supply can provide either 1080 or 2380 watts of power, and each PoE port supports a maximum of either 15.4 or 30 watts of power per power-consuming device. For example, if each PoE power-consuming device attached to a FastIron PoE device consumes 10 watts of power, one 1080 watt power supply will power up to 108 PoE ports. You can install a second PoE power supply for additional PoE power. Power supply specifications are covered in the *Brocade FastIron X Series Chassis Hardware Installation Guide* and in the *Brocade FastIron CX Hardware Installation Guide*.

Dynamic upgrade of PoE power supplies

NOTE

This section applies to the SX 800 and SX 1600 chassis with PoE power supplies.

PoE+ requires higher power levels than standard PoE. In a chassis running software release 07.2.00 or higher, POE power supplies (SX-ACPWR-POE) are upgraded dynamically to 52 or 54 volts, depending on the maximum operating voltage the power supplies are capable of. The preferred voltage mode for PoE+ is 54 volts.

For safety reasons, all PoE power supplies installed in the chassis must operate at the same voltage mode, either 52 volts or 54 volts. The system will select the voltage mode of the power supply with the lowest supported voltage as the voltage mode for all PoE power supplies installed in the chassis. For example, in a FSX 800 chassis with one 52-volt capable PoE power supply and one 54-volt capable PoE power supply, both power supplies will be configured dynamically to operate at 52 volts.

PoE+ voltage selection occurs during each of the following events:

- When the device is powered ON or is rebooted

- When a PoE power supply is installed in the chassis
- When a PoE power supply is removed from the chassis

These events are described in detail in the following sections.

NOTE

A PoE power supply upgrade does not persist beyond a single power cycle. Therefore, an upgrade will occur automatically each time a power supply is re-inserted in the chassis.

You can use the **show inline power detail** command to display detailed information about the PoE power supplies installed in a FastIron PoE device. For more information refer to [“Displaying detailed information about PoE power supplies”](#) on page 678.



CAUTION

The SX-POE-AC-PWR power supply is designed exclusively for use with the Brocade FSX PoE devices. The power supply produces extensive power to support 802.3af and 802.3at applications. Installing the power supply in a device other than the Brocade FSX PoE device will cause extensive damage to your equipment.

Voltage selection during bootup

During bootup, the system will select the voltage mode (either 52 volts or 54 volts) of the power supply with the lowest supported voltage as the voltage mode for all PoE power supplies installed in the chassis. For example, if there is at least one power supply that supports 52 volts maximum, then all power supplies will be configured to operate at 52 volts, even if other supplies are 54 volts-capable. Once the operating voltage is applied, the system will display and log a warning message similar to the following:

```
Brocade(config)#  
Power supply 1 (from left when facing front side) detected.  
Power supply 1 (from left when facing front side) is up.  
WARNING: PoE power supplies in slots 1 are down rev. PoE/PoE+ function will work,  
but output power may be less than 50V under worst case load.
```

If all power supplies are 54 volts-capable, then all power supplies will be configured to operate at 54 volts. In this case, the system will not display or log a warning message.

Voltage selection when a PoE power supply is installed

When a PoE power supply is hotswapped into the chassis, the system will automatically adjust the voltage to match that of the PoE power supply or supplies that are currently installed in the chassis.

The following examples describe how the voltage is selected when a PoE power supply is installed:

- If a 54 volt-capable power supply is installed in a chassis that is operating with 52 volt-capable power supplies, the newly installed power supply will be set to operate at 52 volts.
- If a 54 volt-capable power supply is installed in a chassis that is operating with 54 volt-capable power supplies, the newly installed power supply will be set to operate at 54 volts.

- If a 52 volt-capable power supply is installed in a chassis that is operating with 54 volt-capable power supplies that are actively providing power, the system will reject the newly installed power supply since it cannot safely operate with the 54 volt-capable power supplies. In this case, the 52-volt power supply will be powered OFF and an error message similar to the following will display on the console.

```
Brocade(config)#
Power supply 1 (from left when facing front side) detected.
Power supply 1 (from left when facing front side) is up.
Shutting down power supply in slot 1 because it is not compatible with the
existing PoE power supplies. Please remove and replace.
```

When the system is next reloaded, the power supply voltage will be selected as described in the section [“Voltage selection during bootup”](#) on page 664.

- If a 52 volt-capable power supply is installed in a chassis that is operating with 54 volt-capable power supplies that are *not* actively providing power, the system will configure the power supplies to operate at 52 volts. In this case, the newly installed 52-volt power supply will *not* be powered OFF and a message similar to the following will display on the console.

NOTE: Automatically downgraded all PoE power supplies to 52V.

Voltage selection when a PoE power supply is removed

If a 52 volt PoE power supply is removed from the chassis, the system will survey the remaining power supplies to determine if they are 54 volts-capable. If the remaining supplies are 54 volts-capable and the system is not currently providing power to any PDs, then the software will upgrade the voltage of all supplies to 54 volts. The system will display and log a message similar to the following:

NOTE: Automatically upgraded all PoE power supplies to 54V.

However, if the system is currently providing power to one or more PDs, the system will not upgrade the voltage level. When the system is next reloaded, the power supply voltage will be selected as described in the section [“Voltage selection during bootup”](#) on page 664.

Power over Ethernet cabling requirements

The 802.3af and 802.3at standards currently support PoE and PoE+ on 10/100/1000-Mbps Ethernet ports operating over standard Category 5 unshielded twisted pair (UTP) cable or better. If your network uses cabling categories less than Category 5, you cannot implement PoE without first upgrading your cables to Category 5 UTP cable or better.

Supported powered devices

Brocade PoE devices support a wide range of IP powered devices including the following:

- Voice over IP (VoIP) phones
- Wireless LAN access points
- IP surveillance cameras

The following sections briefly describe these IP powered devices.

VoIP

Voice over IP (VoIP) is the convergence of traditional telephony networks with data networks, utilizing the existing data network infrastructure as the transport system for both services. Traditionally, voice is transported on a network that uses circuit-switching technology, whereas data networks are built on packet-switching technology. To achieve this convergence, technology has been developed to take a voice signal, which originates as an analog signal, and transport it within a digital medium. This is done by devices, such as VoIP telephones, which receive the originating tones and place them in UDP packets, the size and frequency of which is dependant on the coding / decoding (CODEC) technology that has been implemented in the VoIP telephone or device. The VoIP control packets use the TCP/IP format.

IP surveillance cameras

IP surveillance technology provides digital streaming of video over Ethernet, providing real-time, remote access to video feeds from cameras.

The main benefit of using IP surveillance cameras on the network is that you can view surveillance images from any computer on the network. If you have access to the Internet, you can securely connect from anywhere in the world to view a chosen facility or even a single camera from your surveillance system. By using a Virtual Private Network (VPN) or the company intranet, you can manage password-protected access to images from the surveillance system. Similar to secure payment over the Internet, images and information are kept secure and can be viewed only by approved personnel.

Installing PoE firmware

NOTE

The PoE firmware upgrade feature is not supported on FWS devices.

PoE firmware is stored in the PoE controller of the FastIron switch. You can install PoE firmware from the TFTP server on a FastIron switch with the CLI command. To do so, you should have a valid firmware image on the TFTP server.

NOTE

You can install PoE firmware only on one switch at a time. Therefore, to install PoE firmware on a stacking unit, you need to install it individually on every switch of the stack.

NOTE

The CLI syntax to install PoE firmware is different on FSX and FCX platforms.

FSX platform

To install PoE firmware on a FSX platform, enter a command such as the following.

```
Brocade#inline power install-firmware module 1 tftp 10.120.54.161
fsx_poe_07400.fw
```

Syntax: `inline power install-firmware module <slot> tftp <ip-address> <filename>`

Slot refers to the slot of the PoE module.

ip-address refers to the IP address of the TFTP server.

Filename refers to the name of the file, including the pathname.

FCX and ICX platforms

To install PoE firmware on FCX and ICX platforms, enter a command such as the following.

```
Brocade#inline power install-firmware stack-unit 1 tftp 10.120.54.161
fcx_poeplus_07400.fw
```

Syntax: `inline power install-firmware stack-unit <unit-number> tftp <ip-address> <filename>`

Stack-unit refers to the unit-id of the switch. If the switch is not a part of the stack, the unit number will be the default value. The default value for stack-unit is 1.

ip-address refers to the IP address of the tftp server.

Filename refers to the name of the file, including the pathname.

If you want to install firmware on a stack, you need to install firmware on one switch at a time with the above command.

Image file types

This section lists the PoE firmware file types supported and the procedure to install them on the FCX, ICX, and FSX devices.

TABLE 123 PoE Firmware files

Product	PoE Firmware ¹
FESX FSX 800 FSX 1600	fsx_poe_07400.fw
FSX 800 with SX-FI648PP module FSX 1600 with SX-FI648PP module	fsx_poeplus_07400.fw
FCX ICX 6610	fcx_poeplus_07400.fw
ICX 6430 ICX 6450	icx64XX_poeplus_07400.fw

1. The firmware files are specific to these devices only and are not interchangeable. For example, you cannot load FCX PoE firmware on a FSX device, and vice versa.

Installing PoE firmware

1. Place the PoE firmware on a TFTP server to which the Brocade device has access.
2. Copy the PoE firmware from the TFTP server into the switch. To do so, enter a command such as the following.

```
Brocade#inline power install-firmware module 1 tftp 10.120.54.161
fsx_poe_07400.fw
```

The process of PoE installation begins. You should see output similar to the following.

```
PoE Info: Loading firmware from TFTP file fsx_poe_07400.fw.....
```

```
PoE Info: Firmware in PoE module(s) in slot 1 will be installed now.
```

```
PoE Warning: Upgrading firmware in slot 1....DO NOT HOTSWAP OR POWER DOWN THE
MODULE.
```

```
PoE Info: FW Download on slot 1 module 1...(re)sending download command...
PoE Info: FW Download on slot 1 module 1...TPE response received.

PoE Info: FW Download on slot 1 module 1...(re)sending erase command...
PoE Info: FW Download on slot 1 module 1...erase command...accepted.
PoE Info: FW Download on slot 1 module 1...erasing firmware memory...
PoE Info: FW Download on slot 1 module 1...erasing firmware memory...completed

PoE Info: FW Download on slot 1 module 1...(re)sending program command...
PoE Info: FW Download on slot 1 module 1...sending program command...accepted.
PoE Info: FW Download on slot 1 module 1...programming firmware...takes ~ 10
minutes....

PoE Info: Firmware Download on slot 1.....10 percent completed.
PoE Info: Firmware Download on slot 1.....20 percent completed.
PoE Info: Firmware Download on slot 1.....30 percent completed.
PoE Info: Firmware Download on slot 1.....40 percent completed.
PoE Info: Firmware Download on slot 1.....50 percent completed.
PoE Info: Firmware Download on slot 1.....60 percent completed.
PoE Info: Firmware Download on slot 1.....70 percent completed.
PoE Info: Firmware Download on slot 1.....80 percent completed.
PoE Info: Firmware Download on slot 1.....90 percent completed.
PoE Info: Firmware Download on slot 1.....100 percent completed.
PoE Info: FW Download on slot 1 module 1...programming firmware...completed.
```

3. After downloading the firmware into the controller, the controller resets and reboot with the new PoE firmware, You should see output similar to the following.

```
PoE Info: FW Download on slot 1 module 1...upgrading firmware...completed.
Module will be reset.
PoE Info: Resetting module 1 in slot 1....
PoE Info: Resetting module 1 in slot 1....completed.

PoE Info: Programming Brocade defaults on module 1 in slot 1.....
PoE Info: Programming Brocade defaults. Step 1: Writing port defaults on
module 1 in slot 1....
PoE Info: Programming Brocade Defaults: Step 2: Writing PM defaults on module
1 in slot 1.
PoE Info: Programming Brocade defaults. Step 3: Writing user byte 0xf0 on
module 1 in slot 1.
PoE Info: Programming Brocade defaults. Step 4: Saving settings on module 1 in
slot 1.
PoE Info: Programming Brocade defaults on module 1 in slot 1.....completed.
```

NOTE

If you are attempting to transfer a file using TFTP but have received an error message, refer to [“Diagnostic error codes and remedies for TFTP transfers”](#) on page 93.

PoE and CPU utilization

Depending on the number of PoE-configured ports that have powered power devices, there may be a slight and noticeable increase of up to 15 percent in CPU utilization. In typical scenarios, this is normal behavior for PoE and does not affect the functionality of other features on the switch.

Enabling and disabling Power over Ethernet

To enable a port to receive inline power for power consuming devices, enter commands such as the following.

```
Brocade#configure terminal
Brocade(config)# interface ethernet 1/1
Brocade(config-if-e1000-1/1)# inline power
```

After entering the above commands, the console displays the following message.

```
Brocade(config-if-e1000-1/1)#PoE Info: Power enabled on port 1/1.
```

Syntax: [no] inline power

Use the **no** form of the command to disable the port from receiving inline power.

NOTE

Inline power should not be configured between two switches as it may cause unexpected behavior.

NOTE

FastIron PoE and PoE+ devices can automatically detect whether or not a power consuming device is 802.3af- or 802.3at-compliant.

Disabling support for PoE legacy power-consuming devices

Brocade PoE devices automatically support most legacy power consuming devices (devices not compliant with 802.3af 802.3at), as well as all 802.3af- and 802.3at-compliant devices. If desired, you can disable and re-enable support for legacy PoE power consuming devices on a global basis (on the entire device) or on individual slots (chassis devices only). When you disable legacy support, 802.3af- and 802.3at-compliant devices are not affected.

To disable support for legacy power consuming devices on a non-stackable device, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)# no legacy-inline-power
```

To disable support for legacy power consuming devices on a stackable device, enter the following command at the stack unit CONFIG level of the CLI.

```
Brocade(config-unit-2)# no legacy-inline-power
```

On chassis devices, you can disable support for legacy power consuming devices per slot. To disable legacy support on all ports in slot 2, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)# no legacy-inline-power 2
```

NOTE

The **no legacy-inline-power** command does not require a software reload if it is entered prior to connecting the PDs. If the command is entered after the PDs are connected, the configuration must be saved (**write memory**) and the software reloaded after the change is placed into effect.

Syntax: [no] legacy-inline-power [<slotnum>]

To re-enable support for legacy power consuming devices after it has been disabled, enter the **legacy-inline-power** command (without the **no** parameter).

The `<slotnum>` variable is required on chassis devices when disabling or re-enabling legacy support on a slot.

Use the **show run** command to view whether support for PoE legacy power consuming devices is enabled or disabled.

Enabling the detection of PoE power requirements advertised through CDP

Many power consuming devices, such as Cisco VoIP phones and other vendors' devices, use the Cisco Discovery Protocol (CDP) to advertise their power requirements to power sourcing devices, such as Brocade PoE devices. Brocade power sourcing equipment is compatible with Cisco and other vendors' power consuming devices; they can detect and process power requirements for these devices automatically.

NOTE

If you configure a port with a maximum power level or a power class for a power consuming device, the power level or power class will take precedence over the CDP power requirement. Therefore, if you want the device to adhere to the CDP power requirement, do not configure a power level or power class on the port.

Command syntax for PoE power requirements

To enable the Brocade device to detect CDP power requirements, enter the following commands.

```
Brocade# configure terminal
Brocade(config)# cdp run
```

Syntax: `[no] cdp run`

Use the **no** form of the command to disable the detection of CDP power requirements.

Setting the maximum power level for a PoE power-consuming device

When PoE is enabled on a port to which a power consuming device or PD is attached, by default, a Brocade PoE device will supply 15.4 watts of power at the RJ-45 jack, minus any power loss through the cables. A PoE+ device will supply either 15.4 or 30 watts of power (depending on the type of PD connected to the port), minus any power loss through the cables. For example, a PoE port with a default maximum power level of 15.4 watts will receive a maximum of 12.95 watts of power after 2.45 watts of power loss through the cable. This is compliant with the IEEE 802.3af and 802.3at specifications for delivering inline power. Devices that are configured to receive less PoE power, for example, 4.0 watts of power, will experience a lower rate of power loss through the cable.

If desired, you can manually configure the maximum amount of power that the Brocade PoE device will supply at the RJ-45 jack.

Setting power levels configuration note

Consider the following when enabling this feature:

- There are two ways to configure the power level for a PoE or PoE+ power consuming device. The first method is discussed in this section. The other method is provided in the section [“Setting the power class for a PoE power- consuming device”](#) on page 672. For each PoE port, you can configure either a maximum power level or a power class. You cannot configure both. You can, however, configure a maximum power level on one port and a power class on another port.
- The Brocade PoE or PoE+ device will adjust the power on a port only if there are available power resources. If power resources are not available, the following message will display on the console and in the Syslog:

```
PoE: Failed power allocation of 30000 mwatts on port 1/1/21. Will retry when  
more power budget.
```

Configuring power levels command syntax

To configure the maximum power level for a power consuming device, enter commands such as the following.

```
Brocade#configure terminal  
Brocade(config)# interface ethernet 1/1  
Brocade(config-if-e1000-1/1)# inline power power-limit 14000
```

These commands enable inline power on interface ethernet 1 in slot 1 and set the PoE power level to 14,000 milliwatts (14 watts).

Syntax: `inline power power-limit <power level>`

The `<power level>` variable is the maximum power level in number of milliwatts. The following values are supported:

- PoE – Enter a value from 1000 through 15,400. The default is 15,400.
- PoE+ – Enter a value from 1000 through 30,000. The default is 30,000.

NOTE

Do not configure a power level higher than 15,400 for standard PoE PDs, which support a maximum of 15,400 milliwatts. Setting the power level higher than 15,400 could damage the PD.

For information about resetting the maximum power level, refer to [“Resetting PoE parameters”](#) on page 675.

Setting the power class for a PoE power- consuming device

A power class specifies the maximum amount of power that a Brocade PoE or PoE+ device will supply to a power consuming device. [Table 124](#) shows the different power classes and their respective maximum power allocations.

TABLE 124 Power classes for PDs

Class	Usage	Power (watts) from Power Sourcing Device	
		Standard PoE	PoE+
0	default	15.4	30
1	optional	4	4
2	optional	7	7
3	optional	15.4	15.4
4	optional	15.4	30

Setting the power class configuration notes

Consider the following points when setting the power class for a PoE power-consuming device.

- The power class sets the maximum power level for a power consuming device. Alternatively, you can set the maximum power level as instructed in the section [“Setting the maximum power level for a PoE power- consuming device”](#) on page 670. For each PoE port, you can configure either a power class or a maximum power level. You cannot configure both. You can, however, configure a power level on one port and a power class on another port.
- The power class includes any power loss through the cables. For example, a PoE port with a power class of 3 (15.4 watts) will receive a maximum of 12.95 watts of power after 2.45 watts of power loss through the cable. This is compliant with the IEEE 802.3af and 802.3at specifications for delivering inline power. Devices that are configured to receive less PoE power, for example, class 1 devices (4.0 watts), will experience a lower rate of power loss through the cable.
- The Brocade PoE or PoE+ device will adjust the power on a port only if there are available power resources. If power resources are not available, the following message will display on the console and in the Syslog:

```
PoE: Failed power allocation of 30000 mwatts on port 1/1/21. Will retry when
more power budget.
```

Setting the power class command syntax

To configure the power class for a PoE power consuming device, enter commands such as the following.

```
Brocade# configure terminal
Brocade(config)# interface ethernet 1/1
Brocade(config-if-e1000-1/1)# inline power power-by-class 2
```

These commands enable inline power on interface ethernet 1 in slot 1 and set the power class to 2.

Syntax: `inline power power-by-class <class value>`

The *<class value>* variable is the power class. Enter a value between 0 and 4. The default is 0.

[Table 124](#) shows the different power classes and their respective maximum power allocations.

NOTE

Do not configure a class value of 4 on a PoE+ port on which a standard PoE PD is connected. Standard PoE PDs support a maximum of 15.4 watts. Setting the power class value to 4 (30 watts) could damage the PD.

For information about resetting the power class, refer to [“Resetting PoE parameters”](#) on page 675.

Setting the power budget for a PoE interface module

By default, each PoE and PoE+ interface module has a maximum power budget of 65535 watts. If desired, you can change the amount of power allocated to each PoE and PoE+ interface module installed in the chassis. To do so, enter a command such as the following.

```
Brocade(config)# inline power budget 150000 module 7
```

This command allocates 150000 milliwatts (150 watts) to the PoE interface module in slot 7. The command takes effect immediately. The results are displayed in the “power budget” column in the **show inline power detail** output. The configuration (`inline power budget 150000 module 7`) is displayed in the **show running-config** output.

Syntax: `inline power budget <num> module <slot>`

The *<num>* variable is the number of milliwatts to allocate to the module. Enter a value from 0 through 65535000.

The *<slot>* variable specifies the where the PoE or PoE+ module resides in the chassis.

Setting the inline power priority for a PoE port

Each PoE power supply can provide either 1080 or 2380 watts of power, and each PoE port receives a maximum of 15.4 watts of power per PoE power-consuming device, or a maximum of 30 watts of power per PoE+ power-consuming device, minus any power loss through the cable. The power capacity of one or two PoE power supplies is shared among all PoE power consuming devices attached to the FastIron PoE device.

In a configuration where PoE power consuming devices collectively have a greater demand for power than the PoE power supply or supplies can provide, the FastIron PoE device must place the PoE ports that it cannot power in *standby* or *denied* mode (waiting for power) until the available power increases. The available power increases when one or more PoE ports are powered down, or, if applicable, when an additional PoE power supply is installed in the FastIron PoE device.

When PoE ports are in *standby* or *denied* mode (waiting for power) and the FastIron PoE device receives additional power resources, by default, the device will allocate newly available power to the standby ports in priority order, with the highest priority ports first, followed by the next highest priority ports, and so on. Within a given priority, standby ports are considered in ascending order, by

slot number then by port number, provided enough power is available for the ports. For example, PoE port 1/11 should receive power before PoE port 2/1. However, if PoE port 1/11 needs 12 watts of power and PoE port 2/1 needs 10 watts of power, and 11 watts of power become available on the device, the FastIron PoE device will allocate the power to port 2/1 because it does not have sufficient power for port 1/11.

You can configure an *inline power priority* on PoE ports, whereby ports with a higher inline power priority will take precedence over ports with a low inline power priority. For example, if a new PoE port comes online and the port is configured with a high priority, if necessary (if power is already fully allocated to power consuming devices), the FastIron PoE device will remove power from a PoE port or ports that have a lower priority and allocate the power to the PoE port that has the higher value.

Ports that are configured with the same inline power priority are given precedence based on the slot number and port number in ascending order, provided enough power is available for the port. For example, if both PoE port 1/2 and PoE port 2/1 have a high inline power priority value, PoE port 1/2 will receive power before PoE port 2/1. However, if PoE port 1/2 needs 12 watts of power and PoE port 2/1 needs 10 watts of power, and 11 watts of power become available on the device, the FastIron PoE device will allocate the power to PoE port 2/1 because it does not have sufficient power for port 1/2. By default, all ports are configured with a low inline power priority.

Command syntax for setting the inline power priority for a PoE port

To configure an inline power priority for a PoE port on a FastIron PoE device, enter commands such as the following.

```
Brocade#configure terminal
Brocade(config)# interface ethernet 1/1
Brocade(config-if-e1000-1/1)# inline power priority 2
```

These commands enable inline power on interface ethernet 1 in slot 1 and set the inline power priority level to high.

Syntax: [no] **inline power priority** <priority num>

The **priority** <priority num> parameter is the inline power priority number. The default is 3 (low priority). You can specify one of the following values:

- 3 – Low priority
- 2 – High priority
- 1 – Critical priority

Use the **inline power** command (without a priority number) to reset a port priority to the default (low) priority.

Use the **no inline power** command to disable the port from receiving inline power.

For information about resetting the inline power priority, refer to [“Resetting PoE parameters”](#) on page 675.

To view the inline power priority for all PoE ports, issue the **show inline power** command at the Privileged EXEC level of the CLI. Refer to [“Displaying PoE operational status”](#) on page 675.

Resetting PoE parameters

NOTE

Resetting PoE parameters applies to the FastIron X Series PoE chassis.

You can override or reset PoE port parameters including power priority, power class, and maximum power level. To do so, you must specify each PoE parameter in the CLI command line. This section provides some CLI examples.

Example 1—Changing a PoE port power priority from high to low

To change a PoE port power priority from high to low (the default value) and keep the current maximum configured power level of 3000, enter commands such as the following.

```
Brocade# configure terminal
Brocade(config)# interface ethernet 1/1
Brocade(config-if-e1000-1/1)# inline power priority 3 power-limit 3000
```

You must specify both the inline power priority and the maximum power level (**power-limit** command), even though you are keeping the current configured maximum power level at 3000. If you do not specify the maximum power level, the device will apply the default value. Also, you must specify the inline power priority before specifying the power limit.

Example 2—Changing a port power class from 2 to 3

To change a port power class from 2 (7 watts maximum) to 3 (15.4 watts maximum) and keep the current configured power priority of 2, enter commands such as the following.

```
Brocade#configure terminal
Brocade(config)# interface ethernet 1/1
Brocade(config-if-e1000-1/1)# inline power priority 2 power-by-class 3
```

You must specify both the power class and the inline power priority, even though you are not changing the power priority. If you do not specify the power priority, the device will apply the default value of 3 (low priority). Also, you must specify the inline power priority before specifying the power class.

Displaying Power over Ethernet information

This section lists the CLI commands for viewing PoE information.

Displaying PoE operational status

The **show inline power** command displays operational information about Power over Ethernet.

You can view the PoE operational status for the entire device, for a specific PoE module only, or for a specific interface only. In addition, you can use the **show inline power detail** command to display in-depth information about PoE power supplies.

The following shows an example of the **show inline power** display output on a Brocade PoE device.

Displaying Power over Ethernet information

Brocade#show inline power

Power Capacity: Total is 2160000 mWatts. Current Free is 18800 mWatts.

Power Allocations: Requests Honored 769 times

... some lines omitted for brevity...

Port	Admin State	Oper State	---Power(mWatts)---		PD Type	PD Class	Pri	Fault/ Error
			Consumed	Allocated				
4/1	On	On	5070	9500	802.3af	n/a	3	n/a
4/2	On	On	1784	9500	Legacy	n/a	3	n/a
4/3	On	On	2347	9500	802.3af	n/a	3	n/a
4/4	On	On	2441	9500	Legacy	n/a	3	n/a
4/5	On	On	6667	9500	802.3af	Class 3	3	n/a
4/6	On	On	2723	9500	802.3af	Class 2	3	n/a
4/7	On	On	2347	9500	802.3af	n/a	3	n/a
4/8	On	On	2347	9500	802.3af	n/a	3	n/a
4/9	On	On	2347	9500	802.3af	n/a	3	n/a
4/10	On	On	4976	9500	802.3af	Class 3	3	n/a
4/11	On	On	4882	9500	802.3af	Class 3	3	n/a
4/12	On	On	4413	9500	802.3af	Class 1	3	n/a
4/13	On	On	7793	9500	802.3af	n/a	3	n/a
4/14	On	On	7512	9500	802.3af	n/a	3	n/a
4/15	On	On	8075	9500	802.3af	n/a	3	n/a
4/16	On	On	4131	9500	802.3af	Class 1	3	n/a
4/17	On	On	2347	9500	802.3af	n/a	3	n/a
4/18	On	Off	0	9500	n/a	n/a	3	n/a
4/19	On	On	5352	9500	Legacy	n/a	3	n/a
4/20	On	On	7981	9500	802.3af	n/a	3	n/a
4/21	On	On	12958	13000	802.3af	Class 3	3	n/a
4/22	On	On	12958	13000	802.3af	Class 3	3	n/a
4/23	On	On	13052	13000	802.3af	Class 3	3	n/a
4/24	On	On	12864	13000	802.3af	Class 3	3	n/a
Total			137367	242000				

... some lines omitted for brevity...

Grand Total 1846673 2127400

Syntax: show inline power [<port>]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Table 125 provides definitions for the **show inline power** command.

TABLE 125 Field definitions for the **show inline power** command

Column	Definition
Power Capacity	The total PoE power supply capacity and the amount of available power (current free) for PoE power consuming devices. Both values are shown in milliwatts.
Power Allocations	The number of times the FSX fulfilled PoE requests for power.
Port	The slot number and port number.
Admin State	Specifies whether or not Power over Ethernet has been enabled on the port. This value can be one of the following: <ul style="list-style-type: none"> On – The inline power command was issued on the port. Off – The inline power command has not been issued on the port.
Oper State	Shows the status of inline power on the port. This value can be one of the following: <ul style="list-style-type: none"> On – The PoE power supply is delivering inline power to the PD. Off – The PoE power supply is not delivering inline power to the PD. Denied – The port is in standby mode (waiting for power) because the FSX does not currently have enough available power for the port. <p>NOTE: When you enable a port using the CLI, it may take 12 or more seconds before the operational state of that port is displayed correctly in the show inline power output.</p>
Power Consumed	The number of current, actual milliwatts that the PD is consuming.
Power Allocated	The number of milliwatts allocated to the port. This value is either the default or configured maximum power level, or the power class that was automatically detected by the device.
PD Type	The type of PD connected to the port. This value can be one of the following: <ul style="list-style-type: none"> 802.3at – The PD connected to this port is 802.3at-compliant. 802.3af – The PD connected to this port is 802.3af-compliant. Legacy – The PD connected to this port is a legacy product (not 802.3af-compliant). N/A – Power over Ethernet is configured on this port, and one of the following is true: <ul style="list-style-type: none"> The device connected to this port is a non-powered device. No device is connected to this port. The port is in <i>standby</i> or <i>denied</i> mode (waiting for power).
PD Class	Determines the maximum amount of power a PD receives. Table 124 shows the different power classes and their respective maximum power allocations. This field can also be “Unknown”, meaning the device attached to the port cannot advertise its power class. <p>NOTE: If an 802.3at PD with a class 4 value is connected, the Brocade FastIron switch will not be able to do the power negotiation since these switches cannot handle the 802.3at LLDP.</p>
Pri	The port <i>in-line power priority</i> , which determines the order in which the port will receive power while in <i>standby</i> mode (waiting for power). Ports with a higher priority will receive power before ports with a low priority. This value can be one of the following: <ul style="list-style-type: none"> 3 – Low priority 2 – High priority 1 – Critical priority

TABLE 125 Field definitions for the show inline power command (Continued)

Column	Definition
Fault/Error	<p>If applicable, this is the fault or error that occurred on the port. This value can be one of the following:</p> <ul style="list-style-type: none"> critical temperature – The PoE chip temperature limit rose above the safe operating level, thereby powering down the port. detection failed - discharged capacitor – The port failed capacitor detection (legacy PD detection) because of a discharged capacitor. This can occur when connecting a non-PD on the port. detection failed - out of range capacitor – The port failed capacitor detection (legacy PD detection) because of an out-of-range capacitor value. This can occur when connecting a non-PD on the port. internal h/w fault – A hardware problem has hindered port operation. lack of power – The port has shut down due to lack of power. main supply voltage high – The voltage was higher than the maximum voltage limit, thereby tripping the port. main supply voltage low – The voltage was lower than the minimum voltage limit, thereby tripping the port. overload state – The PD consumes more power than the maximum limit configured on the port, based on the default configuration, user configuration, or CDP configuration. over temperature – The port temperature rose above the temperature limit, thereby powering down the port. PD DC fault – A succession of underload and overload states, or a PD DC/DC fault, caused the port to shutdown. short circuit – A short circuit was detected on the port delivering power. underload state – The PD consumes less power than the minimum limit specified in the 802.3af standard. voltage applied from ext src – The port failed capacitor detection (legacy PD detection) because the voltage applied to the port was from an external source.
Total	The total power in milliwatts being <i>consumed</i> by all PDs connected to the Interface module, and the total power in milliwatts <i>allocated</i> to all PDs connected to the Interface module.
Grand Total	The total number of current, actual milliwatts being <i>consumed</i> by all PDs connected to the FastIron PoE device, and the total number of milliwatts <i>allocated</i> to all PDs connected to the FastIron PoE device.

Displaying detailed information about PoE power supplies

The **show inline power detail** command displays detailed operational information about the PoE power supplies in Brocade PoE switches. The command output differs on FCX POE+ switches compared to FastIron X Series switches.

The following is an example of the **show inline power detail** command output on an FCX POE+ switch.

```

Brocade#FCX#show inline power detail

Power Supply Data On stack 1:
+++++++
Power Supply #1:
    Max Curr:      7.5 Amps
    Voltage:       54.0 Volts
    Capacity:      410 Watts

POE Details Info. On Stack 1 :

General PoE Data:
+++++++

Firmware
Version
-----
02.1.0

Cumulative Port State Data:
+++++++

#Ports    #Ports    #Ports    #Ports    #Ports    #Ports    #Ports
Admin-On  Admin-Off  Oper-On   Oper-Off  Off-Denied Off-No-PD  Off-Fault
-----
45         3         0         48        0         45         0

Cumulative Port Power Data:
+++++++

#Ports  #Ports  #Ports      Power      Power
Pri: 1  Pri: 2  Pri: 3  Consumption  Allocation
-----
0        0       45       0.0 W       0.0 W

Power Supply Data On stack 2:
+++++++

Power Supply Data:
+++++++

Power Supply #1:
    Max Curr:      7.5 Amps
    Voltage:       54.0 Volts
    Capacity:      410 Watts

POE Details Info. On Stack 2 :

General PoE Data:
+++++++

Firmware
Version
-----
02.1.0

```

Displaying Power over Ethernet information

... continued from previous page...

Cumulative Port State Data:

+++++

#Ports Admin-On	#Ports Admin-Off	#Ports Oper-On	#Ports Oper-Off	#Ports Off-Denied	#Ports Off-No-PD	#Ports Off-Fault
20	4	0	24	0	20	0

Cumulative Port Power Data:

+++++

#Ports Pri: 1	#Ports Pri: 2	#Ports Pri: 3	Power Consumption	Power Allocation
20	0	0	0.0 W	0.0 W

Power Supply Data On stack 3:

+++++

Power Supply #1:

Max Curr: 7.5 Amps
Voltage: 54.0 Volts
Capacity: 410 Watts

POE Details Info. On Stack 3 :

General PoE Data:

+++++

Firmware

Version

02.1.0

Cumulative Port State Data:

+++++

#Ports Admin-On	#Ports Admin-Off	#Ports Oper-On	#Ports Oper-Off	#Ports Off-Denied	#Ports Off-No-PD	#Ports Off-Fault
22	2	0	24	0	22	0

Cumulative Port Power Data:

+++++

#Ports Pri: 1	#Ports Pri: 2	#Ports Pri: 3	Power Consumption	Power Allocation
0	10	12	0.0 W	0.0 W

The following is an example of the **show inline power detail** command output on a FastIron X Series PoE switch.

```
Brocade#show inline power detail
```

```
Power Supply Data:
```

```
+++++
```

```
PoE+ Max Operating Voltage: 54 V
```

```
Power Supply #1:
```

```

Model Number: 32004000
Serial Number: 093786124716
Firmware Ver: 1.6
Test Date: 9/12/09 (mm/dd/yy)
H/W Status: 807
Max Curr: 50.0 Amps
Voltage: 54.0 Volts
Capacity: 2500 Watts
PoE Capacity: 2260 Watts
Consumption: 2095 Watts
```

```
General PoE Data:
```

```
+++++
```

```
Slot  Firmware
      Version
```

```

-----
3      Device 1: 02.1.0 Device 2: 02.1.0
4      Device 1: 02.1.0 Device 2: 02.1.0
6      02.1.0
7      Device 1: 02.1.0 Device 2: 02.1.0
8      02.1.0
```

```
Cumulative Port State Data:
```

```
+++++
```

Slot	#Ports Admin-On	#Ports Admin-Off	#Ports Oper-On	#Ports Oper-Off	#Ports Off-Denied	#Ports Off-No-PD	#Ports Off-Fault
3	48	0	48	0	0	0	0
4	48	0	48	0	0	0	0
6	24	0	0	24	0	24	0
7	48	0	4	44	44	0	0
8	24	0	0	24	0	24	0
Total:	192	0	100	92	44	48	0

```
... continued on next page...
```

Displaying Power over Ethernet information

... continued from previous page...

Cumulative Port Power Data:

+++++

Slot	#Ports Pri: 1	#Ports Pri: 2	#Ports Pri: 3	Power Consumption	Power Allocation	Power Budget
3	0	0	48	513.90 W	739.200 W	65535.0 W
4	0	0	48	1346.497 W	1440.0 W	65535.0 W
6	0	0	24	0.0 W	0.0 W	65535.0 W
7	0	0	48	43.72 W	61.600 W	65535.0 W
8	0	0	24	0.0 W	0.0 W	65535.0 W
Total:0	0	0	192	1902.659 W	2240.800 W	327675.0 W

Syntax: show inline power detail

[Table 126](#) provides definitions for the statistics displayed in the **show inline power detail** command.

TABLE 126 Field definitions for the **show inline power detail** command

Column	Definition
Power supply data	
PoE+ Max Operating Voltage	This field is applicable to FastIron PoE+ chassis devices only. It displays the maximum operating voltage supported by the PoE power supply. Possible values are: <ul style="list-style-type: none"> 52 V 54 V
Model Number	The manufacturing part number of the PoE power supply. Possible values are: <ul style="list-style-type: none"> 32016000 32007000
Serial Number	The serial number of the PoE power supply, for example, AA100730213.
Firmware Ver	The PoE power supply firmware version.
Test Date	The PoE power supply firmware test date in the format mm/dd/yyyy.
H/W Status	The PoE power supply hardware status code. This field is used by Brocade Technical Support for troubleshooting.
Max Curr	The PoE power supply maximum current capacity.
Voltage	The PoE power supply current input voltage.
Capacity	The PoE power supply total power capacity (in watts).
PoE Capacity	The PoE power supply PoE power capacity (in watts).
Consumption	The total number of watts consumed by PoE power consuming devices and PoE modules in the system, plus any internal or cable power loss. <p>NOTE: Under the lower total inline power consumption level by Powered Devices (PDs) on FastIron SX devices, the power consumption displayed by the power supply units (PSUs) is inaccurately displayed as lower than the actual power consumption of the PSUs due to the sensitivity limitations of power supply measurements.</p>
General PoE data	
Slot	The Interface module / slot number.

TABLE 126 Field definitions for the **show inline power detail** command (Continued)

Column	Definition
Firmware Version	The Interface module / slot number firmware version.
Cumulative port state data	
NOTE: When you enable a port using the CLI, it may take 12 or more seconds before the operational state of that port is displayed correctly in the show inline power output.	
Slot	The Interface module / slot number.
#Ports Admin-On	The number of ports on the Interface module on which the inline power command was issued.
#Ports Admin-Off	The number of ports on the Interface module on which the inline power command was not issued.
#Ports Oper-On	The number of ports on the Interface module that are receiving inline power from the PoE power supply.
#Ports Oper-Off	The number of ports on the Interface module that are not receiving inline power from the PoE power supply.
#Ports Off-Denied	The number of ports on the Interface module that were denied power because of insufficient power.
#Ports Off-No-PD	The number of ports on the Interface module to which no PDs are connected.
#Ports Off-Fault	The number of ports on the Interface module that are not receiving power because of a subscription overload.
Total	The totals for all of the fields in the Cumulative Port State Data report.
Cumulative port power data	
Slot	The Interface module / slot number.
#Ports Pri: 1	The number of PoE ports on the Interface module that have a PoE port priority of 1.
#Ports Pri: 2	The number of PoE ports on the Interface module that have a PoE port priority of 2.
#Ports Pri: 3	The number of PoE ports on the Interface module that have a PoE port priority of 3.
Power Consumption	The total number of watts consumed by PoE power consuming devices, plus any cable loss.
Power Allocation	The number of watts allocated to the Interface module PoE ports. This value is the sum of the ports' default or configured maximum power levels, or power classes automatically detected by the FastIron PoE device.
Power Budget	The power budget allocated to the slot. The default value is 65535 watts. Any other value indicates that the power budget was configured using the CLI command inline power budget .
Total	The totals for all of the fields in the Cumulative Port Power Data report.

Displaying Power over Ethernet information

UDLD and Protected Link Groups

Table 127 lists the individual Brocade FastIron switches and the Uni-Directional Link Detection (UDLD) and protected link group features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 127 Supported UDLD and protected link group features

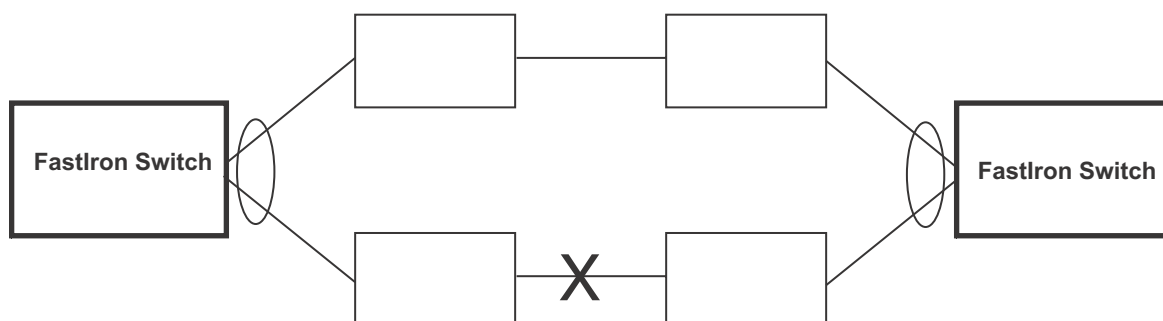
Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Uni-directional Link Detection (UDLD) (Link keepalive)	Yes	Yes	Yes	Yes	Yes
UDLD on tagged ports	Yes	Yes	Yes	Yes	Yes
Protected link groups	Yes	Yes	Yes	Yes	Yes

UDLD overview

Uni-Directional Link Detection (UDLD) monitors a link between two Brocade devices and brings the ports on both ends of the link down if the link goes down at any point between the two devices. This feature is useful for links that are individual ports and for trunk links. Figure 61 shows an example.

FIGURE 61 UDLD example

When link keepalive is enabled, the feature brings down the FastIron ports connected to the failed link.



Normally, a Brocade device load balances traffic across the ports in a trunk group. In this example, each Brocade device load balances traffic across two ports. Without the UDLD feature, a link failure on a link that is not directly attached to one of the Brocade devices is undetected by the Brocade devices. As a result, the Brocade devices continue to send traffic on the ports connected to the failed link.

When UDLD is enabled on the trunk ports on each Brocade device, the devices detect the failed link, disable the ports connected to the failed link, and use the remaining ports in the trunk group to forward the traffic.

Ports enabled for UDLD exchange proprietary health-check packets once every second (the keepalive interval). If a port does not receive a health-check packet from the port at the other end of the link within the keepalive interval, the port waits for two more intervals. If the port still does not receive a health-check packet after waiting for three intervals, the port concludes that the link has failed and takes the port down.

UDLD for tagged ports

The default implementation of UDLD sends the packets untagged, even across tagged ports. If the untagged UDLD packet is received by a third-party switch, that switch may reject the packet. As a result, UDLD may be limited only to Brocade devices, since UDLD may not function on third-party switches.

To solve this issue, you can configure ports to send out UDLD control packets that are tagged with a specific VLAN ID. This feature also enables third party switches to receive the control packets that are tagged with the specified VLAN. For tagged operation, all of the following conditions must be met:

- A VLAN is specified when UDLD is configured
- The port belongs to the configured VLAN as tagged member
- All the devices across the UDLD link are in the same VLAN

For configuration details, refer to [“Enabling UDLD for tagged ports”](#) on page 687.

Configuration notes and feature limitations for UDLD

- UDLD is supported only on Ethernet ports.
- UDLD can be enabled on only one VLAN for tagged port.
- To configure UDLD on a trunk group, you must enable and configure the feature on each port of the group individually. Configuring UDLD on a trunk group primary port enables the feature on that port only.
- When UDLD is enabled on a trunk port, trunk threshold is not supported.
- Dynamic trunking is not supported. If you want to configure a trunk group that contains ports on which UDLD is enabled, you must remove the UDLD configuration from the ports. After you create the trunk group, you can re-add the UDLD configuration.
- If MRP is also enabled on the device, Brocade recommends that you set the MRP preforwarding time slightly higher than the default of 300 ms; for example, to 400 or 500 ms. Refer to [“Changing the hello and preforwarding times”](#) on page 627.

Enabling UDLD

NOTE

This section shows how to configure UDLD for untagged control packets. To configure UDLD for tagged control packets, refer to [“Enabling UDLD for tagged ports”](#).

To enable UDLD on a port, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#link-keepalive ethernet 0/1/1
```

To enable the feature on a trunk group, enter commands such as the following.

```
Brocade(config)#link-keepalive ethernet 0/1/1 ethernet 0/1/2
Brocade(config)#link-keepalive ethernet 0/1/3 ethernet 0/1/4
```

Syntax: [no] link-keepalive ethernet <port> [to <port> | ethernet <port>]

This command is not supported if you downgrade the device to FCX 6.0 or FSX5.1. In this case, use the following command to configure multiple ports:

Syntax: [no] link-keepalive ethernet <port> [ethernet <port>]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Enabling UDLD for tagged ports

To enable ports to receive and send UDLD control packets tagged with a specific VLAN ID, enter commands such as the following.

```
Brocade(config)#link-keepalive ethernet 1/18 vlan 22
```

This command enables UDLD on port 1/18 and allows UDLD control packet tagged with VLAN 22 to be received and sent on port 1/18.

Syntax: [no] link-keepalive ethernet <port> [vlan <vlan-ID>]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

For the <vlan-ID> variable, enter the ID of the VLAN that the UDLD control packets can contain to be received and sent on the port. If a VLAN ID is not specified, then UDLD control packets are sent out of the port as untagged packets.

NOTE

You must configure the same VLANs that will be used for UDLD on all devices across the network; otherwise, the UDLD link cannot be maintained.

Changing the Keepalive interval

By default, ports enabled for UDLD send a link health-check packet once every 500 ms. You can change the interval to a value from 1 – 60, where 1 is 100 ms, 2 is 200 ms, and so on. To change the interval, enter a command such as the following.

```
Brocade(config)#link-keepalive interval 3
```

Syntax: [no] link-keepalive interval <num>

The <num> parameter specifies how often the ports send a UDLD packet. You can specify from 1 – 60, in 100 ms increments. The default is 5 (500 ms).

Changing the Keepalive retries

By default, a port waits one second to receive a health-check reply packet from the port at the other end of the link. If the port does not receive a reply, the port tries four more times by sending up to four more health-check packets. If the port still does not receive a reply after the maximum number of retries, the port goes down.

You can change the maximum number of keepalive attempts to a value from 3 – 64. To change the maximum number of attempts, enter a command such as the following.

```
Brocade(config)#link-keepalive retries 4
```

Syntax: [no] link-keepalive retries <num>

The <num> parameter specifies the maximum number of times the port will try the health check. You can specify a value from 3 – 64. The default is 7.

Displaying UDLD information

This section describes the commands used to display information about a UDLD configuration.

Displaying information for all ports

To display UDLD information for all ports, enter the following command.

```
Brocade#show link-keepalive
Total link-keepalive enabled ports: 4
Keepalive Retries: 3      Keepalive Interval: 1 Sec.
```

Port	Physical Link	Logical Link	State	Link-vlan
4/1	up	up	FORWARDING	3
4/2	up	up	FORWARDING	
4/3	down	down	DISABLED	
4/4	up	down	DISABLED	

Syntax: show link-keepalive

TABLE 128 CLI display of UDLD information

Field	Description
Total link-keepalive enabled ports	The total number of ports on which UDLD is enabled.
Keepalive Retries	The number of times a port will attempt the health check before concluding that the link is down.

TABLE 128 CLI display of UDLD information (Continued)

Field	Description
Keepalive Interval	The number of seconds between health check packets.
Port	The port number.
Physical Link	The state of the physical link. This is the link between the Brocade port and the directly connected device.
Logical Link	The state of the logical link. This is the state of the link between this Brocade port and the Brocade port on the other end of the link.
State	The traffic state of the port.
Link-vlan	The ID of the tagged VLAN in the UDLD packet.

If a port is disabled by UDLD, the change also is indicated in the output of the **show interfaces brief** command. An example is given below.

```
Brocade#show interfaces brief
```

Port	Link	State	Dupl	Speed	Trunk	Tag	Priori	MAC	Name
1/1	Up	LK-DISABLE	None	None	None	No	level0	00e0.52a9.bb00	
1/2	Down	None	None	None	None	No	level0	00e0.52a9.bb01	
1/3	Down	None	None	None	None	No	level0	00e0.52a9.bb02	
1/4	Down	None	None	None	None	No	level0	00e0.52a9.bb03	

If the port was already down before you enabled UDLD for the port, the port state is listed as None.

Syntax: **show interfaces brief**

Displaying information for a single port

To display detailed UDLD information for a specific port, enter a command such as the following.

```
Brocade#show link-keepalive ethernet 4/1
```

Current State	: up	Remote MAC Addr	: 00e0.52d2.5100
Local Port	: 4/1	Remote Port	: 2/1
Local System ID	: e0927400	Remote System ID	: e0d25100
Packets sent	: 254	Packets received	: 255
Transitions	: 1	Link-vlan	: 100
Port blocking	: No	BM disabled	: No

Syntax: **show link-keepalive [ethernet [<slotnum>]/<portnum>]**

TABLE 129 CLI display of detailed UDLD information

Field	Description
Current State	The state of the logical link. This is the link between this Brocade port and the Brocade port on the other end of the link.
Remote MAC Addr	The MAC address of the port or device at the remote end of the logical link.
Local Port	The port number on this Brocade device.
Remote Port	The port number on the Brocade device at the remote end of the link.

TABLE 129 CLI display of detailed UDLD information (Continued)

Field	Description
Local System ID	A unique value that identifies this Brocade device. The ID can be used by Brocade technical support for troubleshooting.
Remote System ID	A unique value that identifies the Brocade device at the remote end of the link.
Packets sent	The number of UDLD health-check packets sent on this port.
Packets received	The number of UDLD health-check packets received on this port.
Transitions	The number of times the logical link state has changed between up and down.
Port blocking	Information used by Brocade technical support for troubleshooting.
Link-vlan	The ID of the tagged VLAN in the UDLD packet.
BM disabled	Information used by Brocade technical support for troubleshooting.

The **show interface ethernet** command also displays the UDLD state for an individual port. In addition, the line protocol state listed in the first line will say “down” if UDLD has brought the port down. An example is given below.

```
Brocade#show interface ethernet 1/1
FastEthernet1/1 is down, line protocol is down, link keepalive is enabled
  Hardware is FastEthernet, address is 00e0.52a9.bbca (bia 00e0.52a9.bbca)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown
  Member of L2 VLAN ID 1, port is untagged, port state is DISABLED
  STP configured to ON, priority is level0, flow control enabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 0 packets
  19 packets output, 1216 bytes, 0 underruns
  Transmitted 0 broadcasts, 19 multicasts, 0 unicasts
  0 output errors, 0 collisions, DMA transmitted 19 packets
```

In this example, the port has been brought down by UDLD. Notice that in addition to the information in the first line, the port state on the fourth line of the display is listed as DISABLED.

Clearing UDLD statistics

To clear UDLD statistics, enter the following command.

```
Brocade#clear link-keepalive statistics
```

Syntax: clear link-keepalive statistics

This command clears the Packets sent, Packets received, and Transitions counters in the **show link keepalive ethernet** [*<slotnum>/<portnum>*] display.

Protected link groups

A protected link group minimizes disruption to the network by protecting critical links from loss of data and power. In a protected link group, one port in the group acts as the primary or active link, and the other ports act as secondary or standby links. The active link carries the traffic. If the active link goes down, one of the standby links takes over.

During normal operation, the active port in a protected link group is enabled and the standby ports are logically disabled. If the active port fails, the Brocade device immediately enables one of the standby ports, and switches traffic to the standby port. The standby port becomes the new, active port.

About active ports

When you create a protected link group, you can optionally specify which port in the protected link group is the active port. If you do not explicitly configure an active port, the Brocade device dynamically assigns one. A dynamic active port is the first port in the protected link group that comes up (usually the lowest numbered port in the group).

Static and dynamic active ports operate as follows:

- A static active port (an active port that you explicitly configured) pre-empt other ports in the protected link group. So, if a static active link comes back up after a failure, the Brocade device will revert to this link as the active link.
- A dynamic active port (an active port assigned by the software) is non-pre-emptive. Therefore, if a dynamic active link comes back up after a failure, the Brocade device does not revert to this link, but continues carrying traffic on the current active link.

Using UDLD with protected link groups

You can use UDLD with protected link groups to detect uni-directional link failures and to improve the speed at which the device detects a failure in the link. Use UDLD and protected link groups simultaneously when the FastIron X Series device is connected to a device with slower link detection times.

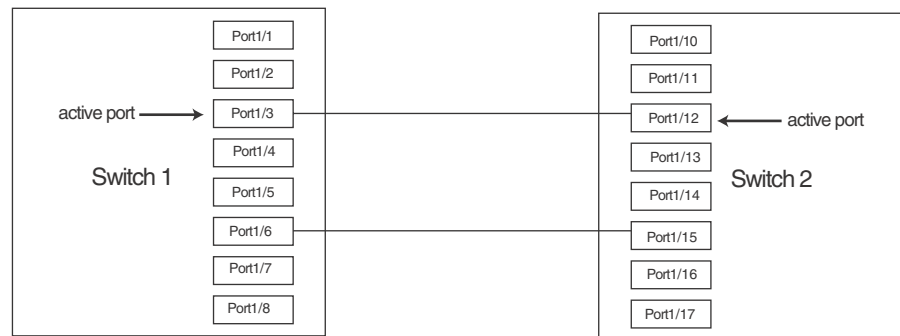
NOTE

When UDLD and protected links are configured on a port and the link goes down, protected links will not come up after UDLD becomes “healthy” again without first physically disabling then re-enabling the link.

UDLD with protected link groups configuration notes

- You can configure a maximum of 32 protected link groups.
- There is no restriction on the number of ports in a protected link group.
- Each port can belong to one protected link group at a time.
- On FastIron X Series devices, there is no restriction on the type of ports in a protected link group. FastIron X Series devices support protected link groups consisting of 10-GbE ports, Gbps fiber ports, 10/100/1000 copper ports, and 10/100 ports, or any combination thereof.

- FastIron WS and Brocade FCX Series devices support protected link groups consisting of Gbps fiber ports, 10/100/1000 copper ports, and 10/100 ports, or any combination thereof. These devices do not support protected link groups on 10-GbE ports.
- This feature is supported with tagged and untagged ports.
- This feature is supported with trunk ports.
- The protected link groups feature is not supported with LACP.
- There is no restriction on the properties of ports in a protected link group. For example, member ports can be in the same VLAN or in different VLANs.
- When two switches are connected together with links in a protected link group, and the ports connecting the switches together are part of a protected link group, you must configure two connecting ports (one port on each switch) as active ports of the protected link group. The following example illustrates this scenario.



The configuration for the above illustration is as follows.

Switch 1

```
Brocade(config)# protected-link-group 1 e 1/3 e 1/6
Brocade(config)# protected-link-group 1 active-port 1/3
```

Switch 2

```
Brocade(config)# protected-link-group 1 e 1/12 e 1/15
Brocade(config)# protected-link-group 1 active-port 1/12
```

Creating a protected link group and assigning an active port

Follow the steps given below to create a protected link group.

1. Specify the member ports in the protected link group. Enter a command such as the following.

```
Brocade(config)#protected-link-group 10 e 1 to 4
```

2. Optionally specify which port will be the active port for the protected link group. Enter a command such as the following.

```
Brocade(config)#protected-link-group 10 active-port 1
```

NOTE

If you do not explicitly configure an active port, the Brocade device automatically assigns one as the first port in the protected link group to come up.

These commands configure port e1 as the active port and ports e2 – e4 as standby ports. If port 1 goes down, the Brocade device enables the first available standby port, and switches the traffic to that port. Since the above configuration consists of a statically configured active port, the active port pre-empt's other ports in the protected link group. Refer to [“About active ports”](#) on page 691.

Syntax: `[no] protected-link-group <group-ID> ethernet <port> to <port>`

The `<group-ID>` parameter specifies the protected link group number. Enter a number from 1 – 32.

Each `ethernet <port> to <port>` specifies the ports in the protected link group. Specify the `port` variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – `stack-unit/slotnum/portnum`
- FSX 800 and FSX 1600 chassis devices – `slotnum/portnum`
- ICX devices – `slotnum/portnum`
- FESX compact switches – `portnum`

[no] protected-link-group <group-ID> active-port ethernet <port>

The `<group-ID>` parameter specifies the protected link group number. Enter a number from 1 – 32.

The `active-port ethernet <port>` defines the active port. Specify the `port` variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – `stack-unit/slotnum/portnum`
- FSX 800 and FSX 1600 chassis devices – `slotnum/portnum`
- ICX devices – `slotnum/portnum`
- FESX compact switches – `portnum`

Viewing information about protected link groups

You can use the following **show** commands to view information about protected link groups:

- **show protected-link-group**
- **show interface brief**
- **show interface**

The following shows example output for the **show protected-link-group** command.

```
Brocade#show protected-link-group
Group ID: 1
Member Port(s): ethe 1 to 7
Configured Active Port: 7
Current Active Port: 7
Standby Port(s): ethe 5
Total Number of Protected Link Groups: 1
```

Syntax: `show protected-link-group [<group-ID>]`

TABLE 130 CLI display of protected link group information

Field	Description
Group ID	The ID number of the protected link group.
Member Port(s)	The ports that are members of the protected link group.
Configured Active Port	The statically configured active port. If you do not statically configure an active port, this value will be "None".
Current Active Port	The current active port for the protected link group. If all member ports are down, this value will be "None".
Standby Port(s)	The member ports that are on standby.

The **show interface brief** command also displays information about protected link groups.

Example

```
Brocade#show int brief e 3 to 4
Port  Link    State    Dupl Speed Trunk Tag Priori MAC          Name
3     Up      Inactive Full Auto  None  Yes level0 0012.f2a8.7140
4     Up      Forward  Full 1G   None  Yes level0 0012.f2a8.7140
```

In the above output, the **State** of port 3 is **Inactive**, which means port 3 is an inactive port in a protected link group. For active ports in a protected link group, the **State** will be Active.

Syntax: **show interface brief ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **show interface** command also displays information about protected link groups.

```
Brocade#show int e 3
GigabitEthernet3 is up, line protocol is up, link keepalive is enabled
  Hardware is GigabitEthernet, address is 0012.f2a8.7140 (bia 0012.f2a8.7142)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of 3 L2 VLANs, port is tagged, port state is protected-link-inactive
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0
  ....
  some lines omitted for brevity
```

In the above output, the port state is **protected-link-inactive** which means port 3 is an inactive port in a protected link group.

Syntax: **show interface ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*

- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Trunk Groups and Dynamic Link Aggregation

Table 131 lists the individual Brocade FastIron switches and the trunk groups and dynamic link aggregation features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 131 Supported trunk group and dynamic link aggregation features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Trunk groups	Yes	Yes	Yes	Yes	Yes
Trunk threshold for static trunk groups	Yes	Yes	Yes	Yes	Yes
Flexible trunk group membership	Yes	Yes	Yes	Yes	Yes
Option to include Layer 2 in trunk hash calculation	No	Yes	Yes	Yes	Yes
802.3ad link aggregation (dynamic trunk groups)	Yes	Yes	Yes	Yes	Yes
Link Aggregation Control Protocol (LACP)	Yes	Yes	Yes	Yes	Yes
Single link LACP	Yes	Yes	Yes	Yes	Yes
Single link static trunk	Yes	Yes	Yes	Yes	Yes
Singleton LACP trunk	Yes	Yes	Yes	Yes	Yes

Trunk group overview

The trunk group feature allows you to manually configure multiple high-speed load-sharing links between two Brocade Layer 2 switches or Layer 3 switches or between a Brocade Layer 2 switch and Layer 3 switch and a server.

In addition to enabling load sharing of traffic, trunk groups provide redundant, alternate paths for traffic if any of the segments fail.

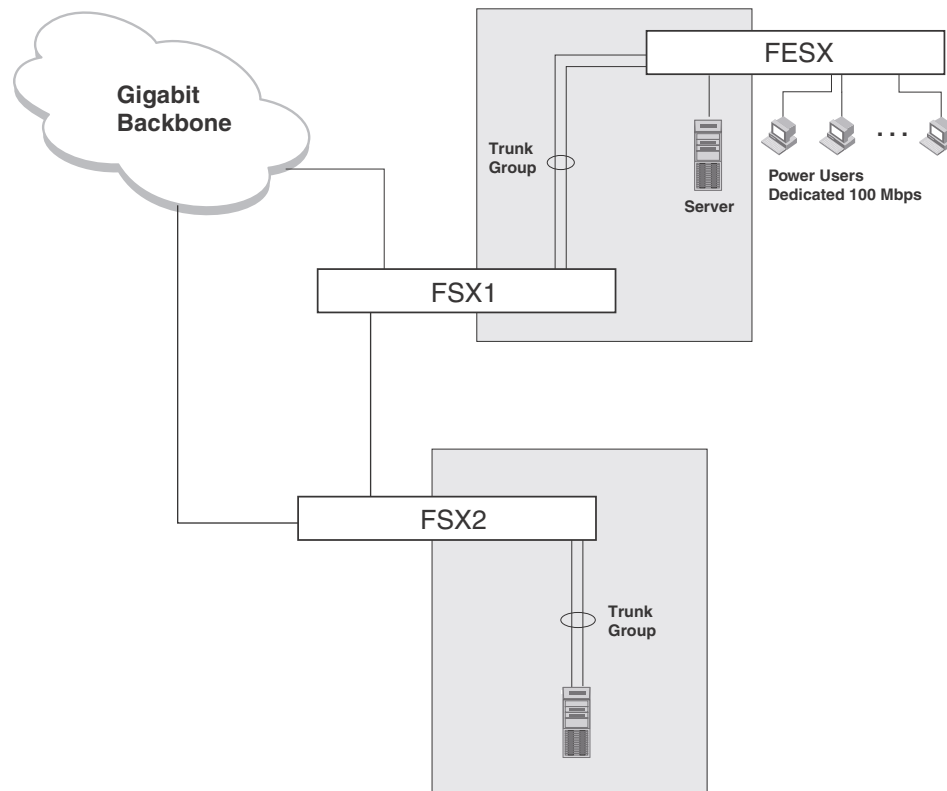
Trunk groups are manually configured aggregate links containing multiple ports. 802.3ad link aggregation is a protocol that dynamically creates and manages trunk groups.

NOTE

You can use both types of trunking on the same device. However, you can use only one type of trunking for a given port. For example, you can configure port 1/1 as a member of a static trunk group or you can enable 802.3ad link aggregation on the port, but you cannot do both.

Figure 62 shows an example of a configuration that uses trunk groups.

FIGURE 62 Trunk group application within a FastIron network



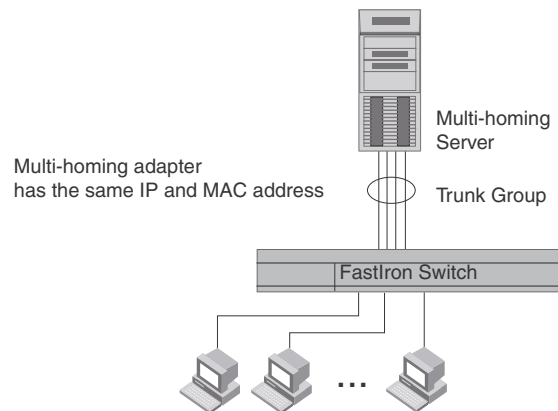
NOTE

The ports in a trunk group make a single logical link. Therefore, all the ports in a trunk group must be connected to the same device at the other end.

Trunk group connectivity to a server

To support termination of a trunk group, the server must have multiple network interface cards (NICs) or either a dual or quad interface card installed. The trunk server is designated as a server with multiple adapters or a single adapter with multiple ports that share the same MAC address and IP address.

[Figure 63](#) shows an example of a trunk group between a server and a Brocade device.

FIGURE 63 Trunk group between a server and a Brocade compact Layer 2 switch or Layer 3 switch

Trunk group rules

Table 132 lists the maximum number of trunk groups you can configure on a Brocade device and the valid number of ports in a trunk group. The table applies to static and LACP trunk ports.

TABLE 132 Trunk group support

Model	Maximum number of trunk groups		Valid number of ports in a group
	Static	LACP	
FCX624 FCX648	124	64	2, 3, 4, 5, 6, 7, or 8
FESX624	31	31	2, 3, 4, 5, 6, 7, or 8
FESX648	31	31	2, 3, 4, 5, 6, 7, or 8
FSX 800 FSX 1600	31	31	First Generation IPv4 devices: 2, 3, or 4 Second Generation IPv6 devices: 2, 3, 4, 5, 6, 7, or 8
	255	64	Third Generation devices: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, or 12
FWS624	13	13	2, 3, 4, 5, 6, 7, or 8
FWS648	25	25	2, 3, 4, 5, 6, 7, or 8
ICX 6430	29	29	2, 3, 4, 5, 6, 7, or 8
ICX 6450	124	64	
ICX 6610	124	64	

NOTE

On FastIron X Series devices, multi-slot trunk groups are supported on 1 GbE and 10 GbE ports. The valid number of slots in a multi-slot trunk group is as follows: First Generation IPv4 devices - 2, 3, or 4 ports; Second Generation IPv6 devices - 2, 3, 4, 5, 6, 7, or 8 ports; Third-generation devices - 2 to 12 ports.

- Legacy ports and 48 Gbps copper ports cannot be members of the same trunk group in hardware configurations such as the following:

- 48-port 10/100/1000 Mbps (RJ45) Ethernet PoE interface module (SX-FI48GPP) and IPv4/IPv6 interface modules or management modules with user ports.
- Combination of different generations of ports cannot be members of the same trunk group under the following hardware configurations:
 - 24-port fiber and copper Ethernet PoE/Fiber interfaces (SX-FI24GPP, SX-FI24GF) and IPv4/IPv6 interface modules or management modules with user ports
 - 2-port 10G and 8-port 10G interfaces (SX-FI8XG, SX-FI2XG) and the IPv4/IPv6 interface modules or management modules with user ports
- The FastIron SX chassis supports up to 12 ports per trunk (both static and LACP) and a maximum of 255 trunk groups only when the following interface modules are installed in the chassis:
 - SX-FI48GPP—48-port 10/100/1000 Mbps Ethernet PoE interface module
 - SX-FI24GPP—24-port Gigabit Ethernet copper interface module
 - SX-FI24HF—24-port Gigabit Ethernet fiber interface module
 - SX-FI2XG—2-port 10 Gigabit Ethernet interface module
 - SX-FI8XG—8-port 10 Gigabit Ethernet interface module
- Cross module trunking between different generation cards is not recommended when the SX chassis is populated with multiple generation cards (First, second and third). Cross module trunking is only supported between the same generation SX cards.
- You cannot configure a port as a member of a trunk group if 802.3ad link aggregation is enabled on the port.
- Unlike the FES and other Brocade devices, trunk groups on devices listed in [Table 132](#) are not classified as switch trunk groups or server trunk groups.
- Trunking is supported on 10-GbE ports.
- You cannot combine 1-Gbps and 10-Gbps ports in the same trunk group.
- For FastIron WS devices, mixed speed static and dynamic (LACP) trunks are not allowed and if configured will not come up. In these devices, the first four ports are 1 Gbps, the remaining ports are 10/100 Mbps. With both static and LACP trunking, all members of the trunk should be either the 1 Gbps ports or the 10/100 Mbps ports.
- Port assignment on a module need not be consecutive. The port range can contain gaps. For example, you can configure ports 1, 3, and 4 (excluding 2). Refer to [“Support for flexible trunk group membership”](#) on page 704.
- Although the FastIron devices have port ranges, they do not apply to trunk groups.
- You can select any port to be the primary port of the trunk group.
- Make sure the device on the other end of the trunk link can support the same number of ports in the link. For example, if you configure a 3-port trunk group on the FESX and the other end is a different type of switch, make sure the other switch can support a 3-port trunk group.
- All the ports must be connected to the same device at the other end.
- All trunk group member properties must match the lead port of the trunk group with respect to the following parameters:
 - Port tag type (untagged or tagged port)
 - Statically configured port speed and duplex
 - QoS priority

To change port parameters, you must change them on the primary port. The software automatically applies the changes to the other ports in the trunk group.

Configuration notes for FastIron devices in an IronStack

In a Brocade IronStack system, a trunk group may have port members distributed across multiple stack units. Both static and dynamic trunking are supported.

NOTE

Cascaded trunks between stack units are supported on Brocade ICX devices only.

To configure trunk groups for FastIron devices in an IronStack, use the CLI syntax in [“CLI syntax for configuring consecutive ports in a trunk group”](#) on page 707.

The following notes apply to FastIron stackable devices that are part of an IronStack:

- If a stack unit fails or is removed from the stack, its static trunk configuration becomes a reserved configuration on the Active Controller. Any remaining ports of the static trunk in the IronStack continue to function.
- When a new stack unit is added to an IronStack, the new unit receives running configuration and trunk-related information, including a list of ports that are up and are members of a trunk, from the Active Controller.
- Before merging two IronStack devices, make sure that there are no static trunks configured between them. This can result in self-looped ports.
- When an IronStack device with static trunks partitions into multiple IronStacks, loops and forwarding errors may occur. In these cases, user intervention is required to remove the loops.
- 10 Gbps links support up to eight ports in a trunk for stackable units.

Trunk group configuration examples

[Figure 64](#) shows some examples of valid 2-port trunk group links between devices. The trunk groups in this example are switch trunk groups between two Brocade devices. Ports in a valid 2-port trunk group on one device are connected to two ports in a valid 2-port trunk group on another device. The same rules apply to 3-port trunk groups, 4-port trunk groups, and so on.

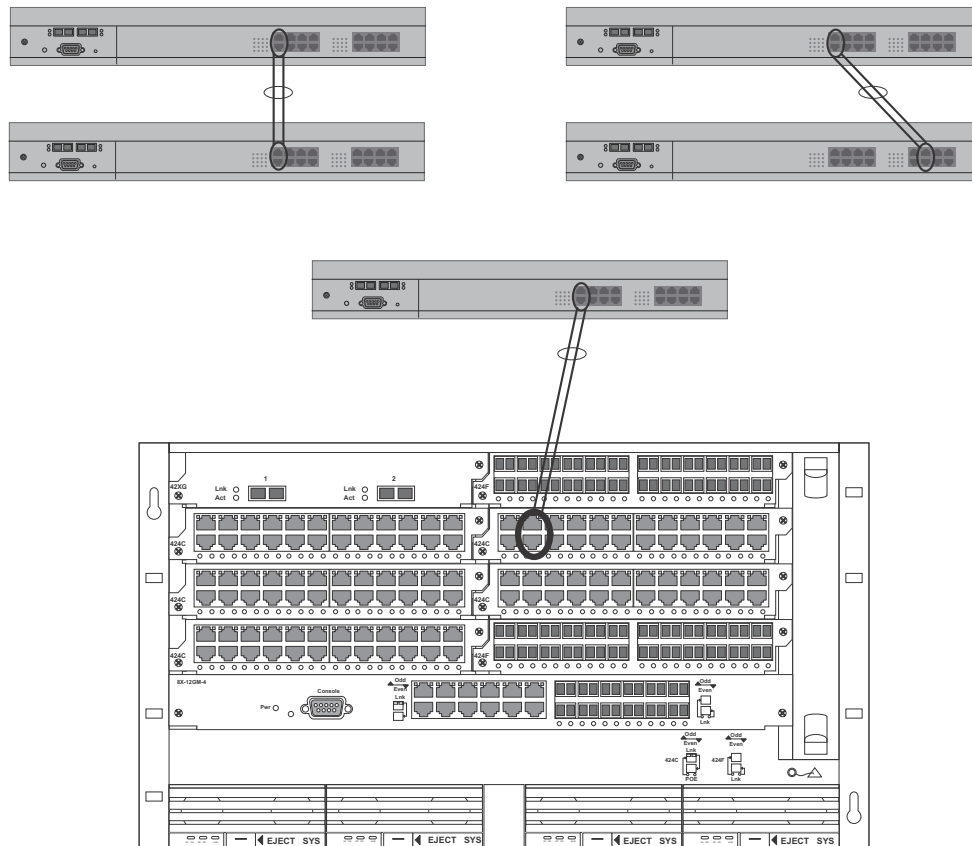
FIGURE 64 Examples of 2-port and 3-port trunk groups

Figure 65 shows examples of two chassis devices connected by multi-slot trunk groups.

FIGURE 65 Examples of multi-slot trunk groups

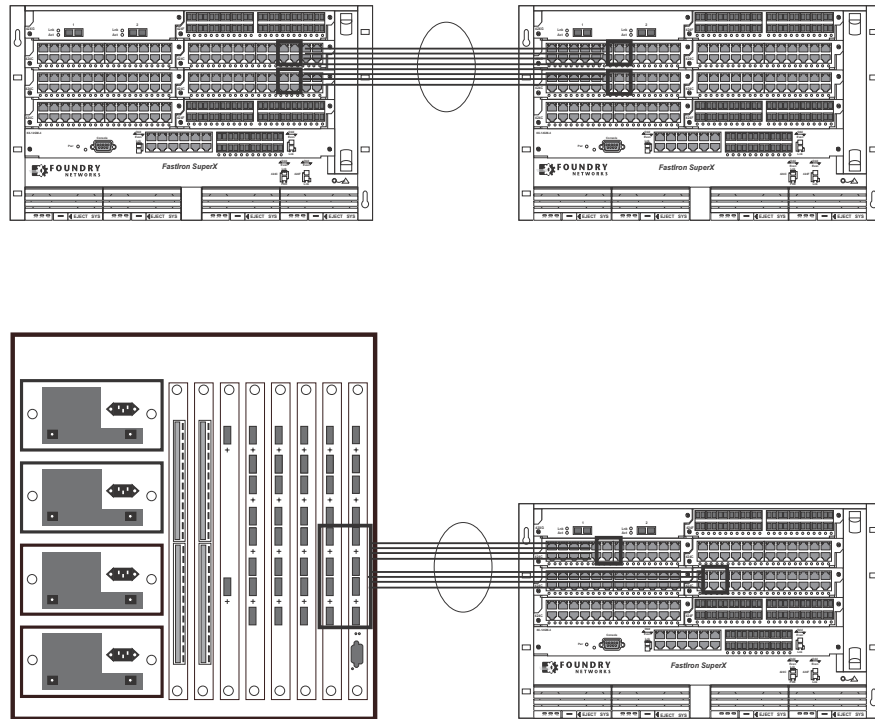
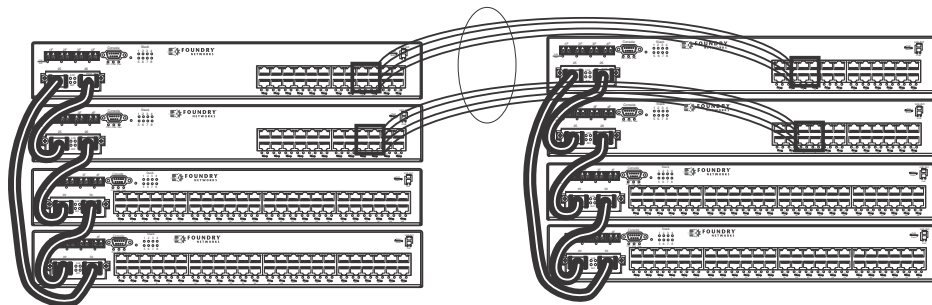


Figure 66 shows two IronStacks connected by multi-slot trunk groups.

FIGURE 66 Two IronStacks connected by multi-slot trunk groups



Support for flexible trunk group membership

FastIron devices support flexible trunk group membership, which eliminates the requirement for port membership to be consecutive, and allows the trunking of ports on non-consecutive interfaces. For example, you can configure ports e 2/4, 2/6, and 2/7 (excluding e 2/5) together on a module as a trunk group. This feature is supported on static and LACP trunk ports, as well as 1-GbE and 10-GbE ports. Flexible trunk ports follow the same rules as listed in “[Trunk group rules](#)” on page 699.

NOTE

For FCX devices only, flexible trunk group membership is supported from Web Management, but not from SNMP. For all other FastIron devices, this feature is *not* supported from Web Management or SNMP.

For configuration details, refer to [“CLI syntax for configuring non-consecutive ports in a trunk group”](#) on page 708.

Trunk group load sharing

Brocade devices load-share across the ports in the trunk group. The method used for the load sharing depends on the device type and traffic type (Layer 2 or Layer 3).

NOTE

Layer 2 and Layer 3 AppleTalk traffic is not load-balanced. Layer 3 routed IP or IPX traffic also is not load balanced. These traffic types will however still be forwarded on the trunk ports.

Support for IPv6 when sharing traffic across a trunk group

Brocade devices that support IPv6 take the IPv6 address for a packet into account when sharing traffic across a trunk group. The load sharing is performed in the same way it is for IPv4 addresses; that is, trunk types with a traffic load that is shared based on IPv4 address information can now use IPv6 addresses to make the load sharing decision.

Load sharing occurs as described in [Table 133](#) or [Table 134](#).

Load balancing for unknown unicast, multicast, and broadcast traffic

Brocade devices load balance unknown unicast, multicast, and broadcast traffic based on the source port and VLAN ID and not on any source or destination information in the packet.

For example, when the switch receives unknown unicast, multicast, and broadcast packets, and the packets are from the same source port, the packets are forwarded to the same port of the trunk group. Conversely, when the switch receives unknown unicast, multicast, and broadcast packets, and the packets are from different source ports, the packets are load balanced across all the ports of the trunk group.

Note that this does not apply to known unicast traffic, which is always load balanced across all the ports of a trunk group based on the traffic's Layer 2 and Layer 3 source and destination parameters.

How trunk load sharing works

The load balancing method for bridged traffic varies depending on the traffic type. Load balancing for routed traffic is always based on the source and destination IP addresses and protocol field (not applicable for FastIron stackable devices).

[Table 133](#) and [Table 134](#) show how the different Brocade devices load balance traffic.

NOTE

[Table 133](#) and [Table 134](#) do not include unknown unicast, multicast, and broadcast traffic. Refer to [“Load balancing for unknown unicast, multicast, and broadcast traffic”](#).

[Table 133](#) shows how the FastIron X Series devices load balance traffic across the ports in a trunk group.

TABLE 133 Trunk group load sharing on FastIron X Series devices

Traffic type	Load balancing method
Layer 2 Bridged non-IP	Source and destination MAC addresses
Layer 2 Bridged TCP/UDP	Source and destination MAC addresses, source and destination IP addresses, and source and destination TCP/UDP ports
Layer 2 Bridged IP (non-TCP/UDP)	Source and destination MAC addresses, and source and destination IP addresses
Layer 3 Routed traffic	Source and destination IP addresses and protocol field

[Table 134](#) describes how the FastIron stackable devices load balance traffic.

TABLE 134 Trunk group load sharing on FastIron stackable devices

Traffic type	Load balancing method
Layer 2 Bridged Non-IP	Source and destination MAC addresses
Layer 2 Bridged IPv4 TCP/UDP	Source and destination IP addresses, and source and destination TCP/UDP ports
Layer 2 Bridged IPv4 Non-TCP/UDP	Source and destination IP addresses
Layer 2 Bridged IPv6 TCP/UDP	Source and destination IP addresses, source and destination TCP and UDP ports, and flow label
Layer 2 Bridged IPv6 Non-TCP/UDP	Source and destination TCP and UDP ports, and flow label
Layer 3 Routed traffic	Source and destination IP addresses and protocol field

Adding Layer 2 information to trunk hash output

NOTE

Adding Layer 2 information to trunk hash output is *not* supported on FastIron X Series devices.

FastIron stackable devices support the option to include Layer 2 information in the trunk hash calculation for IP packets. Use the following CLI command.

```
Brocade(config)# trunk hash-options include-layer2
```

The **trunk hash-options include-layer2** command adds Layer 2 information (text in bold) to the following load-balancing parameters:

- Non-IP: Source and destination MAC addresses
- IPv4 TCP/UDP: Source and destination IP addresses, and source and destination TCP/UDP ports, **Source MAC, Destination MAC**
- IPv4 Non-TCP/UDP: Source and destination IP addresses, **Source MAC, Destination MAC**
- IPv6 TCP/UDP: Source and destination IP addresses, source and destination TCP and UDP ports, and flow label, **Source MAC, Destination MAC**
- IPv6 Non-TCP/UDP: Source and destination TCP and UDP ports, and flow label, **Source MAC, Destination MAC**

Syntax: [no] trunk hash-options include-layer2

Configuring a trunk group

1. Disconnect the cables from those ports on both systems that will be connected by the trunk group. Do not configure the trunk groups with the cables connected.

NOTE

If you connect the cables before configuring the trunk groups and rebooting, the traffic on the ports can create a spanning tree loop.

2. Configure the trunk group on one of the two Layer 2 switches or Layer 3 switches involved in the configuration.

NOTE

Downtime is incurred when adding a new port to a trunk group. It is suggested that you schedule the addition of ports to a trunk group to minimize downtime and its impact to the production network.

3. Save the configuration changes to the startup-config file.
4. Dynamically place the new trunk configuration into effect by entering the **trunk deploy** command at the global CONFIG level of the CLI.
5. If the device at the other end of the trunk group is another Layer 2 switch or Layer 3 switch, repeat steps 2 through 4 for the other device.
6. When the trunk groups on both devices are operational, reconnect the cables to those ports that are now configured as trunk groups, starting with the first port (lead port) of each trunk group.
7. To verify the link is operational, use the **show trunk** command.

CLI syntax for configuring consecutive ports in a trunk group

This section describes the CLI syntax for configuring consecutive ports in a trunk group. To configure non-consecutive ports, refer to [“CLI syntax for configuring non-consecutive ports in a trunk group”](#) on page 708. Configuration examples are shown in later sections of this chapter.

To configure a trunk group consisting of two groups of two ports each, enter commands such as the following.

```
Brocade(config)#trunk ethernet 1/1 to 1/2 ethernet 3/3 to 3/4
Trunk will be created in next trunk deploy
Brocade(config)#write memory
Brocade(config)#trunk deploy
```

Syntax: [no] trunk ethernet <primary-port> to <port> [ethernet <primary-port> to <port>]

Syntax: trunk deploy

Each **ethernet** parameter introduces a port group.

The *<primary-port>* variable specifies the primary port. Notice that each port group must begin with a primary port. The primary port of the first port group specified (which must be the group with the lower port numbers) becomes the primary port for the entire trunk group.

Specify the *<primary-port>* and *<port>* variables in one of the following formats:

- FWS and FCX stackable switches – *<stack-unit/slotnum/portnum>*
- FSX 800 and FSX 1600 chassis devices – *<slotnum/portnum>*
- ICX devices = *<slotnum/portnum>*
- FESX compact switches – *<portnum>*

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or use a combination of both.

CLI syntax for configuring non-consecutive ports in a trunk group

This section describes the CLI syntax for configuring non-consecutive ports in a trunk group. Configuration examples are shown in later sections of this chapter.

To configure a 4-port trunk with non-consecutive ports on a FastIron chassis device, enter a command such as the following.

```
Brocade(config)#trunk ethernet 1/7 ethernet 1/9 ethernet 1/11 ethernet 1/21
```

This creates a 4-port trunk group with the following members.

members 1/7, 1/9, 1/11, and 1/21.

To configure a 4-port trunk with non-consecutive ports on a FastIron stackable device, enter a command such as the following.

```
Brocade(config)#trunk ethernet 1/1/7 ethernet 1/1/9 ethernet 1/1/11 ethernet 1/1/21
```

This creates a 4-port trunk group with the following members.

members 1/1/7, 1/1/9, 1/1/11, and 1/1/21.

Syntax: [no] trunk ethernet *<port>* ethernet *<port>* | to ethernet *<port>*...

The *port* variable specifies an individual port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can enter the **ethernet port** parameter multiple times to specify a list.

The **to** keyword indicates that you are specifying a range of ports. Specify the lower port number in the range first, then **to**, then the higher port number in the range.

Example 1: Configuring the trunk groups

To configure the trunk groups shown in [Figure 62](#), enter the following commands. Notice that the commands are entered on multiple devices.

To configure the trunk group link between FSX1 and the FESX, enter the following commands.

NOTE

The text shown in *italics* in the following CLI example shows messages echoed to the screen in answer to the CLI commands entered.

```
Brocade(config)#trunk e 1/5 to 1/8
Trunk will be created in next trunk deploy
Brocade(config)#write memory
Brocade(config)#trunk deploy
```

To configure the trunk group link between FSX2 and the server, enter the following commands.

```
Brocade(config)#trunk e 1/2 to 1/4
Trunk will be created in next trunk deploy
Brocade(config)#write memory
Brocade(config)#trunk deploy
```

You then configure the trunk group on the FESX.

```
Brocade(config)#trunk ethernet 17 to 18
Trunk will be created in next trunk deploy
Brocade(config)#write memory
Brocade(config)#trunk deploy
```

NOTE

The **trunk deploy** command dynamically places trunk configuration changes into effect, without a software reload.

Example 2: Configuring a trunk group that spans two Ethernet modules in a chassis device

This section shows how to configure a trunk group that spans two modules in a chassis device.

Multi-slot trunk groups are supported on 1-GbE ports, 10-GbE ports, as well as on static and LACP trunk ports. For multi-slot trunk group rules, refer to [Table 136](#) on page 723.

To configure a trunk group consisting of two groups of ports, 1/1 to 1/2 on module 1 and 4/5 to 4/6 on module 4, enter the following commands.

```
Brocade(config)#trunk ethernet 1/1 to 1/2 ethernet 4/5 to 4/6
Trunk will be created in next trunk deploy
Brocade(config)#write memory
Brocade(config)#trunk deploy
```

NOTE

The **trunk deploy** command dynamically places trunk configuration changes into effect, without a software reload.

NOTE

If you disable a module that is part of a multi-slot trunk group, the corresponding trunk ports will go down, but the remaining ports in the trunk will remain up and running. However, when you re-enable the module, all of the trunk ports will go down and then come back up. In other words, trunk ports are redeployed when a module is re-enabled.

Example 3: Configuring a multi-slot trunk group with one port per module

You can select one port per module in a multi-slot trunk group. This feature is supported on 1-GbE and 10-GbE ports, as well as on static and LACP trunk ports. For multi-slot trunk group rules, refer to [Table 136](#) on page 723.

To configure a 2-port multi-slot trunk group consisting of port 1/1 on module 1 and port 2/1 on module 2, enter the following commands.

```
Brocade(config)#trunk ethernet 1/1 to 1/1 ethernet 2/1 to 2/1
Trunk will be created in next trunk deploy
Brocade(config)#write memory
Brocade(config)#trunk deploy
```

NOTE

The **trunk deploy** command dynamically places trunk configuration changes into effect, without a software reload.

NOTE

If you disable a module that is part of a multi-slot trunk group, the corresponding trunk ports will remain up and running. However, when you re-enable the module, all of the trunk ports will go down and then come back up. In other words, trunk ports are redeployed when a module is re-enabled.

Example 4: Configuring a trunk group of 10 Gbps Ethernet ports

You can configure 10 Gbps Ethernet ports together in a trunk group.

To configure a trunk group containing two 10 Gbps Ethernet ports, enter commands such as the following.

```
Brocade(config)#trunk ethernet 1/1 to 2/1
Brocade(config-trunk-1/1-2/1)# write memory
Brocade(config-trunk-1/1-2/1)# exit
Brocade(config)#trunk deploy
```

The commands configure a trunk group consisting of 10 Gbps Ethernet ports 1/1 and 2/1, and then deploy the trunk group. The trunk configuration does not take effect until you deploy it.

Example 5: Configuring a static trunk group for devices in an IronStack

The following example shows how to configure a static trunk group for units in an IronStack, and the result of the configured trunk group in the **show trunk** output.

```
4 Router(config)#sh trunk
Configured trunks:
Trunk ID: 1
Hw Trunk ID: 1
Ports_Configured: 2
Primary Port Monitored: Individually
Ports  PortName Port_Status Monitor Rx_Mirr Tx_Mirr Monitor_Dir
1/1/1  none     enable    off    N/A     N/A     N/A
1/1/2  none     enable    on     1/1/15 1/1/15  both
Operational trunks:
Trunk ID: 1
Hw Trunk ID: 1
Duplex: Full
Speed: 1G
Tag: No
Priority: level0
Active Ports: 2
Ports  Link_Status port_state
1/1/1  active      Forward
1/1/2  active      Forward
```

Additional trunking options

The following trunking options can be performed on ports in deployed trunks. These options are supported on static trunk ports. Except where noted, these options are also supported on dynamic (LACP) trunk ports on all Brocade FastIron devices.

- Naming a trunk port
- Disabling or re-enabling a trunk port
- Deleting a static trunk group (applies to static trunks only)
- Specifying the minimum number of ports in a trunk group (applies to static trunks only)
- Monitoring a trunk port
- Configuring outbound rate shaping on a trunk port
- Enabling sFlow forwarding on an individual port in a trunk
- Setting the sFlow sampling rate on an individual port in a trunk

NOTE

Depending on the operational state of LACP-enabled ports, at any time these ports may join a trunk group, change trunk group membership, exit a trunk group, or possibly never join a trunk group. Therefore, before configuring trunking options on LACP-enabled ports (for example, naming the port, disabling the port, and so on), verify the actual trunk group port membership using the **show trunk** command. To view the status of LACP, use the **show link-aggregate** command.

Naming a trunk port

Naming a trunk port is supported on individual ports of a static trunk group.

To name an individual port in a trunk group, enter a command such as the following at the trunk group configuration level.

```
Brocade(config)#trunk e 4/1 to 4/4
Brocade(config-trunk-4/1-4/4)#port-name customer1 ethernet 4/2
```

This command assigns the name “customer1” to port 4/2 in the trunk group consisting of ports 4/1 to 4/4.

Syntax: [no] **port-name** <ASCII string> **ethernet** <port>

The <ASCII string> variable specifies the port name. The name can be up to 49 characters long.

The <port> variable is a valid port in the trunk group.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Disabling or re-enabling a trunk port

NOTE

When you enable or disable auto-negotiated combination ports on FESX devices, the ports may flap for a few seconds before the link is up.

Disabling or re-enabling trunk ports is supported on individual ports of a static trunk group.

You can disable or re-enable individual ports in a trunk group. To disable an individual port in a trunk group, enter commands such as the following at the trunk group configuration level.

```
Brocade(config)#trunk e 4/1 to 4/4
Brocade(config-trunk-4/1-4/4)#config-trunk-ind
Brocade(config-trunk-4/1-4/4)#disable ethernet 4/2
```

Syntax: [no] **config-trunk-ind**

Syntax: [no] **disable ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **config-trunk-ind** command enables configuration of individual ports in the trunk group. If you do not use this command, the **disable** and **enable** commands will be valid only for the primary port in the trunk group and will disable or enable all ports in the trunk group. You need to enter the **config-trunk-ind** command only once in a trunk group. After you enter the command, all applicable port configuration commands apply to individual ports only.

NOTE

If you enter **no config-trunk-ind**, all port configuration commands are removed from the individual ports and the configuration of the primary port is applied to all the ports. Also, once you enter the **no config-trunk-ind** command, the **enable**, **disable**, and **monitor** commands are valid only on the primary port and apply to the entire trunk group.

To enable an individual port in a trunk group, enter commands such as the following at the trunk group configuration level.

```
Brocade(config-trunk-4/1-4/4)#config-trunk-ind
Brocade(config-trunk-4/1-4/4)#enable ethernet 4/2
```

Syntax: **enable ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Disabling or re-enabling a range or list of trunk ports

The **disable port-name** command disables the port. The states of other ports in the trunk group are not affected.

If you have configured a name for the trunk port, you can specify the port name, as shown in the following example.

```
Brocade(config-trunk-4/1-4/4)#config-trunk-ind
Brocade(config-trunk-4/1-4/4)#disable port-name customer1
```

Syntax: **disable port-name** <portname>

To disable a range of ports in a trunk group, enter commands such as the following.

```
Brocade(config)#trunk ethernet 2/1 to 2/4
Brocade(config-trunk-2/1-2/4)#config-trunk-ind
Brocade(config-trunk-2/1-2/4)#disable ethernet 2/3 to 2/4
```

This command disables ports 2/3 and 2/4 in trunk group 2/1 to 2/4.

To disable a list of ports, enter a command such as the following.

```
Brocade(config-trunk-2/1-2/4)#disable ethernet 2/1 ethernet 2/3 ethernet 2/4
```

This command disables ports 2/1, 2/3, and 2/4 in the trunk group.

You can specify a range and a list on the same command line. For example, to re-enable some trunk ports, enter a command such as the following.

```
Brocade(config-trunk-2/1-2/4)#enable ethernet 2/1 to 2/2 ethernet 2/4
```

Syntax: [no] disable ethernet <port> to <port> | ethernet <port>

Syntax: [no] enable ethernet <port> to <port> | ethernet <port>

The <port> variable specifies an individual port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can enter the **ethernet** <port> parameter multiple times to specify a list.

The **to** keyword indicates that you are specifying a range. Specify the lower port number in the range first, then **to**, then the higher port number in the range.

Deleting a static trunk group

Use the **no trunk ethernet** command to delete a static trunk group.

NOTE

To delete an LACP trunk group, use the **no link-aggregate active | passive** command.

To delete a trunk group, use the **no** form of the command you used to create the trunk group. For example, to remove one of the trunk groups configured in the previous examples, enter the following command.

```
Brocade(config)#no trunk ethernet 1/1 to 1/2 ethernet 3/3 to 3/4
```

Syntax: no trunk ethernet <port> to <port> [ethernet <port> to <port>]...

The *port* variable specifies an individual port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You can enter the **ethernet** *port* parameter multiple times to specify a list.

The **to** keyword indicates that you are specifying a range of ports. Specify the lower port number in the range first, then **to**, then the higher port number in the range.

Specifying the minimum number of ports in a static trunk group

You can configure Brocade devices to disable all of the ports in a trunk group when the number of active member ports drops below a specified threshold value. For example, if a trunk group has four ports, and the threshold for the trunk group is three, then the trunk group is disabled if the number of available ports in the trunk group drops below three. If the trunk group is disabled, then traffic is forwarded over a different link or trunk group.

For example, the following commands establish a trunk group consisting of four ports, and then establish a threshold for this trunk group of three ports.

```
Brocade(config)#trunk e 3/31 to 3/34
Brocade(config-trunk-3/31-3/34)#threshold 3
```

In this example, if the number of active ports drops below three, then all the ports in the trunk group are disabled.

Syntax: [no] **threshold** <number>

The <number> variable specifies a threshold number from 2 (default) up to the number of ports in a trunk group. The total number of threshold ports must be greater than 1.

NOTE

When using the **no threshold** command, it is not necessary to enter a number.

Configuration notes for specifying ports in a static trunk group

- Specifying ports in a trunk group is supported on static trunk groups only. It is not supported on LACP trunk groups.
- When unidirectional link detection (UDLD) is enabled on a trunk port, the trunk threshold is not supported.
- The **disable module** command can be used to disable the ports on a module. However, on 10 Gbps modules, the **disable module** command does not cause the remote connection to be dropped. If a trunk group consists of 10 Gbps ports, and you use the **disable module** command to disable ports in the trunk group, which then causes the number of active ports in the trunk group to drop below the threshold value, the trunk group is not disabled.
- If you establish a threshold for a trunk used in conjunction with Metro Ring Protocol (MRP) on 10 Gbps interfaces, then you must also enable Link Fault Signaling (LFS).
- If you specify a threshold for a trunk group, the other end of the trunk group must also have the same threshold configuration.

Monitoring a trunk port

You can monitor the traffic on an individual port of a static trunk group. For configuration details, refer to [“Monitoring an individual trunk port”](#) on page 930.

Configuring outbound rate shaping for a trunk port

You can configure the maximum rate at which outbound traffic is sent out on a static trunk port. For configuration details, refer to [“Configuring outbound rate shaping for a trunk port”](#) on page 1953.

Enabling sFlow forwarding on a trunk port

You can enable sFlow forwarding on individual ports of a static trunk group. For configuration details, refer to [“Enabling sFlow forwarding on individual trunk ports”](#) on page 546.

Setting the sFlow sampling rate on a trunk port

You can configure an individual trunk port to use a different sampling rate than the global default sampling rate. This feature is supported on static trunk ports. For configuration details, refer to [“Changing the sampling rate for a trunk port”](#) on page 544.

Displaying trunk group configuration information

To display configuration information for the trunk groups, use the **show trunk** command. This command displays information for configured trunk groups and operational trunk groups. A configured trunk group is one that has been configured in the software but has not been placed into operation by a reset or reboot. An operational trunk group is one that has been placed into operation by a reset or reboot.

Enter the following command at any CLI level.

Syntax: **show trunk** [**ethernet** <port> **to** <port>]

The **ethernet** parameter introduces a port or port group.

The <port> variable specifies an individual port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **to** keyword indicates that you are specifying a range of ports. Specify the lower port number in the range first, then **to**, then the higher port number in the range.

NOTE

The **show trunk** command does not display any form of trunk when links are up.

[Table 135](#) describes the information displayed by the **show trunk** command.

TABLE 135 CLI trunk group information

Field	Description
Trunk ID	The trunk group number. The software numbers the groups in the display to make the display easy to use.
HW Trunk ID	The trunk ID.
Duplex	<p>The mode of the port, which can be one of the following:</p> <ul style="list-style-type: none"> • None – The link on the primary trunk port is down. • Full – The primary port is running in full-duplex. • Half – The primary port is running in half-duplex. <p>NOTE: This field and the following fields apply only to operational trunk groups.</p>
Speed	<p>The speed set for the port. The value can be one of the following:</p> <ul style="list-style-type: none"> • None – The link on the primary trunk port is down. • 10 – The port speed is 10 Mbps. • 100 – The port speed is 100 Mbps. • 1 Gbps – The port speed is 1000 Mbps.

TABLE 135 CLI trunk group information (Continued)

Field	Description
Tag	Indicates whether the ports have 802.1Q VLAN tagging. The value can be Yes or No.
Priority	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 through 7.
Active Ports	The number of ports in the trunk group that are currently active.
Ports	The ports in the trunk group.
Link_Status	The link status of each port in the trunk group.
LACP_Status	<p>Displays the status of the aggregate links, which could be one of the following states:</p> <ul style="list-style-type: none"> • Ready - The port is functioning normally in the trunk group and is able to transmit and receive LACP packets. • Expired - The time has expired (as determined by timeout values) and the port has shut down because the port on the other side of the link has stopped transmitting packets. • Down - The port physical link is down. <p>For more information about this feature, refer to the section “Displaying and determining the status of aggregate links” on page 731.</p>
Load Sharing	The number of traffic flows currently being load balanced on the trunk ports. All traffic exchanged within the flow is forwarded on the same trunk port. For information about trunk load sharing, refer to “Trunk group load sharing” on page 705.

Viewing the first and last ports in a trunk group

Output for many of the **show** commands will show the first and last port in a trunk as FirstPort-LastPort, if the ports are consecutive, and FirstPort*LastPort if the ports are not consecutive. This output is shown in the following **show mac** command, which displays the first and last ports.

```
Brocade#show mac
Total active entries from all ports = 1
MAC-Address Port Type Index
0007.e910.c201 1/1/7*1/1/21 Dynamic 2920
```

For a trunk group with members 1/1/7 to 1/1/9, the output from the **show mac** command resembles the following.

```
Brocade#show mac
Total active entries from all ports = 1
MAC-Address Port Type Index
0007.e910.c201 1/1/7-1/1/9 Dynamic 2920
```

Dynamic link aggregation

Brocade software supports the IEEE 802.3ad standard for link aggregation. This standard describes the Link Aggregation Control Protocol (LACP), a mechanism for allowing ports on both sides of a redundant link to form a trunk link (aggregate link), without the need for manual configuration of the ports into trunk groups.

When you enable link aggregation on a group of Brocade ports, the Brocade ports can negotiate with the ports at the remote ends of the links to establish trunk groups.

The link aggregation feature automates trunk configuration but can coexist with the Brocade trunk group feature. Link aggregation parameters do not interfere with trunk group parameters.

NOTE

Use the link aggregation feature only if the device at the other end of the link you want to aggregate also supports IEEE 802.3ad link aggregation. Otherwise, you must manually configure the trunk links.

Link aggregation support is disabled by default. You can enable the feature on an individual port basis, in active or passive mode:

- **Active mode** – When you enable a port for active link aggregation, the Brocade port can exchange standard LACP Data Unit (LACPDU) messages to negotiate trunk group configuration with the port on the other side of the link. In addition, the Brocade port actively sends LACPDU messages on the link to search for a link aggregation partner at the other end of the link, and can initiate an LACPDU exchange to negotiate link aggregation parameters with an appropriately configured remote port.
- **Passive mode** – When you enable a port for passive link aggregation, the Brocade port can exchange LACPDU messages with the port at the remote end of the link, but the Brocade port cannot search for a link aggregation port or initiate negotiation of an aggregate link. Thus, the port at the remote end of the link must initiate the LACPDU exchange.

NOTE

Brocade recommends that you disable or remove the cables from the ports you plan to enable for dynamic link aggregation. Doing so prevents the possibility that LACP will use a partial configuration to talk to the other side of a link. A partial configuration does not cause errors, but does sometimes require LACP to be disabled and re-enabled on both sides of the link to ensure that a full configuration is used. It is easier to disable a port or remove its cable first. This applies both for active link aggregation and passive link aggregation.

The following rules apply to units in an IronStack:

- With LACP trunk configurations, the LACP system ID is the MAC address of the Active Controller. If the LACP system ID changes, the entire trunk flaps and an STP reconvergence occurs.
- Link aggregation can be used to form multi-slot aggregate links on stack units, but the link aggregation keys must match for the port groups on each stack unit. For example, to configure an aggregate link containing ports 1/1/1 through 1/1/4, and 3/1/5 through 3/1/8, you must change the link aggregation key on one or both port groups so that the key is the same for all eight ports. Refer to [“IronStack LACP trunk group configuration example”](#).

IronStack LACP trunk group configuration example

To configure a trunk group consisting of two groups of two ports each on an IronStack, enter commands similar to the following.

```
Brocade(config)#interface ethernet 1/1/1 to 1/1/4
Brocade(config-mif-1/1/1-1/1/4)#link-aggregate off
Brocade(config-mif-1/1/1-1/1/4)#link-aggregate configure key 10000
Brocade(config-mif-1/1/1-1/1/4)#link-aggregate active
Brocade(config-mif-1/1/1-1/1/4)#interface ethernet 3/1/5 to 3/1/8
Brocade(config-mif-3/1/5-3/1/8)#link-aggregate off
Brocade(config-mif-3/1/5-3/1/8)#link-aggregate configure key 10000
Brocade(config-mif-3/1/5-3/1/8)#link-aggregate active
```

This command sequence changes the key for ports 1/1/1 through 1/1/4 and 3/1/5 through 3/1/8 to 10000. Because all ports in an aggregate link must have the same key, this example forms a multi-slot aggregate link for ports 1/1/1 through 1/1/4 and 3/1/5 through 3/1/8.

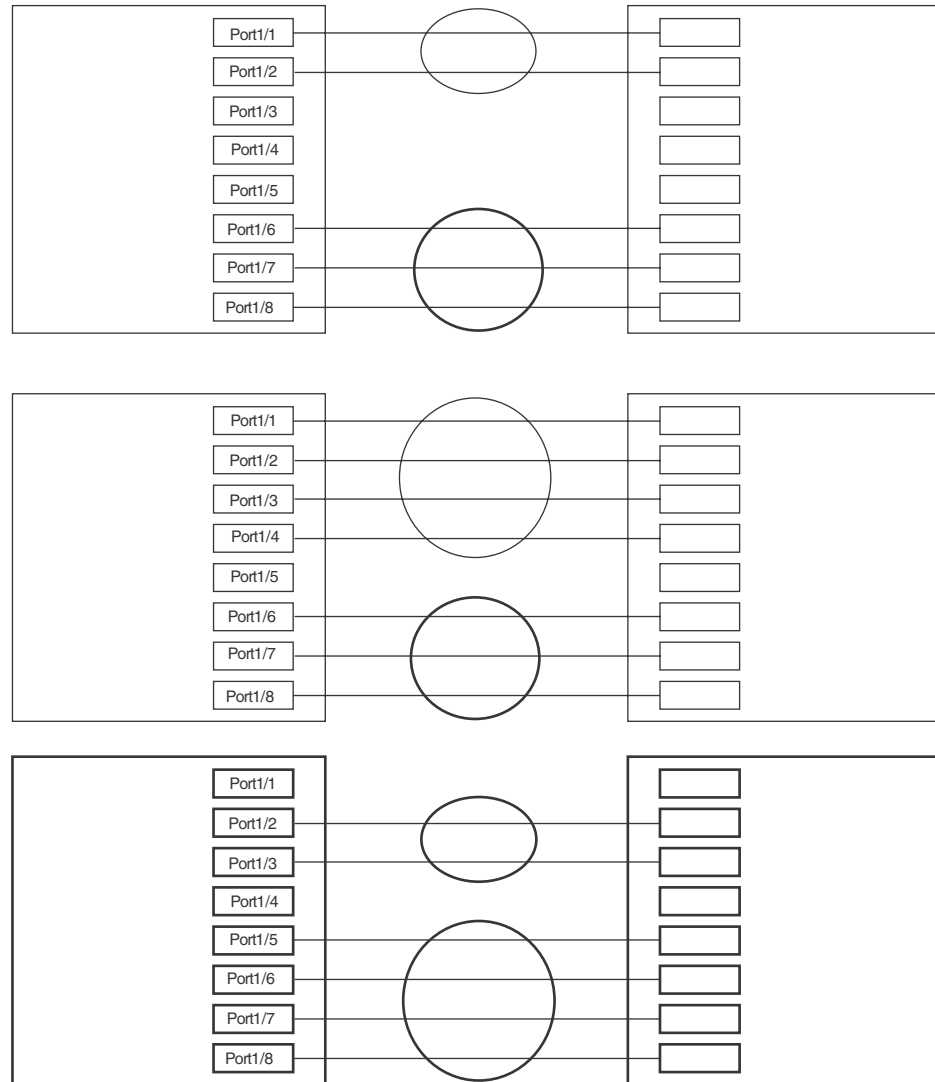
Examples of valid LACP trunk groups

Brocade ports follow the same configuration rules for dynamically created aggregate links as they do for statically configured trunk groups. Refer to [“Trunk group rules”](#) on page 699 and [“Trunk group load sharing”](#) on page 705.

[Figure 67](#) shows some examples of valid aggregate links.

FIGURE 67 Examples of valid aggregate links

Brocade ports enabled for link aggregation follow the same rules as ports configured for trunk groups.



The examples assume that link aggregation is enabled on all of the links between the Brocade device on the left and the device on the right (which can be either a Brocade device or another vendor device). The ports that are members of aggregate links in the examples are following the configuration rules for trunk links on Brocade devices.

The Brocade rules apply to a Brocade device even if the device at the other end is from another vendor and uses different rules. Refer to [“Trunk group rules”](#) on page 699.

Configuration notes and limitations for configuring IronStack LACP trunk groups

This section lists the configuration considerations and limitations for dynamic link aggregation.

FastIron stackable devices

The following notes and feature limitations apply to the FastIron WS and Brocade FCX Series devices:

- The dynamic link aggregation (802.3ad) implementation allows any number of ports up to eight to be aggregated into a link.
- The default key assigned to an aggregate link is based on the port type (1 Gbps port or 10 Gbps port). The device assigns different keys to 10 Gbps ports than to 1 Gbps ports so that ports with different physical capabilities will not be able to form a trunk.

NOTE

The trunks that will be formed by link aggregation will strictly adhere to the static trunking rules on the stackable devices. Be careful in selecting keys if you are manually configuring link aggregation keys. Make sure that the possible trunks that you expect to be formed conform to the static trunking rules.

- When you enable link aggregation (LACP) on a group of Brocade ports, you must also assign a unique key (other than the default key) to all of the ports in the aggregate link.

FastIron stackable devices in an IronStack

- If a stack unit fails, or is removed from the stack, its LACP configuration becomes a reserved configuration on the Active Controller. Any remaining ports of the dynamic trunk in the IronStack continue to function.
- Merging two IronStacks with a dynamic trunk configured between them results in self-looped ports, which are detected and corrected by the Spanning Tree Protocol (STP).
- When an IronStack with dynamic trunks partitions into multiple IronStacks, the protocol will take care of splitting the dynamic trunk in the partner. No user intervention is required.

FastIron X Series devices

The following notes and feature limitations apply to the FastIron X Series devices:

- You cannot use 802.3ad link aggregation on a port configured as a member of a static trunk group.
- The dynamic link aggregation (802.3ad) implementation on FastIron X Series devices allows different numbers of ports to be aggregated in a link, depending on the IP version (IPv6 or IPv4) and the software version running on the device. For details, refer to [Table 132](#) on page 699.
- The default key assigned to an aggregate link is based on the port type (1 Gbps port or 10 Gbps port). The Brocade device assigns different keys to 10 Gbps ports than to 1 Gbps ports so that ports with different physical capabilities will not be able to form a trunk.

NOTE

The trunks that will be formed by link aggregation will strictly adhere to the static trunking rules on the Brocade device. Be careful in selecting keys if you are manually configuring link aggregation keys. Make sure that the possible trunks that you expect to be formed conform to the static trunking rules.

- When the FastIron X Series device dynamically adds or changes a trunk group, the **show trunk** command displays the trunk as both configured and active. However, the **show running-config** or **write terminal** command does not contain a trunk command defining the new or changed trunk group.
- If the FastIron X Series device places a port into a trunk group as a secondary port, all configuration information except information related to link aggregation is removed from the port. For example, if port 1/3 has an IP interface, and the link aggregation feature places port 1/3 into a trunk group consisting of ports 1/1 through 1/4, the IP interface is removed from the port.
- If the FastIron X Series device (Layer 3) is running OSPF or BGP4, the device causes these protocols to reset when a dynamic link change occurs. The reset includes ending and restarting neighbor sessions with OSPF and BGP4 peers, and clearing and relearning dynamic route entries and forwarding cache entries. Although the reset causes a brief interruption, the protocols automatically resume normal operation.
- You can enable link aggregation on 802.1Q tagged ports (ports that belong to more than one port-based VLAN), as well as on untagged ports.

Adaptation to trunk disappearance

The Brocade device will tear down an aggregate link if the device at the other end of the link reboots or brings all the links down. Tearing the aggregate link down prevents a mismatch if the other device has a different trunk configuration following the reboot or re-establishment of the links. Once the other device recovers, dynamic link aggregation can renegotiate the link without a mismatch.

Flexible trunk eligibility

The criteria for trunk port eligibility in an aggregate link are flexible. A range of ports can contain down ports and remain eligible to become an aggregate link.

By default, the device places the ports into 2-port groups, consisting of an odd-numbered port and the next even-numbered port. For example, ports 1/1 and 1/2 are a 2-port group, as are ports 1/3 and 1/4, 9/1 and 9/2, and so on. If either of the ports in a 2-port group is up, the device considers both ports to be eligible to be in an aggregate link.

Figure 68 shows an example of 2-port groups in a range of four ports on which link aggregation is enabled. Based on the states of the ports, some or all of them will be eligible to be used in an aggregate link.

FIGURE 68 2-port groups used to determine aggregation eligibility

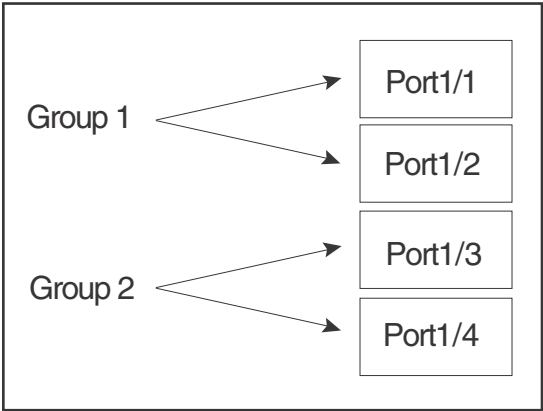


Table 136 shows examples of the ports from Figure 68 that will be eligible for an aggregate link based on individual port states.

TABLE 136 Port eligibility for link aggregation

	Port group 1		Port group 2		Trunk eligibility
	1/1	1/2	1/3	1/4	
Link State	Up	Up	Up	Up	4-port 1/1 – 1/4
	Up	Up	Up	Down	4-port 1/1 – 1/4
	Up	Down	Up	Down	4-port 1/1 – 1/4
	Up	Up	Down	Up	4-port 1/1 – 1/4
	Down	Down	Down	Up	2-port 1/3 – 1/4
	Up	Down	Down	Down	2-port 1/1 – 1/2

As shown in [Table 136](#), all or a subset of the ports within a port range will be eligible for formation into an aggregate link based on port states. Notice that the sets of ports that are eligible for the aggregate link must be valid static trunk configurations.

Enabling dynamic link aggregation

By default, link aggregation is disabled on all ports. To enable link aggregation on a set of ports, enter commands such as the following at the Interface configuration level of the CLI.

NOTE

Configuration commands for link aggregation differ depending on whether you are using the default link aggregation key automatically assigned by the software, or if you are assigning a different, unique key. For more information about keys, refer to [“Key”](#) on page 726.

Using the default key assigned by the software

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e1000-1/1)#link-aggregate active
Brocade(config)#interface ethernet 1/2
Brocade(config-if-e1000-1/2)#link-aggregate active
```

The commands in this example enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages. Note that these ports will use the default key, because one has not been explicitly configured.

NOTE

In conformance with the 802.3ad specification, the default key assigned to an aggregate link is based on the port type (1 Gbps port or 10 Gbps port). The Brocade device assigns different keys to 10 Gbps ports than to 1 Gbps ports so that ports with different physical capabilities will not be able to form a trunk.

Assigning a unique key

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e1000-1/1)#link-aggregate configure key 10000
Brocade(config-if-e1000-1/1)#link-aggregate active
Brocade(config)#interface ethernet 1/2
Brocade(config-if-e1000-1/2)#link-aggregate configure key 10000
Brocade(config-if-e1000-1/2)#link-aggregate active
```

The commands in this example assign the key 10000 and enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages.

NOTE

As shown in the example, when configuring a key, you must assign the key prior to enabling link aggregation.

The following commands enable passive link aggregation on ports 1/5 through 1/8.

```
Brocade(config)#interface ethernet 1/5 to 1/8
Brocade(config-mif-1/5-1/8)#link-aggregate passive
```

The commands in this example enable the passive mode of link aggregation on ports 1/5 through 1/8. These ports wait for the other end of the link to contact them. After this occurs, the ports can send and receive LACPDU messages.

To disable link aggregation on a port, enter a command such as the following.

```
Brocade(config-if-e1000-1/8)#link-aggregate off
```

Syntax: [no] link-aggregate active | passive | off

Syntax: [no] link-aggregate configure [system-priority <num>] | [port-priority <num>] | [key <num>]

NOTE

For more information about keys, including details about the **link-aggregate** and **link-aggregate configure** commands, refer to [“Key”](#) on page 726.

How changing the VLAN membership of a port affects trunk groups and dynamic keys

When you change a port VLAN membership and the port is currently a member of a trunk group, the following changes occur to the trunk group:

- The Brocade device tears down the existing trunk group.
- All ports in the trunk group get a new key.
- The new key group aggregates into a new trunk group.

When you change a port VLAN membership, and the port is not a member of a trunk group, the following changes occur:

- The port gets a new key depending on changes to the port VLAN tag type, as follows:
 - Tagged to Tagged VLAN – The primary port of the trunk group gets a new key.
 - Tagged to Untagged VLAN – The port gets the default key for untagged ports.
 - Untagged to Tagged VLAN – If the Brocade device finds a port with matching port properties, the port gets that port key. If it does not find one, the port gets a new key.
 - Untagged to Untagged VLAN – The port gets a new key depending on whether it is in the default VLAN. If there is a trunk group associated with the key, it is not affected.
- All other ports keep their existing keys.
- The new key groups try to aggregate into trunk groups.

Additional trunking options for LACP trunk ports

Additional trunking options are supported on individual ports that are part of an 802.3ad aggregate link. Refer to [“Additional trunking options”](#) on page 711.

Link aggregation parameters

You can change the settings on individual ports for the following link aggregation parameters:

- System priority
- Port priority
- Timeout
- Key

System priority

The system priority parameter specifies the link aggregation priority on the Brocade device relative to the devices at the other ends of the links on which link aggregation is enabled. A higher value indicates a lower priority. You can specify a priority from 0 through 65535. The default is 1.

NOTE

If you are connecting the Brocade device to another vendor device and the link aggregation feature is not working, set the system priority on the Brocade device to a lower priority (a higher priority value). In some cases, this change allows the link aggregation feature to operate successfully between the two devices.

Port priority

The port priority parameter determines the active and standby links. When a group of ports is negotiating with a group of ports on another device to establish a trunk group, the Brocade port with the highest priority becomes the default active port. The other ports (with lower priorities) become standby ports in the trunk group. You can specify a priority from 0 through 65535. A higher value indicates a lower priority. The default is 1.

NOTE

The primary port in the port group becomes the default active port. The primary port is the lowest-numbered port in a valid trunk port group.

Timeout

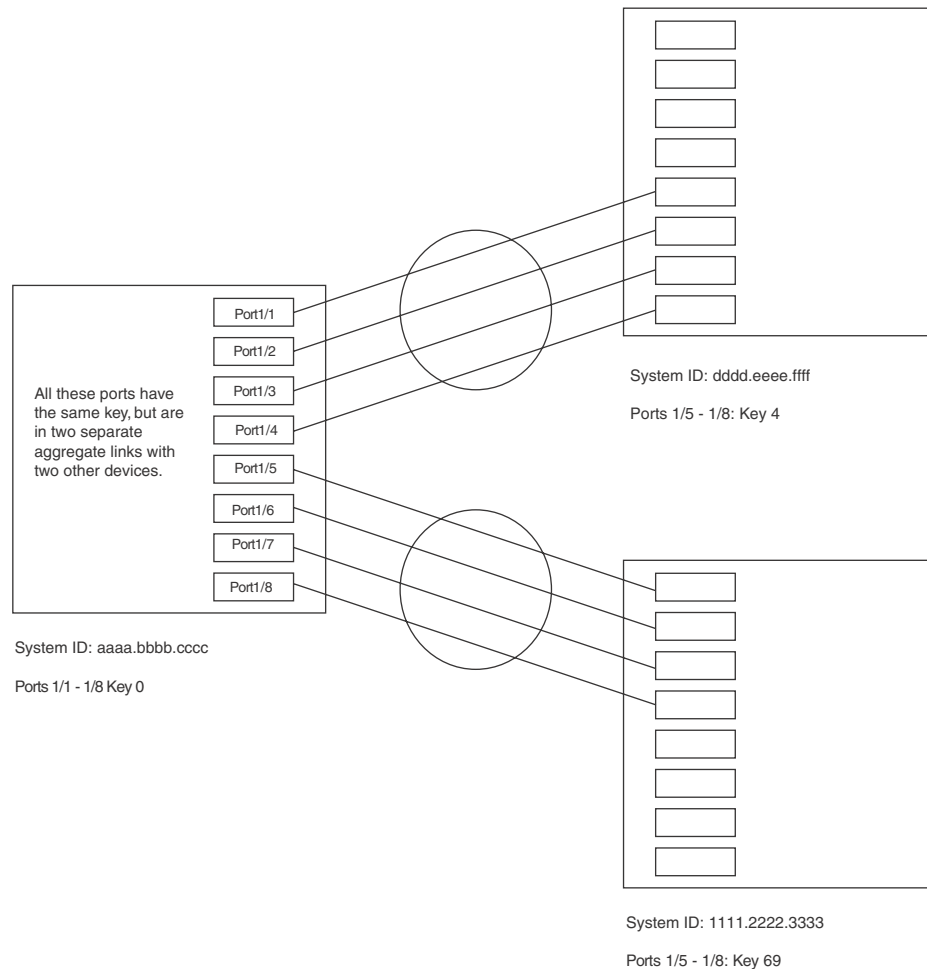
You can specify a timeout mode, which determines how fast ports are removed from a trunk. Refer to [“Configuring link aggregation parameters”](#) on page 729 for more information.

Key

Every port that is link aggregation-enabled has a key. The key identifies the group of potential trunk ports to which the port belongs. Ports with the same key are called a key group and are eligible to be in the same trunk group.

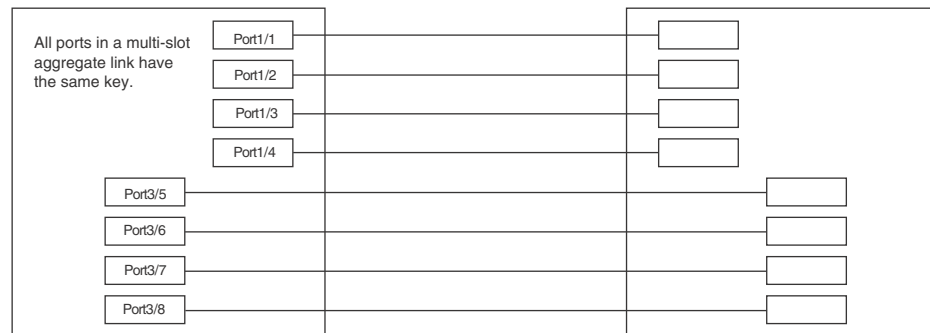
When you enable link aggregation on an untagged port, the software assigns a default key to the port. For tagged ports, you must manually configure link aggregation keys. Refer to [“Configuring keys for ports with link aggregation enabled”](#) on page 729.

All ports within an aggregate link must have the same key. However, if the device has ports that are connected to two different devices, and the port groups allow the ports to form into separate aggregate links with the two devices, then each group of ports can have the same key while belonging to separate aggregate links with different devices. [Figure 69](#) on page 727 shows an example.

FIGURE 69 Ports with the same key in different aggregate links

Notice that the keys between one device and another do not need to match. The only requirement for key matching is that all the ports within an aggregate link on a given device must have the same key.

Devices that support multi-slot trunk groups can form multi-slot aggregate links using link aggregation. However, the link aggregation keys for the groups of ports on each module must match. For example, if you want to allow link aggregation to form an aggregate link containing ports 1/1 through 1/4 and 3/5 through 3/8, you must change the link aggregation key on one or both groups of ports so that the key is the same on all eight ports. [Figure 70](#) on page 728 shows an example of a multi-slot aggregate link.

FIGURE 70 Multi-slot aggregate link

System ID: aaaa.bbbb.cccc

Ports 1/1 - 1/4: Key 0

Ports 3/5 - 3/8: Key 0

By default, the device ports are divided into 4-port groups. The software dynamically assigns a unique key to each 4-port group. If you need to divide a 4-port group into two 2-port groups, change the key in one of the groups so that the two 2-port groups have different keys. For example, if you plan to use ports 1/1 and 1/2 in VLAN 1, and ports 1/3 and 1/4 in VLAN 2, change the key for ports 1/3 and 1/4.

Viewing keys for tagged ports

To display link aggregation information, including the key for a specific port, enter a command such as the following at any level of the CLI.

```
Brocade#show link-aggregate ethernet 1/1
System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp]
1/1      0      0      0   No  L   No  No  No  No  No  No
```

The command in this example shows the key and other link aggregation information for port 1/1.

To display link aggregation information, including the key for all ports on which link aggregation is enabled, enter the following command at any level of the CLI.

```

Brocade#show link-aggregate
System ID: 0004.8055.b200
Long timeout: 90, default: 90
Short timeout: 3, default: 3
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1      1      1    10000  Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
1/2      1      1    10000  Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
2/1      1      1    10000  Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
2/2      1      1    10000  Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
4/1      1      1     480   Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
4/2      1      1     480   Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
4/3      1      1     480   Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
4/4      1      1     480   Yes  S   Agg  Syn  Col  Dis  Def  No  Dwn
4/17     1      1     481   Yes  S   Agg  Syn  Col  Dis  Def  No  Ope
4/18     1      1     481   Yes  S   Agg  Syn  Col  Dis  Def  No  Ope
4/19     1      1     481   Yes  S   Agg  Syn  Col  Dis  Def  No  Ope
4/20     1      1     481   Yes  S   Agg  Syn  Col  Dis  Def  No  Ope

```

Syntax: `show link-aggregate [ethernet <port>]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Configuring link aggregation parameters

You can configure one or more parameters on the same command line, and in any order.

NOTE

For key configuration only, configuration commands differ depending on whether or not link aggregation is enabled on the ports.

Configuring a port group key if link aggregation is disabled

Use the following command sequence to change the key for ports that do not have link aggregation enabled, and for all other link aggregation parameters (for example, system priority, port priority).

For example, to change the software-assigned key for a port group to another value, enter commands similar to the following.

```

Brocade(config)#interface ethernet 1/1 to 1/4
Brocade(config-mif-1/1-1/4)#link-aggregate configure key 10000
Brocade(config-mif-1/1-1/4)#interface ethernet 3/5 to 3/8
Brocade(config-mif-3/5-3/8)#link-aggregate configure key 10000

```

Configuring keys for ports with link aggregation enabled

As shown in the following command sequence, to change the key on ports that already have link aggregation enabled, you must first turn off link aggregation, configure the new key, and then re-enable link aggregation.

```
Brocade(config)#interface ethernet 1/1 to 1/4
Brocade(config-mif-1/1-1/4)#link-aggregate off
Brocade(config-mif-1/1-1/4)#link-aggregate configure key 10000
Brocade(config-mif-1/1-1/4)#link-aggregate active
Brocade(config-mif-1/1-1/4)#interface ethernet 3/5 to 3/8
Brocade(config-mif-3/5-3/8)#link-aggregate off
Brocade(config-mif-3/5-3/8)#link-aggregate configure key 10000
Brocade(config-mif-3/5-3/8)#link-aggregate active
```

These commands change the key for ports 1/1 through 1/4 and 3/5 through 3/8 to 10000. Because all ports in an aggregate link must have the same key, the command in this example enables ports 1/1 through 1/4 and 3/5 through 3/8 to form a multi-slot aggregate link.

Syntax: [no] link-aggregate configure [system-priority <num>] | [port-priority <num>] | [key <num>]

The **system-priority** <num> parameter specifies the Brocade device link aggregation priority. A higher value indicates a lower priority. You can specify a priority from 0 through 65535. The default is 1.

The **port-priority** <num> parameter specifies an individual port priority within the port group. A higher value indicates a lower priority. You can specify a priority from 0 through 65535. The default is 1.

The **key** <num> parameter identifies the group of ports that are eligible to be aggregated into a trunk group. The software automatically assigns a key to each group of ports. The software assigns the keys in ascending numerical order, beginning with 0. You can change a port group key to a value from 10000 through 65535.

Configuring port timeout

You can control the time it takes to remove ports from a trunk with link aggregation enabled by configuring the link aggregated port with a “short” timeout mode. Once a port is configured with a timeout mode, it will remain in that timeout mode whether it is up or down or whether it is part of a trunk.

All ports in a trunk should have the same timeout mode, which is checked when link aggregation is enabled on ports.

To configure a port with a short timeout mode, enter a command such as the following.

```
Brocade(config)#interface ethernet8/1
Brocade(config-if-e1000-8/1)#link-aggregate configure timeout short
```

Syntax: [no] link-aggregate configure timeout [short]

If the timeout mode is not configured for a port and link aggregation is enabled, the port starts with a short timeout mode. Once a trunk is formed, the timeout mode is changed to the long timeout mode. The value for “long” and “short” is displayed in the output for the **show link-aggregate** command.

NOTE

LACP short timeout mode is not supported on FESX combination copper ports. If these ports are configured with a short timeout mode, the ports will flap and will not be aggregated into a trunk group.

Displaying and determining the status of aggregate links

The **show link-aggregate** command provides the ability to view the status of dynamic links. You can determine the status of ports that are members of an aggregate link, and whether LACP messages are being transmitted between the ports.

The following section provides details about the events that can affect the status of ports in an aggregate link and the status of LACP messages exchanged between the ports. Later sections provide instructions for viewing these status reports.

Events that affect the status of ports in an aggregate link

Brocade devices can block traffic on a port or shut down a port that is part of a trunk group or aggregate link when a port joins a trunk group and the port on the other end of the link shuts down or stops transmitting LACP packets. Depending on the timeout value set on the port, the link aggregation information expires. If this occurs, the Brocade device shuts down the port and notifies all the upper layer protocols that the port is down.

Brocade devices can also block traffic on a port that is initially configured with link aggregation. The port is blocked until it joins a trunk group. In this case, traffic is blocked, but the port is still operational.

A port remains blocked until one of the following events occurs:

- Both ports in the aggregate link have the same key.
- LACP brings the port back up.
- The port joins a trunk group.

Displaying link aggregation and port status information

Use the **show link-aggregate** command to determine the operational status of ports associated with aggregate links.

To display the link aggregation information for a specific port, enter a command such as the following at any level of the CLI.

```
Brocade#show link-aggregate ethernet 1/1
System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp] [Ope]
1/1      0      0      0   No   L   No   No   No   No   No   No   Ope
```

The command in this example shows the link aggregation information for port 1/1.

To display the link aggregation information for all ports on which link aggregation is enabled, enter the following command at any level of the CLI.

Displaying and determining the status of aggregate links

```
Brocade#show link-aggregate
System ID: 00e0.52a9.bb00
Long timeout: 120, default: 120 Short timeout: 3, default: 3
Port  [Sys P] [Port P] [Key] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1      1      1      0   No   L   Agg  Syn  No   No   Def  Exp  Ope
1/2      1      1      0   No   L   Agg  Syn  No   No   Def  Exp  Ina
1/3      1      1      0   No   L   Agg  Syn  No   No   Def  Exp  Ina
1/4      1      1      0   No   L   Agg  Syn  No   No   Def  Exp  Blo
1/5      1      1      1   No   L   Agg  No   No   No   Def  Exp  Ope
1/6      1      1      1   No   L   Agg  No   No   No   Def  Exp  Ope
1/7      1      1      1   No   L   Agg  No   No   No   Def  Exp  Dwn
1/8      1      1      1   No   L   Agg  No   No   No   Def  Exp  Dwn
```

Syntax: `show link-aggregate [ethernet <port>]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

NOTE

Ports that are configured as part of an aggregate link must also have the same key. For more information about assigning keys, refer to [“Key”](#) on page 726.

The `show link-aggregate` command shows the following information.

TABLE 137 Description of show link-aggregate command output

Field	Description
System ID	Lists the base MAC address of the device. This is also the MAC address of port 1 (or 1/1).
Short timeout	The short timeout value on the FSX.
Long timeout	The long timeout value on the FSX.
Port	Lists the port number.
Sys P	Lists the system priority configured for this port.
Port P	Lists the port link aggregation priority.
Key	Lists the link aggregation key. This column displays “singleton” if the port is configured with a single instance of LACP. (Refer to “Single instance LACP configuration” on page 734 for more details.)
Act	Indicates the link aggregation mode, which can be one of the following: <ul style="list-style-type: none">• No – The mode is passive or link aggregation is disabled (off) on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link.• Yes – The mode is active. The port can send and receive LACPDU messages.

TABLE 137 Description of show link-aggregate command output (Continued)

Field	Description
Tio	<p>Indicates the timeout value of the port. The timeout value can be one of the following:</p> <ul style="list-style-type: none"> • L – Long. The trunk group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used as confirmation of the health of the aggregate link. • S – Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	<p>Indicates the link aggregation state of the port. The state can be one of the following:</p> <ul style="list-style-type: none"> • Agg – Link aggregation is enabled on the port. • No – Link aggregation is disabled on the port.
Syn	<p>Indicates the synchronization state of the port. The state can be one of the following:</p> <ul style="list-style-type: none"> • No – The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a trunk link. • Syn – The port is in sync with the remote port. The port understands the status of the LACPDU message exchange process, and therefore knows the trunk group to which it belongs, the link aggregation state of the remote port, and so on.
Col	<p>Indicates the collection state of the port, which determines whether the port is ready to send traffic over the trunk link:</p> <ul style="list-style-type: none"> • Col – The port is ready to send traffic over the trunk link. • No – The port is not ready to send traffic over the trunk link.
Dis	<p>Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the trunk link:</p> <ul style="list-style-type: none"> • Dis – The port is ready to receive traffic over the trunk link. • No – The port is not ready to receive traffic over the trunk link.
Def	<p>Indicates whether the port is using default link aggregation values. The port uses default values if it has not received link aggregation information through LACP from the port at the remote end of the link. This field can have one of the following values:</p> <ul style="list-style-type: none"> • Def – The port has not received link aggregation values from the port at the other end of the link and is therefore using its default link aggregation LACP settings. • No – The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	<p>Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values:</p> <ul style="list-style-type: none"> • Exp – The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No – The link aggregation values that this port negotiated with the port at the other end of the link have not expired, so the port is still using the negotiated settings.
Ope	<ul style="list-style-type: none"> • Ope (operational) - The port is operating normally. • Ina (inactive) - The port is inactive because the port on the other side of the link is down or has stopped transmitting LACP packets. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a trunk group. To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.

Displaying link aggregation and port status information for FastIron stackable devices

To display link aggregation information for devices in an IronStack, enter the **show link-aggregate** command. The output for an Ironstack resembles the following.

```
Brocade(config)#show link-aggregate
System ID: 0012.f2e5.a200
Long timeout: 120, default: 120
Short timeout: 3, default: 3
Port [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1/1 1          1      13000 Yes  L   Agg  Syn  Col  Dis  No   No   Ope
2/1/1 1          1      13000 Yes  L   Agg  Syn  Col  Dis  No   No   Ope
3/1/1 1          1      13000 Yes  L   Agg  Syn  Col  Dis  No   No   Ope
4/1/1 1          1      13000 Yes  L   Agg  Syn  Col  Dis  No   No   Ope
```

Displaying LACP status information

Use the **show trunk** command to determine the status of LACP. Refer to [“Displaying trunk group configuration information”](#) on page 716.

Clearing the negotiated aggregate links table

When a group of ports negotiates a trunk group configuration, the software stores the negotiated configuration in a table. You can clear the negotiated link aggregation configurations from the software. When you clear the information, the software does not remove link aggregation parameter settings you have configured. Only the configuration information negotiated using LACP is removed.

NOTE

The software automatically updates the link aggregation configuration based on LACPDU messages. However, clearing the link aggregation information can be useful if you are troubleshooting a configuration.

To clear the link aggregation information, enter the following command at the Privileged EXEC level of the CLI.

```
Brocade#clear link-aggregate
```

Syntax: clear link-aggregate

Single instance LACP configuration

A single instance of link aggregation (or single link LACP) can be used for unidirectional link detection. Single link LACP is based on 802.3ad LACP, but allows you to form an aggregated link with only one Ethernet port. It is the preferred method for detecting unidirectional links across multi-vendor devices, instead of unidirectional link detection (UDLD), because it is based on a standard rather than on a proprietary solution.

Configuration notes for single link LACP

- Single link LACP is supported on 1-GbE and 10-GbE ports, as well as across modules.
- Single link LACP is not supported on static trunk ports.
- Single link LACP is not intended for the creation of trunk groups.
- The single link LACP timer is always short (3 seconds) and is not configurable. PDUs are sent out every three seconds.
- Single link LACP is not supported on ports that have unidirectional link detection (UDLD) configured.

CLI syntax for single link LACP

To form a single link LACP, the port on both sides of the link must have LACP enabled. You can then define a single link LACP at the interface level of the device by entering the following commands.

```
Brocade(config)#interface ethernet 8/1
Brocade(config-if-e1000-8/1)#link-aggregate configure singleton
Link-aggregation active
```

Syntax: [no] link-aggregate configure

When single link LACP is configured, the **show link-aggregate** command displays the following information.

```
Brocade#show link-aggregate
System ID: 00e0.5200.0118
Long timeout: 120, default: 120 Short timeout: 3, default: 3
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
2/1      1      1      1   Yes  S   Agg  Syn  No   No   Def  Exp  Ina
2/2      1      1      1   Yes  S   Agg  Syn  No   No   Def  Exp  Ina
2/3      1      1  singleton Yes  S   Agg  Syn  No   No   Def  Exp  Ina
2/4      1      1  singleton Yes  S   Agg  Syn  No   No   Def  Exp  Dwn
```

If the **singleton** keyword is configured on the port, the “Key” column displays “singleton”. Refer to [“Description of show link-aggregate command output”](#) on page 732 to interpret the information on the displayed output.

Also, when ports are logically brought up or down while the **singleton** keyword is configured on the port, the following syslog messages are generated.

```
Logical link on interface ethernet <slot#/port#> is up.
Logical link on interface ethernet <slot#/port#> is down.
```

Single instance LACP configuration

VLANs

[Table 138](#) lists the individual Brocade FastIron switches and the virtual LAN (VLAN) features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 138 Supported VLAN features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
VLAN Support	Yes	Yes	Yes	Yes	Yes
4096 maximum VLANs	Yes	Yes	Yes	Yes	Yes
802.1Q with tagging	Yes	Yes	Yes	Yes	Yes
802.1ad (Q-in-Q) tagging	Yes	Yes	Yes	Yes	Yes
802.1ad (Q-in-Q) tag profiles	No	Yes	Yes	Yes	Yes
Dual-mode VLANs	Yes	Yes	Yes	Yes	Yes
Port-based VLANs	Yes	Yes	Yes	Yes	Yes
Uplink Ports Within a Port-Based VLAN	Yes	Yes	Yes	Yes	Yes
Protocol VLANs (AppleTalk, IPv4, dynamic IPv6, and IPX)	Yes	Yes	Yes	Yes	Yes
Layer 3 Subnet VLANs (Appletalk, IP subnet network, and IPX)	Yes	Yes	Yes	Yes	Yes
VLAN groups	Yes	Yes	Yes	Yes	Yes
Multi-range VLANs	Yes	Yes	Yes	Yes	Yes
Private VLANs (PVLANS)	Yes	Yes	Yes	Yes	Yes ICX 6450 uses hardware forwarding, while ICX 6430 uses software forwarding.
Super Aggregated VLANs	Yes	Yes	Yes	Yes	Yes
VLAN Q-in-Q Tagging (tag-type 8100 over 8100 encapsulation)	Yes	Yes	Yes	Yes	Yes

VLAN overview

The following sections provide details about the VLAN types and features supported on the FastIron family of switches.

Types of VLANs

This section describes the VLAN types supported on Brocade devices.

VLAN support on FastIron devices

You can configure the following types of VLANs on FastIron devices:

- **Layer 2 port-based VLAN** – a set of physical ports that share a common, exclusive Layer 2 broadcast domain
- **Layer 3 protocol VLANs** – a subset of ports within a port-based VLAN that share a common, exclusive broadcast domain for Layer 3 broadcasts of the specified protocol type
- **IP subnet VLANs** – a subset of ports in a port-based VLAN that share a common, exclusive subnet broadcast domain for a specified IP subnet
- **IPv6 VLANs** – a subset of ports in a port-based VLAN that share a common, exclusive network broadcast domain for IPv6 packets
- **IPX network VLANs** – a subset of ports in a port-based VLAN that share a common, exclusive network broadcast domain for a specified IPX network
- **AppleTalk cable VLANs** – a subset of ports in a port-based-based VLAN that share a common, exclusive network broadcast domain for a specified AppleTalk cable range

When a FastIron device receives a packet on a port that is a member of a VLAN, the device forwards the packet based on the following VLAN hierarchy:

- If the port belongs to an IP subnet VLAN, IPX network VLAN, or AppleTalk cable VLAN and the packet belongs to the corresponding IP subnet, IPX network, or AppleTalk cable range, the device forwards the packet to all the ports within that VLAN.
- If the packet is a Layer 3 packet but cannot be forwarded as described above, but the port is a member of a Layer 3 protocol VLAN for the packet protocol, the device forwards the packet on all the Layer 3 protocol VLAN ports.
- If the packet cannot be forwarded based on either of the VLAN membership types listed above, but the packet can be forwarded at Layer 2, the device forwards the packet on all the ports within the receiving port port-based VLAN.

Protocol VLANs differ from IP subnet, IPX network, and AppleTalk VLANs in an important way. Protocol VLANs accept any broadcast of the specified protocol type. An IP subnet, IPX network, or AppleTalk VLAN accepts only broadcasts for the specified IP subnet, IPX network, or AppleTalk cable range.

NOTE

Protocol VLANs are different from IP subnet, IPX network, and AppleTalk cable VLANs. A port-based VLAN cannot contain both an IP subnet, IPX network, or AppleTalk cable VLAN and a protocol VLAN for the same protocol. For example, a port-based VLAN cannot contain both an IP protocol VLAN and an IP subnet VLAN.

Layer 2 port-based VLANs

On all Brocade devices, you can configure port-based VLANs. A port-based VLAN is a subset of ports on a Brocade device that constitutes a Layer 2 broadcast domain.

By default, all the ports on a Brocade device are members of the default VLAN. Thus, all the ports on the device constitute a single Layer 2 broadcast domain. When you configure a port-based VLAN, the device automatically removes the ports you add to the VLAN from the default VLAN.

You can configure multiple port-based VLANs. You can configure up to 4094 port-based VLANs on a Layer 2 switch or Layer 3 switch. On both device types, valid VLAN IDs are 1 – 4095. You can configure up to the maximum number of VLANs within that ID range.

NOTE

VLAN IDs 4087, 4090, and 4093 are reserved for Brocade internal use only. VLAN 4094 is reserved for use by Single STP. Also, if you are running an earlier release, VLAN IDs 4091 and 4092 may be reserved for Brocade internal use only. If you want to use VLANs 4091 and 4092 as configurable VLANs, you can assign them to different VLAN IDs. For more information, refer to [“Assigning different VLAN IDs to reserved VLANs 4091 and 4092”](#) on page 767.

Each port-based VLAN can contain either tagged or untagged ports. A port cannot be a member of more than one port-based VLAN unless the port is tagged. **802.1Q tagging** allows the port to add a four-byte tag field, which contains the VLAN ID, to each packet sent on the port. You also can configure port-based VLANs that span multiple devices by tagging the ports within the VLAN. The tag enables each device that receives the packet to determine the VLAN the packet belongs to. 802.1Q tagging applies only to Layer 2 VLANs, not to Layer 3 VLANs.

Because each port-based VLAN is a separate Layer 2 broadcast domain, by default each VLAN runs a separate instance of the Spanning Tree Protocol (STP).

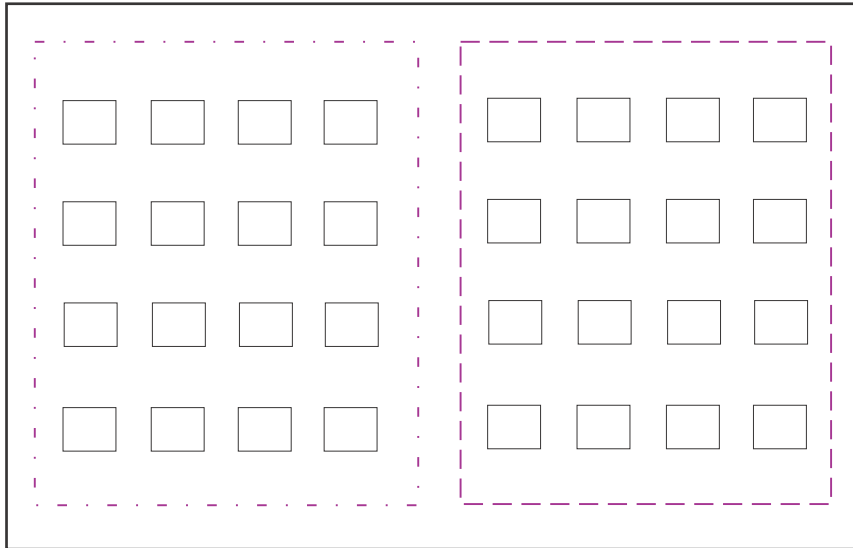
Layer 2 traffic is bridged within a port-based VLAN and Layer 2 broadcasts are sent to all the ports within the VLAN.

[Figure 71](#) shows an example of a Brocade device on which a Layer 2 port-based VLAN has been configured.

FIGURE 71 Brocade device containing user-defined Layer 2 port-based VLAN

DEFAULT-VLAN
VLAN ID = 1
Layer 2 Port-based VLAN

User-configured port-based VLAN



When you add a port-based VLAN, the device removes all the ports in the new VLAN from DEFAULT-VLAN.

Configuring port-based VLANs

Port-based VLANs allow you to provide separate spanning tree protocol (STP) domains or broadcast domains on a port-by-port basis.

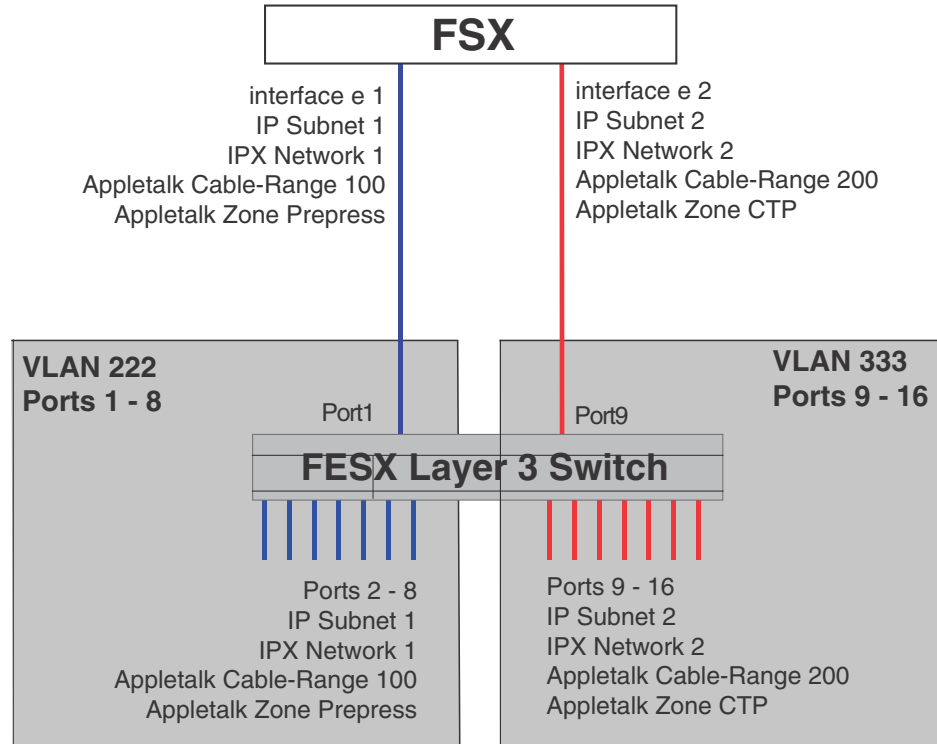
This section describes how to perform the following tasks for port-based VLANs using the CLI:

- Create a VLAN
- Delete a VLAN
- Modify a VLAN
- Change a VLAN priority
- Enable or disable STP on the VLAN

Example 1—Simple port-based VLAN configuration

[Figure 72](#) shows a simple port-based VLAN configuration using a single Brocade Layer 2 switch. All ports within each VLAN are untagged. One untagged port within each VLAN is used to connect the Layer 2 switch to a Layer 3 switch (in this example, a FSX) for Layer 3 connectivity between the two port-based VLANs.

FIGURE 72 Port-based VLANs 222 and 333



To create the two port-based VLANs shown in [Figure 72](#), enter the following commands.

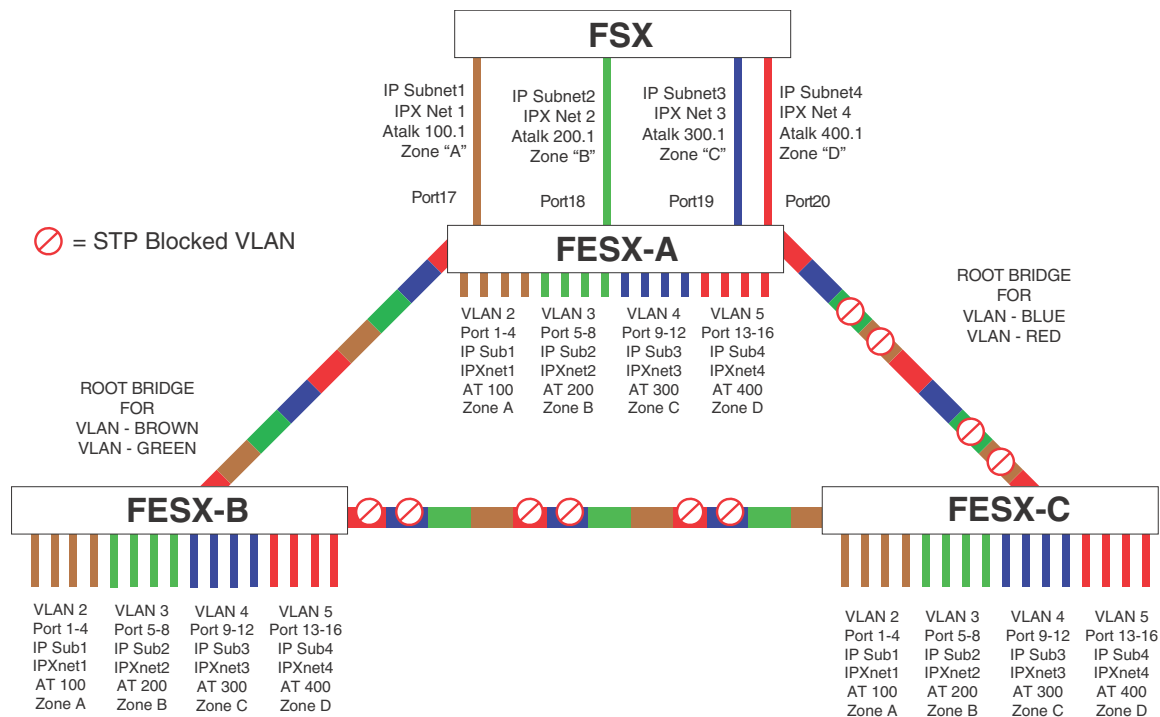
```
Brocade(config)# vlan 222 by port
Brocade(config-vlan-222)# untagged ethernet 1 to 8
Brocade(config-vlan-222)# vlan 333 by port
Brocade(config-vlan-333)# untagged ethernet 9 to 16
```

Syntax: `vlan <vlan-id> by port`

Syntax: `untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]`

Example 2—More complex port-based VLAN configuration

Figure 73 shows a more complex port-based VLAN configuration using multiple Layer 2 switches and IEEE 802.1Q VLAN tagging. The backbone link connecting the three Layer 2 switches is tagged. One untagged port within each port-based VLAN on FESX-A connects each separate network wide Layer 2 broadcast domain to the router for Layer 3 forwarding between broadcast domains. The STP priority is configured to force FESX-A to be the root bridge for VLANs RED and BLUE. The STP priority on FESX-B is configured so that FESX-B is the root bridge for VLANs GREEN and BROWN.

FIGURE 73 More complex port-based VLAN

To configure the Port-based VLANs on the FESX Layer 2 switches in Figure 73, use the following method.

Configuring port-based VLANs on FESX-A

Enter the following commands to configure FESX-A.

```
Brocade> enable
Brocade# configure terminal
Brocade(config)# hostname FESX-A
Brocade-A(config)# vlan 2 name BROWN
Brocade-A(config-vlan-2)# untagged ethernet 1 to 4 ethernet 17
Brocade-A(config-vlan-2)# tagged ethernet 25 to 26
Brocade-A(config-vlan-2)# spanning-tree
Brocade-A(config-vlan-2)# vlan 3 name GREEN
Brocade-A(config-vlan-3)# untagged ethernet 5 to 8 ethernet 18
Brocade-A(config-vlan-3)# tagged ethernet 25 to 26
Brocade-A(config-vlan-3)# spanning-tree
Brocade-A(config-vlan-3)# vlan 4 name BLUE
```

```

Brocade-A(config-vlan-4)# untagged ethernet 9 to 12 ethernet 19
Brocade-A(config-vlan-4)# tagged ethernet 25 to 26
Brocade-A(config-vlan-4)# spanning-tree
Brocade-A(config-vlan-4)# spanning-tree priority 500
Brocade-A(config-vlan-4)# vlan 5 name RED
Brocade-A(config-vlan-5)# untagged ethernet 13 to 16 ethernet 20
Brocade-A(config-vlan-5)# tagged ethernet 25 to 26
Brocade-A(config-vlan-5)# spanning-tree
Brocade-A(config-vlan-5)# spanning-tree priority 500
Brocade-A(config-vlan-5)# end
Brocade-A# write memory

```

Configuring port-based VLANs on FESX-B

Enter the following commands to configure FESX-B.

```

Brocade> en
Brocade# configure terminal
Brocade(config)# hostname FESX-B
Brocade-B(config)# vlan 2 name BROWN
Brocade-B(config-vlan-2)# untagged ethernet 1 to 4
Brocade-B(config-vlan-2)# tagged ethernet 25 to 26
Brocade-B(config-vlan-2)# spanning-tree
Brocade-B(config-vlan-2)# spanning-tree priority 500
Brocade-B(config-vlan-2)# vlan 3 name GREEN
Brocade-B(config-vlan-3)# untagged ethernet 5 to 8
Brocade-B(config-vlan-3)# tagged ethernet 25 to 26
Brocade-B(config-vlan-3)# spanning-tree
Brocade-B(config-vlan-3)# spanning-tree priority 500
Brocade-B(config-vlan-3)# vlan 4 name BLUE
Brocade-B(config-vlan-4)# untagged ethernet 9 to 12
Brocade-B(config-vlan-4)# tagged ethernet 25 to 26
Brocade-B(config-vlan-4)# vlan 5 name RED
Brocade-B(config-vlan-5)# untagged ethernet 13 to 16
Brocade-B(config-vlan-5)# tagged ethernet 25 to 26
Brocade-B(config-vlan-5)# end
Brocade-B# write memory

```

Configuring port-based VLANs on FESX-C

Enter the following commands to configure FESX-C.

```

Brocade> en
Brocade# configure terminal
Brocade(config)# hostname FESX-C
Brocade-C(config)# vlan 2 name BROWN
Brocade-C(config-vlan-2)# untagged ethernet 1 to 4
Brocade-C(config-vlan-2)# tagged ethernet 25 to 26
Brocade-C(config-vlan-2)# vlan 3 name GREEN
Brocade-C(config-vlan-3)# untagged ethernet 5 to 8
Brocade-C(config-vlan-3)# tagged ethernet 25 to 26
Brocade-C(config-vlan-3)# vlan 4 name BLUE
Brocade-C(config-vlan-4)# untagged ethernet 9 to 12
Brocade-C(config-vlan-4)# tagged ethernet 25 to 26
Brocade-C(config-vlan-4)# vlan 5 name RED
Brocade-C(config-vlan-5)# untagged ethernet 13 to 16
Brocade-C(config-vlan-5)# tagged ethernet 25 to 26
Brocade-C(config-vlan-5)# end
Brocade-C# write memory

```

Syntax: `vlan <vlan-id> by port`

Syntax: `untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]`

Syntax: `tagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]`

Syntax: `[no] spanning-tree`

Syntax: `spanning-tree [ethernet [<slotnum>/]<portnum> path-cost <value> priority <value>] forward-delay <value> hello-time <value> maximum-age <time> priority <value>`

Modifying a port-based VLAN

You can make the following modifications to a port-based VLAN:

- Add or delete a VLAN port.
- Enable or disable STP.

Removing a port-based VLAN

Suppose you want to remove VLAN 5 from the example in [Figure 73](#). To do so, use the following procedure.

1. Access the global CONFIG level of the CLI on FESX-A by entering the following commands.

```
Brocade-A> enable
No password has been assigned yet...
Brocade-A# configure terminal
Brocade-A(config)#
```

2. Enter the following command.

```
Brocade-A(config)# no vlan 5
Brocade-A(config)#
```

3. Enter the following commands to exit the CONFIG level and save the configuration to the system-config file on flash memory.

```
Brocade-A(config)#
Brocade-A(config)# end
Brocade-A# write memory
Brocade-A#
```

4. Repeat steps 1 – 3 on FESX-B.

Syntax: `no vlan <vlan-id> by port`

Removing a port from a VLAN

Suppose you want to remove port 11 from VLAN 4 on FESX-A shown in [Figure 73](#). To do so, use the following procedure.

1. Access the global CONFIG level of the CLI on FESX-A by entering the following command.

```
Brocade-A> enable
No password has been assigned yet...
Brocade-A# configure terminal
Brocade-A(config)#
```

2. Access the level of the CLI for configuring port-based VLAN 4 by entering the following command.

```
Brocade-A(config)#
Brocade-A(config)# vlan 4
Brocade-A(config-vlan-4)#
```

3. Enter the following commands.

```
Brocade-A(config-vlan-4)#
Brocade-A(config-vlan-4)# no untagged ethernet 11
deleted port ethe 11 from port-vlan 4.
Brocade-A(config-vlan-4)#
```

4. Enter the following commands to exit the VLAN CONFIG mode and save the configuration to the system-config file on flash memory.

```
Brocade-A(config-vlan-4)#
Brocade-A(config-vlan-4)# end
Brocade-A# write memory
```

You can remove all the ports from a port-based VLAN without losing the rest of the VLAN configuration. However, you cannot configure an IP address on a virtual routing interface unless the VLAN contains ports. If the VLAN has a virtual routing interface, the virtual routing interface IP address is deleted when the ports associated with the interface are deleted. The rest of the VLAN configuration is retained.

Multi-range VLAN

The multi-range VLAN feature allows users to use a single command to create and configure multiple VLANs. These VLANs can be continuous, for example from 2 to 7 or discontinuous, for example, 2 4 7.

NOTE

The maximum number of VLANs you can create or configure with a single command is 64.

Creating a multi-range VLAN

To create more than one VLAN with a single command, you can specify the VLAN number and range.

Syntax: **[no] vlan <num> to <num>**

The <num> parameter specifies the VLAN ID.

To create a continuous range of VLANs, enter command such as the following.

```
Brocade(config)#vlan 2 to 7
Brocade(config-mvlan-2-7)#
```

Syntax: **[no] vlan <num> to <num>**

To create discontinuous VLANs, enter command such as the following.

```
Brocade(config)#vlan 2 4 7
Brocade(config-mvlan-2*7)#exit
```

Syntax: **[no] vlan <num> <num> <num>**

You can also create continuous and discontinuous VLANs. To create continuous and discontinuous VLANs, enter command such as the following.

```
Brocade(config)#vlan 2 to 7 20 25
Brocade(config-mvlan-2*25)#
```

Syntax: [no] vlan <num> to <num> <num>

Deleting a multi-range VLAN

You can also delete multiple VLANs with a single command.

To delete a continuous range of VLANs, enter command such as the following.

```
Brocade(config)#no vlan 2 to 7
```

Syntax: [no] vlan <num> to <num>

To delete discontinuous VLANs, enter command such as the following.

```
Brocade(config)#no vlan 2 4 7
```

Syntax: [no] vlan <num> <num> <num>

You can also delete continuous and discontinuous VLANs. To delete continuous and discontinuous VLANs, enter command such as the following.

```
Brocade(config)#no vlan 2 to 7 20 25
```

Syntax: [no] vlan <num> to <num> <num>

If a single multi-range VLAN command contains more than 64 VLANs, the CLI does not add the VLAN IDs but instead displays an error message. An example is given below.

```
Brocade(config)#vlan 100 to 356
ERROR -can't have more than 64 vlans at a time in a multi-range vlan command
```

Configuring a multi-range VLAN

You can configure multiple VLANs with a single command from the multi-range VLAN configuration level. For example, if you want to add tagged ethernet port 1/1/1 in the VLAN 16 17 20 21 22 23 24, enter the following commands.

```
Brocade(config)#vlan 16 17 20 to 24
Brocade(config-mvlan-16*24)#tag e 1/1/1
Brocade(config-mvlan-16*24)#
```

The first command will take you to the multi-range VLAN configuration mode. The second command will add tagged ethernet port 1/1/1 in the specified VLANs, VLAN 16 17 20 21 22 23 and 24.

The following VLAN parameters can be configured with the specified VLAN range.

Command	Explanation
atalk-protocol	Set AppleTalk protocol VLAN
clear	Clear table/statistics/keys
decnet-protocol	Set decnet protocol VLAN
end	End Configuration level and goto Privileged level
exit	Exit current level
ip-protocol	Set IP protocol VLAN
ipv6-protocol	Set IPv6 protocol VLAN
ipx-protocol	Set IPX protocol VLAN
mac-vlan-permit	Define port to be used for MAC Based VLAN
monitor	Monitor Ingress Traffic on this VLAN(Enable VLAN Mirroring)
multicast	IGMP snooping on this VLAN
netbios-protocol	Set netbios protocol VLAN
no	Undo/disable commands
other-protocol	Set other protocol VLAN
quit	Exit to User level
show	Show system information
spanning-tree	Set spanning tree for this VLAN
static-mac-address	Configure static MAC for this VLAN
tagged	802.1Q tagged port
uplink-switch	Define uplink ports and enable uplink switching
vsrp	Configure VSRP
vsrp-aware	Configure VSRP Aware parameters
write	Write running configuration to flash or terminal

NOTE

In FSX and FESX platforms, the **mac-vlan-permit** command is not available in the multi-range vlan configuration mode.

The VLAN parameters configured for the VLAN range are written in the configuration file of the individual VLANs. These VLAN parameters can also be removed or modified from the individual VLANs. In the following example, as the first step, create VLANs 16 17 20 21 22 23 24. Further, as the second step, add Ethernet port 1/1/1 in all the VLANs. As the third step, enabled 802.1w spanning tree on all these VLANs.

```
Brocade(config)#vlan 16 17 20 to 24
Brocade(config-mvlan-16*24)#tag e 1/1/1
Brocade(config-mvlan-16*24)#
Added tagged port(s) ethe 1/1/1 to port-vlan16.
Added tagged port(s) ethe 1/1/1 to port-vlan 17.
Added tagged port(s) ethe 1/1/1 to port-vlan 20.
Added tagged port(s) ethe 1/1/1 to port-vlan 21.
```

```

Added tagged port(s) ethe 1/1/1 to port-vlan 22.
Added tagged port(s) ethe 1/1/1 to port-vlan 23.
Added tagged port(s) ethe 1/1/1 to port-vlan 24.
Brocade(config-mvlan-16*24)#span 802-1w

```

The Ethernet port e 1/1/1 and spanning tree 802.1w is added to the database of each VLAN separately. You can verify the configuration with the show **running-config** command. See the example below.

```

Brocade(config-mvlan-16*24)#show run
Current configuration:
!
!

```

output omitted

```

!
!
vlan 1 name DEFAULT-VLAN by port
!
vlan 16 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 17 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 20 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 21 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 22 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 23 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 24 by port
tagged ethe 1/1/1
spanning-tree 802-1w

!
!

```

output omitted

```

!
!

```

Now you can modify any one or some of the VLANs. See the example below.

In the following example, disable the spanning tree 802.1w on VLANs 22,23 and 24, And, verify with show **running-config** output that the spanning tree 802.1w is disabled on specified VLANs, VLAN 22, 23 and 24 and not on the VLANs 16, 17, 20 and 21.

```
Brocade(config)#vlan 22 to 24
Brocade(config-mvlan-22-24)#no span 8
Brocade(config-mvlan-22-24)#exit
```

```
Brocade(config)#show run
Current configuration:
```

output omitted

```
!
!
vlan 1 name DEFAULT-VLAN by port
!
vlan 16 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 17 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 20 by port
tagged ethe 1/1/1
spanning-tree 802-1w
!
vlan 21 by port
tagged ethe 1/1/1
spanning-tree 802-1w

vlan 22 by port
tagged ethe 1/1/1
!
vlan 23 by port
tagged ethe 1/1/1
!
vlan 24 by port
tagged ethe 1/1/1
```

output omitted

Multi-range VLAN show commands

This section describes the show commands for multi-range VLAN parameters.

In the multi-range VLAN mode, some of the Show commands are also available. The output of the Show commands in multi-range VLAN mode displays the information related to the specific VLANs only. See the example below.

In the following example, the first command will change the interface configuration level to the multi-range VLAN mode for the VLANs 4, 5 and 6. In the multi-range VLAN mode, enter the command *show 802.1w*. The output will display the information of STP for VLANs 4, 5 and 6

```
Brocade(config)#vlan 4 to 6
```

VLAN overview

```
Brocade(config-mvlan-4-6)#show 802-1w
```

```
--- VLAN 4 [ STP Instance owned by VLAN 4 ] -----
```

```
Bridge IEEE 802.1W Parameters:
```

Bridge Identifier	Bridge MaxAge	Bridge Hello	Bridge FwdDly	Bridge Force Version	tx Hold
hex	sec	sec	sec		cnt
8000002022227700	20	2	15	Default	3

RootBridge Identifier	RootPath Cost	DesignatedBridge Identifier	Root Port	Max Age	Fwd Dly	Hel lo
hex		hex		sec	sec	sec
8000002022227700	0	8000002022227700	Root	20	15	2

```
Port IEEE 802.1W Parameters:
```

```

      <--- Config Params --><----- Current state ----->
Port   Pri PortPath P2P Edge Role      State      Designa- Designated
Num    Cost      Mac Port      DESIGNATED FORWARDING 0      ted cost  bridge
1/1/1  128 20000    F   F        0      8000002022227700

```

```
--- VLAN 5 [ STP Instance owned by VLAN 5 ] -----
```

```
Bridge IEEE 802.1W Parameters:
```

Bridge Identifier	Bridge MaxAge	Bridge Hello	Bridge FwdDly	Bridge Force Version	tx Hold
hex	sec	sec	sec		cnt
8000002022227700	20	2	15	Default	3

RootBridge Identifier	RootPath Cost	DesignatedBridge Identifier	Root Port	Max Age	Fwd Dly	Hel lo
hex		hex		sec	sec	sec
8000002022227700	0	8000002022227700	Root	20	15	2

```
Port IEEE 802.1W Parameters:
```

```

      <--- Config Params --><----- Current state ----->
Port   Pri PortPath P2P Edge Role      State      Designa- Designated
Num    Cost      Mac Port      DESIGNATED FORWARDING 0      ted cost  bridge
1/1/1  128 20000    F   F        0      8000002022227700

```

```
--- VLAN 6 [ STP Instance owned by VLAN 6 ] -----
```

```
Bridge IEEE 802.1W Parameters:
```

Bridge Identifier	Bridge MaxAge	Bridge Hello	Bridge FwdDly	Bridge Force Version	tx Hold
hex	sec	sec	sec		cnt
8000002022227700	20	2	15	Default	3

RootBridge Identifier	RootPath Cost	DesignatedBridge Identifier	Root Port	Max Age	Fwd Dly	Hel lo
hex		hex		sec	sec	sec
8000002022227700	0	8000002022227700	Root	20	15	2

```
Port IEEE 802.1W Parameters:
```

```

          <--- Config Params ---><----- Current state ----->
Port    Pri PortPath P2P Edge Role      State      Designa-  Designated
Num     Cost      Mac Port      State      ted cost   bridge
1/1/1  128 20000    F   F    DESIGNATED FORWARDING 0      8000002022227700

```

The following **show** parameters can be viewed for the specified VLAN range from the multi-range VLAN configuration mode. The output of these commands displays information about the specified VLANs only.

TABLE 139 VLAN show parameters

Command	Definition
802-1w	Rapid Spanning tree IEEE 802.1w status
mac-address	MAC address table
span	Spanning tree status
vlan	VLAN status
vsrp	Show VSRP commands

Layer 3 protocol-based VLANs

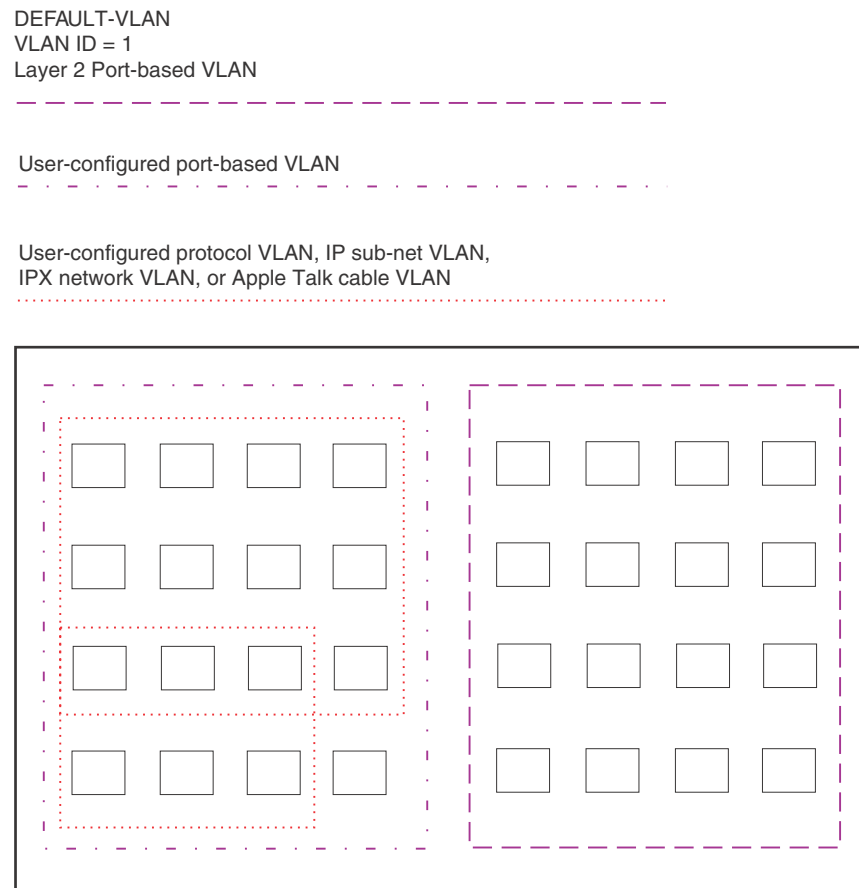
If you want some or all of the ports within a port-based VLAN to be organized according to Layer 3 protocol, you must configure a Layer 3 protocol-based VLAN within the port-based VLAN.

You can configure each of the following types of protocol-based VLAN within a port-based VLAN. All the ports in the Layer 3 VLAN must be in the same Layer 2 VLAN.

Layer 3 protocol-based VLANs are as follows:

- **AppleTalk** – The device sends AppleTalk broadcasts to all ports within the AppleTalk protocol VLAN.
- **IP** – The device sends IP broadcasts to all ports within the IP protocol VLAN.
- **IPv6** – The device sends IPv6 broadcasts to all ports within the IPv6 protocol VLAN.
- **IPX** – The device sends IPX broadcasts to all ports within the IPX protocol VLAN.
- **DECnet** – The device sends DECnet broadcasts to all ports within the DECnet protocol VLAN.
- **NetBIOS** – The device sends NetBIOS broadcasts to all ports within the NetBIOS protocol VLAN.
- **Other** – The device sends broadcasts for all protocol types other than those listed above to all ports within the VLAN.

Figure 74 shows an example of Layer 3 protocol VLANs configured within a Layer 2 port-based VLAN.

FIGURE 74 Layer 3 protocol VLANs within a Layer 2 port-based VLAN

You can add Layer 3 protocol VLANs or IP sub-net, IPX network, and AppleTalk cable VLANs to port-based VLANs.

Layer 3 VLANs cannot span Layer 2 port-based VLANs.

However, Layer 3 VLANs can overlap within a Layer 2 port-based VLAN.

Integrated Switch Routing (ISR)

The Brocade ***Integrated Switch Routing (ISR)*** feature enables VLANs configured on Layer 3 switches to route Layer 3 traffic from one protocol VLAN or IP subnet, IPX network, or AppleTalk cable VLAN to another. Normally, to route traffic from one IP subnet, IPX network, or AppleTalk cable VLAN to another, you would need to forward the traffic to an external router. The VLANs provide Layer 3 broadcast domains for these protocols but do not in themselves provide routing services for these protocols. This is true even if the source and destination IP subnets, IPX networks, or AppleTalk cable ranges are on the same device.

ISR eliminates the need for an external router by allowing you to route between VLANs using virtual routing interfaces (vifs). A **virtual routing interface** is a logical port on which you can configure Layer 3 routing parameters. You configure a separate virtual routing interface on each VLAN that you want to be able to route from or to. For example, if you configure two IP subnet VLANs on a Layer 3 switch, you can configure a virtual routing interface on each VLAN, then configure IP routing parameters for the subnets. Thus, the Layer 3 switch forwards IP subnet broadcasts within each VLAN at Layer 2 but routes Layer 3 traffic between the VLANs using the virtual routing interfaces.

NOTE

The Layer 3 switch uses the lowest MAC address on the device (the MAC address of port 1 or 1/1) as the MAC address for all ports within all virtual routing interfaces you configure on the device.

The routing parameters and the syntax for configuring them are the same as when you configure a physical interface for routing. The logical interface allows the Layer 3 switch to internally route traffic between the protocol-based VLANs without using physical interfaces.

All the ports within a protocol-based VLAN must be in the same port-based VLAN. The protocol-based VLAN cannot have ports in multiple port-based VLANs, unless the ports in the port-based VLAN to which you add the protocol-based VLAN are 802.1Q tagged.

You can configure multiple protocol-based VLANs within the same port-based VLAN. In addition, a port within a port-based VLAN can belong to multiple protocol-based VLANs of the same type or different types. For example, if you have a port-based VLAN that contains ports 1 – 10, you can configure port 5 as a member of an AppleTalk protocol VLAN, an IP protocol VLAN, and an IPX protocol VLAN, and so on.

IP subnet, IPX network, and AppleTalk cable VLANs

The protocol-based VLANs described in the previous section provide separate protocol broadcast domains for specific protocols. For IP, IPX, and AppleTalk, you can provide more granular broadcast control by instead creating the following types of VLAN:

- **IP subnet VLAN** – An IP subnet broadcast domain for a specific IP subnet.
- **IPX network VLAN** – An IPX network broadcast domain for a specific IPX network.
- **AppleTalk cable VLAN** – An AppleTalk broadcast domain for a specific cable range.

You can configure these types of VLANs on Layer 3 switches only. The Layer 3 switch sends broadcasts for the IP subnet, IPX network, or AppleTalk cable range to all ports within the IP subnet, IPX network, or AppleTalk cable VLAN at Layer 2.

The Layer 3 switch routes packets between VLANs at Layer 3. To configure an IP subnet, IPX network, or AppleTalk cable VLAN to route, you must add a virtual routing interface to the VLAN, then configure the appropriate routing parameters on the virtual routing interface.

NOTE

The Layer 3 switch routes packets between VLANs of the same protocol. The Layer 3 switch cannot route from one protocol to another.

NOTE
IP subnet VLANs are not the same thing as IP protocol VLANs. An IP protocol VLAN sends all IP broadcasts on the ports within the IP protocol VLAN. An IP subnet VLAN sends only the IP subnet broadcasts for the subnet of the VLAN. You cannot configure an IP protocol VLAN and an IP subnet VLAN within the same port-based VLAN.

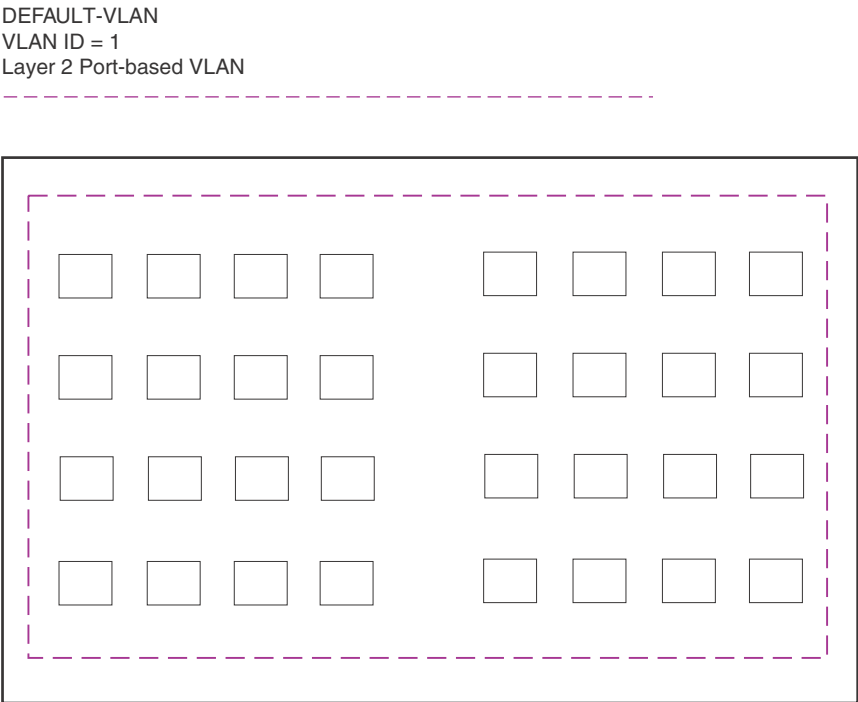
This note also applies to IPX protocol VLANs and IPX network VLANs, and to AppleTalk protocol VLANs and AppleTalk cable VLANs.

Default VLAN

By default, all the ports on a FastIron device are in a single port-based VLAN. This VLAN is called the DEFAULT-VLAN and is VLAN number 1. FastIron devices do not contain any protocol VLANs or IP subnet, IPX network, or AppleTalk cable VLANs by default.

Figure 75 shows an example of the default Layer 2 port-based VLAN.

FIGURE 75 Default Layer 2 port-based VLAN



By default, all ports belong to a single port-based VLAN, DEFAULT-VLAN. Thus, all ports belong to a single Layer 2 broadcast domain.

When you configure a port-based VLAN, one of the configuration items you provide is the ports that are in the VLAN. When you configure the VLAN, the Brocade device automatically removes the ports that you place in the VLAN from DEFAULT-VLAN. By removing the ports from the default VLAN, the Brocade device ensures that each port resides in only one Layer 2 broadcast domain.

NOTE

Information for the default VLAN is available only after you define another VLAN.

Some network configurations may require that a port be able to reside in two or more Layer 2 broadcast domains (port-based VLANs). In this case, you can enable a port to reside in multiple port-based VLANs by tagging the port. Refer to the following section.

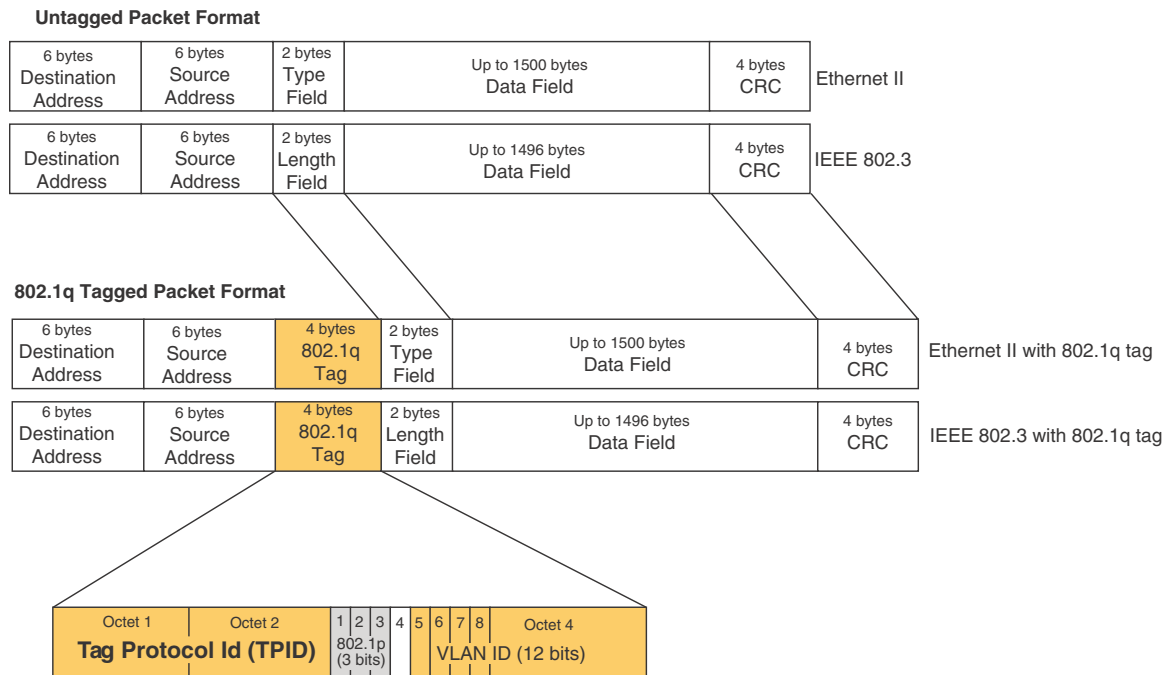
If your network requires that you use VLAN ID 1 for a user-configured VLAN, you can reassign the default VLAN to another valid VLAN ID. Refer to [“Assigning a different VLAN ID to the default VLAN”](#) on page 766.

802.1Q tagging

802.1Q tagging is an IEEE standard that allows a networking device to add information to a Layer 2 packet in order to identify the VLAN membership of the packet. Brocade devices tag a packet by adding a four-byte tag to the packet. The tag contains the tag value, which identifies the data as a tag, and also contains the VLAN ID of the VLAN from which the packet is sent.

- The default tag value is 8100 (hexadecimal). This value comes from the 802.1Q specification. You can change this tag value on a global basis on Brocade devices if needed to be compatible with other vendors' equipment.
- The VLAN ID is determined by the VLAN on which the packet is being forwarded.

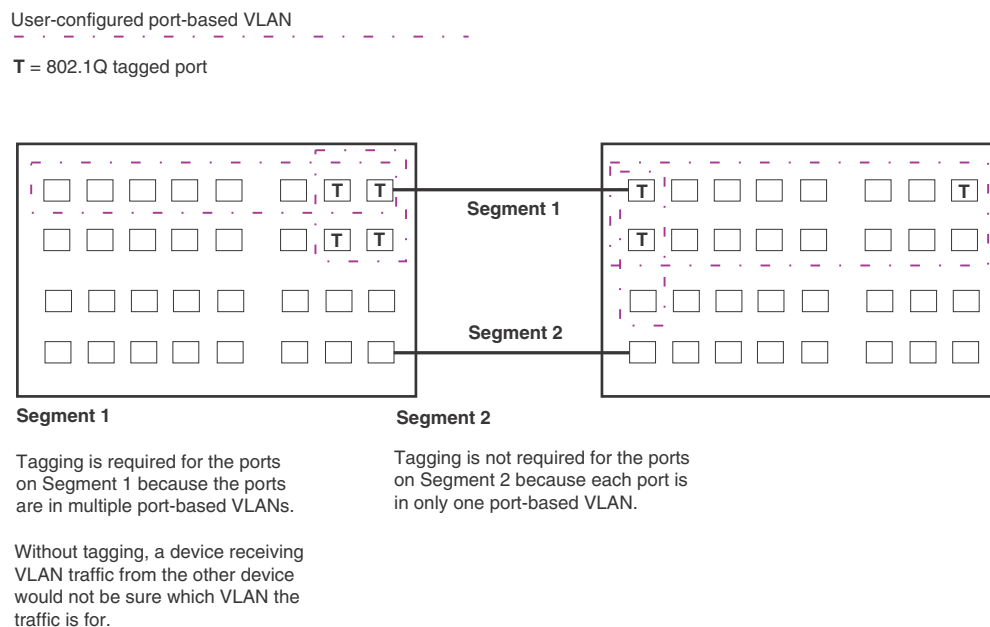
[Figure 76](#) shows the format of packets with and without the 802.1Q tag. The tag format is vendor-specific. To use the tag for VLANs configured across multiple devices, make sure all the devices support the same tag format.

FIGURE 76 Packet containing a Brocade 802.1Q VLAN tag

If you configure a VLAN that spans multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If a port connecting one device to the other is a member of only a single port-based VLAN, tagging is not required.

If you use tagging on multiple devices, each device must be configured for tagging and must use the same tag value. In addition, the implementation of tagging must be compatible on the devices. The tagging on all Brocade devices is compatible with other Brocade devices.

Figure 77 shows an example of two devices that have the same Layer 2 port-based VLANs configured across them. Notice that only one of the VLANs requires tagging.

FIGURE 77 VLANs configured across multiple devices

Support for 802.1ad (Q-in-Q) tagging

Brocade devices provide finer granularity for configuring 802.1Q tagging, enabling you to configure 802.1Q tag-types on a group of ports, thereby enabling the creation of two identical 802.1Q tags (802.1ad tagging) on a single device. This enhancement improves SAV interoperability between Brocade devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

- FastIron X Series devices support one value for **tag-type**, which is defined at the global level.
- FastIron WS and Brocade FCX Series devices support one value for **tag-type**, which is defined at the global level, and one value for **tag-profile**, which is defined at the global and interface level of the CLI.

802.1ad tagging for FastIron WS and Brocade FCX Series devices

The following enhancements allow the FastIron WS and Brocade FCX Series devices, including those in an IronStack, to use Q-in-Q and SAV, by allowing the changing of a tag profile for ports:

- In addition to the default tag type 0x8100, you can now configure one additional global tag profile with a number from 0xffff.
- Tag profiles on a single port, or a group of ports can be configured to point to the global tag profile.

For example applications and configuration details, refer to [“802.1ad tagging configuration”](#) on page 804.

To configure a global tag profile, enter the following command in the configuration mode.

```
Brocade(config)# tag-profile 9500
```

Syntax: [no] tag-profile <tag-no>

<tag-no> - the number of the tag, can be 0x8100 (default), or 0xffff

To direct individual ports or on a range of ports to this tag profile, enter commands similar to the following.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e1000-1/1/1)# tag-profile enable
Brocade(config-mif-1/1/1,1/2/1)# tag-profile enable
```

Spanning Tree Protocol (STP)

The default state of STP depends on the device type:

- STP is disabled by default on Brocade Layer 3 switches.
- STP is enabled by default on Brocade Layer 2 switches.

Also by default, each port-based VLAN has a separate instance of STP. Thus, when STP is globally enabled, each port-based VLAN on the device runs a separate spanning tree.

You can enable or disable STP on the following levels:

- **Globally** – Affects all ports on the device.

NOTE

If you configure a port-based VLAN on the device, the VLAN has the same STP state as the default STP state on the device. Thus, on Layer 2 switches, new VLANs have STP enabled by default. On Layer 3 switches, new VLANs have STP disabled by default. You can enable or disable STP in each VLAN separately. In addition, you can enable or disable STP on individual ports.

- **Port-based VLAN** – Affects all ports within the specified port-based VLAN.

STP is a Layer 2 protocol. Thus, you cannot enable or disable STP for individual protocol VLANs or for IP subnet, IPX network, or AppleTalk cable VLANs. The STP state of a port-based VLAN containing these other types of VLANs determines the STP state for all the Layer 2 broadcasts within the port-based VLAN. This is true even though Layer 3 protocol broadcasts are sent on Layer 2 within the VLAN.

It is possible that STP will block one or more ports in a protocol VLAN that uses a virtual routing interface to route to other VLANs. For IP protocol and IP subnet VLANs, even though some of the physical ports of the virtual routing interface are blocked, the virtual routing interface can still route so long as at least one port in the virtual routing interface protocol VLAN is not blocked by STP.

If you enable Single STP (SSTP) on the device, the ports in all VLANs on which STP is enabled become members of a single spanning tree. The ports in VLANs on which STP is disabled are excluded from the single spanning tree.

For more information, refer to [Chapter 27, “Spanning Tree Protocol”](#).

Virtual routing interfaces

A virtual routing interface is a logical routing interface that Brocade Layer 3 switches use to route Layer 3 protocol traffic between protocol VLANs.

Brocade devices send Layer 3 traffic at Layer 2 within a protocol VLAN. However, Layer 3 traffic from one protocol VLAN to another must be routed.

If you want the device to be able to send Layer 3 traffic from one protocol VLAN to another, you must configure a virtual routing interface on each protocol VLAN, then configure routing parameters on the virtual routing interfaces. For example, to enable a Layer 3 switch to route IP traffic from one IP subnet VLAN to another, you must configure a virtual routing interface on each IP subnet VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

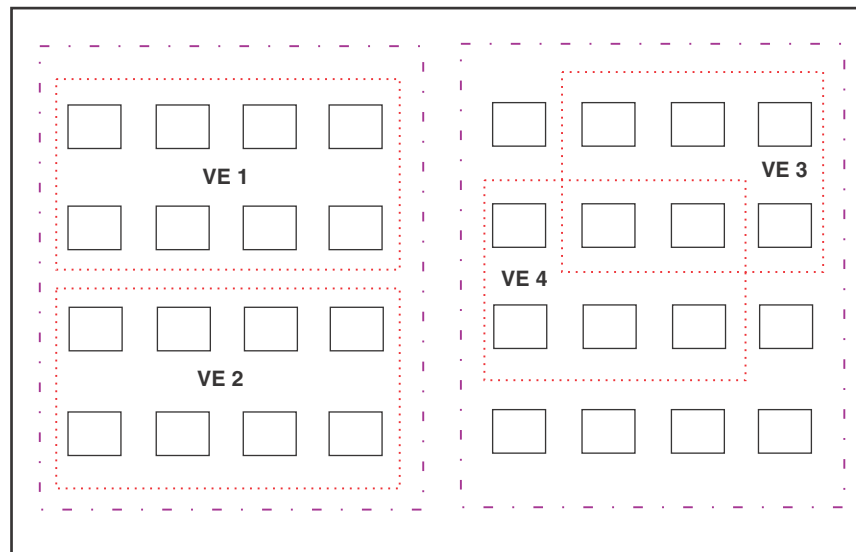
[Figure 78](#) shows an example of Layer 3 protocol VLANs that use virtual routing interfaces for routing.

FIGURE 78 Use virtual routing interfaces for routing between Layer 3 protocol VLANs

User-configured port-based VLAN

User-configured protocol VLAN, IP sub-net VLAN,
IPX network VLAN, or AppleTalk cable VLAN

VE = virtual interface
("VE" stands for "Virtual Ethernet")



Layer 2 and Layer 3 traffic within a VLAN
is bridged at Layer 2.

Layer 3 traffic between protocol VLANs
is routed using virtual interfaces (VE).
To route to one another, each protocol
VLAN must have a virtual interface.

VLAN and virtual routing interface groups

Brocade FastIron devices support the configuration of VLAN groups. To simplify configuration, you can configure VLAN groups and virtual routing interface groups. When you create a VLAN group, the VLAN parameters you configure for the group apply to all the VLANs within the group. Additionally, you can easily associate the same IP subnet interface with all the VLANs in a group by configuring a virtual routing interface group with the same ID as the VLAN group.

For configuration information, refer to [“VLAN groups and virtual routing interface group”](#) on page 791.

Dynamic, static, and excluded port membership

When you add ports to a protocol VLAN, IP subnet VLAN, IPX network VLAN, or AppleTalk cable VLAN, you can add them dynamically or statically:

- Dynamic ports
- Static ports

You also can explicitly exclude ports.

Dynamic ports

Dynamic ports are added to a VLAN when you create the VLAN. However, if a dynamically added port does not receive any traffic for the VLAN protocol within ten minutes, the port is removed from the VLAN. However, the port remains a candidate for port membership. Thus, if the port receives traffic for the VLAN protocol, the device adds the port back to the VLAN.

After the port is added back to the VLAN, the port can remain an active member of the VLAN up to 20 minutes without receiving traffic for the VLAN protocol. If the port ages out, it remains a candidate for VLAN membership and is added back to the VLAN when the VLAN receives protocol traffic. At this point, the port can remain in the VLAN up to 20 minutes without receiving traffic for the VLAN protocol, and so on.

Unless you explicitly add a port statically or exclude a port, the port is a dynamic port and thus can be an active member of the VLAN, depending on the traffic it receives.

NOTE

You cannot configure dynamic ports in an AppleTalk cable VLAN. The ports in an AppleTalk cable VLAN must be static. However, ports in an AppleTalk protocol VLAN can be dynamic or static.

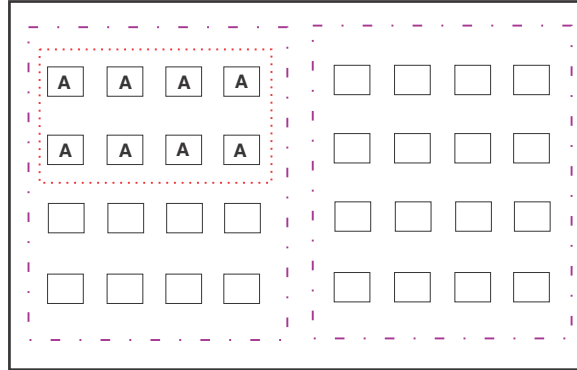
[Figure 79](#) shows an example of a VLAN with dynamic ports. Dynamic ports not only join and leave the VLAN according to traffic, but also allow some broadcast packets of the specific protocol to “leak” through the VLAN. Refer to [“Broadcast leaks”](#) on page 763.

FIGURE 79 VLAN with dynamic ports—all ports are active when you create the VLAN

A = active port

C = candidate port

When you add ports dynamically, all the ports are added when you add the VLAN.

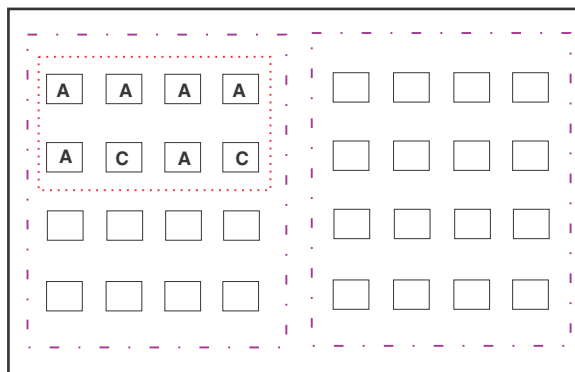


SUBNET Ports in a new protocol VLAN that do not receive traffic for the VLAN protocol age out after 10 minutes and become candidate ports. [Figure 80](#) shows what happens if a candidate port receives traffic for the VLAN protocol.

FIGURE 80 VLAN with dynamic ports—candidate ports become active again if they receive protocol traffic

Ports that time out remain candidates for membership in the VLAN and become active again if they receive traffic for the VLAN's protocol, IP sub-net, IPX network, or AppleTalk cable range.

When a candidate port rejoins a VLAN, the timeout for that port becomes 20 minutes. Thus, the port remains an active member of the VLAN even if it does not receive traffic for 20 minutes. After that, the port becomes a candidate port again.



Static ports

Static ports are permanent members of the protocol VLAN. The ports remain active members of the VLAN regardless of whether the ports receive traffic for the VLAN protocol. You must explicitly identify the port as a static port when you add it to the VLAN. Otherwise, the port is dynamic and is subject to aging out.

Excluded ports

If you want to prevent a port in a port-based VLAN from ever becoming a member of a protocol, IP subnet, IPX network, or AppleTalk cable VLAN configured in the port-based VLAN, you can explicitly exclude the port. You exclude the port when you configure the protocol, IP subnet, IPX network, or AppleTalk cable VLAN.

Excluded ports do not leak broadcast packets. Refer to [“Broadcast leaks”](#) on page 763.

Broadcast leaks

A dynamic port becomes a member of a Layer 3 protocol VLAN when traffic from the VLAN's protocol is received on the port. After this point, the port remains an active member of the protocol VLAN, unless the port does not receive traffic from the VLAN's protocol for 20 minutes. If the port does not receive traffic for the VLAN's protocol for 20 minutes, the port ages out and is no longer an active member of the VLAN.

To enable a host that has been silent for awhile to send and receive packets, the dynamic ports that are currently members of the Layer 3 protocol VLAN "leak" Layer 3 broadcast packets to the ports that have aged out. When a host connected to one of the aged out ports responds to a leaked broadcast, the port is added to the protocol VLAN again.

To "leak" Layer 3 broadcast traffic, an active port sends 1/8th of the Layer 3 broadcast traffic to the inactive (aged out) ports.

Static ports do not age out and do not leak broadcast packets.

Super aggregated VLANs

Brocade FastIron devices support Super Aggregated VLANs. You can aggregate multiple VLANs within another VLAN. This feature allows you to construct Layer 2 paths and channels. This feature is particularly useful for Virtual Private Network (VPN) applications in which you need to provide a private, dedicated Ethernet connection for an individual client to transparently reach its subnet across multiple networks.

For an application example and configuration information, refer to [“Super-aggregated VLAN configuration”](#) on page 796.

Trunk group ports and VLAN membership

A trunk group is a set of physical ports that are configured to act as a single physical interface. Each trunk group port configuration is based on the configuration of the lead port, which is the lowest numbered port in the group.

If you add a trunk group lead port to a VLAN, all of the ports in the trunk group become members of that VLAN.

Summary of VLAN configuration rules

A hierarchy of VLANs exists between the Layer 2 and Layer 3 protocol-based VLANs:

- Port-based VLANs are at the lowest level of the hierarchy.
- Layer 3 protocol-based VLANs, IP, IPv6, IPX, AppleTalk, Decnet, and NetBIOS are at the middle level of the hierarchy.
- IP subnet, IPX network, and AppleTalk cable VLANs are at the top of the hierarchy.

NOTE

You cannot have a protocol-based VLAN and a subnet or network VLAN of the same protocol type in the same port-based VLAN. For example, you can have an IPX protocol VLAN and IP subnet VLAN in the same port-based VLAN, but you cannot have an IP protocol VLAN and an IP subnet VLAN in the same port-based VLAN, nor can you have an IPX protocol VLAN and an IPX network VLAN in the same port-based VLAN.

As a Brocade device receives packets, the VLAN classification starts from the highest level VLAN first. Therefore, if an interface is configured as a member of both a port-based VLAN and an IP protocol VLAN, IP packets coming into the interface are classified as members of the IP protocol VLAN because that VLAN is higher in the VLAN hierarchy.

Multiple VLAN membership rules

- A port can belong to multiple, unique, overlapping Layer 3 protocol-based VLANs without VLAN tagging.
- A port can belong to multiple, overlapping Layer 2 port-based VLANs only if the port is a tagged port. Packets sent out of a tagged port use an 802.1Q-tagged frame.
- When both port and protocol-based VLANs are configured on a given device, all protocol VLANs must be strictly contained within a port-based VLAN. A protocol VLAN cannot include ports from multiple port-based VLANs. This rule is required to ensure that port-based VLANs remain loop-free Layer 2 broadcast domains.
- IP protocol VLANs and IP subnet VLANs cannot operate concurrently on the system or within the same port-based VLAN.
- IPX protocol VLANs and IPX network VLANs cannot operate concurrently on the system or within the same port-based VLAN.
- If you first configure IP and IPX protocol VLANs before deciding to partition the network by IP subnet and IPX network VLANs, then you need to delete those VLANs before creating the IP subnet and IPX network VLANs.
- One of each type of protocol VLAN is configurable within each port-based VLAN on the Layer 2 switch.
- Multiple IP subnet and IPX network VLANs are configurable within each port-based VLAN on the Layer 2 switch.
- Removing a configured port-based VLAN from a Brocade Layer 2 switch or Layer 3 switch automatically removes any protocol-based VLAN, IP subnet VLAN, AppleTalk cable VLAN, or IPX network VLAN, or any Virtual Ethernet router interfaces defined within the Port-based VLAN.

Routing between VLANs

Brocade Layer 3 switches can locally route IP, IPX, and Appletalk between VLANs defined within a single router. All other routable protocols or protocol VLANs (for example, DecNet) must be routed by another external router capable of routing the protocol.

Virtual routing interfaces (Layer 2 switches only)

You need to configure virtual routing interfaces if an IP, IPX, or Appletalk protocol VLAN, IP subnet VLAN, AppleTalk cable VLAN, or IPX network VLAN needs to route protocols to another port-based VLAN on the same router. A virtual routing interface can be associated with the ports in only a single port-based VLAN. Virtual router interfaces must be defined at the highest level of the VLAN hierarchy.

If you do not need to further partition the port-based VLAN by defining separate Layer 3 VLANs, you can define a single virtual routing interface at the port-based VLAN level and enable IP, IPX, and Appletalk routing on a single virtual routing interface.

Some configurations may require simultaneous switching and routing of the same single protocol across different sets of ports on the same router. When IP, IPX, or Appletalk routing is enabled on a Brocade Layer 3 switch, you can route these protocols on specific interfaces while bridging them on other interfaces. In this scenario, you can create two separate backbones for the same protocol, one bridged and one routed.

To bridge IP, IPX, or Appletalk at the same time these protocols are being routed, you need to configure an IP protocol, IP subnet, IPX protocol, IPX network, or Appletalk protocol VLAN and not assign a virtual routing interface to the VLAN. Packets for these protocols are bridged or switched at Layer 2 across ports on the router that are included in the Layer 3 VLAN. If these VLANs are built within port-based VLANs, they can be tagged across a single set of backbone fibers to create separate Layer 2 switched and Layer 3 routed backbones for the same protocol on a single physical backbone.

Routing between VLANs using virtual routing interfaces (Layer 3 switches only)

Brocade calls the ability to route between VLANs with virtual routing interfaces ***Integrated Switch Routing (ISR)***. There are some important concepts to understand before designing an ISR backbone.

Virtual router interfaces can be defined on port-based, IP protocol, IP subnet, IPX protocol, IPX network, AppleTalk protocol, and AppleTalk cable VLANs.

To create any type of VLAN on a Brocade Layer 3 switch, Layer 2 forwarding must be enabled. When Layer 2 forwarding is enabled, the Layer 3 switch becomes a Switch on all ports for all non-routable protocols.

If the router interfaces for IP, IPX, or AppleTalk are configured on physical ports, then routing occurs independent of the Spanning Tree Protocol (STP). However, if the router interfaces are defined for any type VLAN, they are virtual routing interfaces and are subject to the rules of STP.

If your backbone consists of virtual routing interfaces all within the same STP domain, it is a bridged backbone, not a routed one. This means that the set of backbone interfaces that are blocked by STP will be blocked for routed protocols as well. The routed protocols will be able to cross these paths only when the STP state of the link is FORWARDING. This problem is easily avoided by proper network design.

When designing an ISR network, pay attention to your use of virtual routing interfaces and the spanning-tree domain. If Layer 2 switching of your routed protocols (IP, IPX, AppleTalk) is not required across the backbone, then the use of virtual routing interfaces can be limited to edge switch ports within each router. Full backbone routing can be achieved by configuring routing on each physical interface that connects to the backbone. Routing is independent of STP when configured on a physical interface.

If your ISR design requires that you switch IP, IPX, or Appletalk at Layer 2 while simultaneously routing the same protocols over a single backbone, then create multiple port-based VLANs and use VLAN tagging on the backbone links to separate your Layer 2 switched and Layer 3 routed networks.

There is a separate STP domain for each port-based VLAN. Routing occurs independently across port-based VLANs or STP domains. You can define each end of each backbone link as a separate tagged port-based VLAN. Routing will occur independently across the port-based VLANs. Because each port-based VLAN STP domain is a single point-to-point backbone connection, you are guaranteed to never have an STP loop. STP will never block the virtual router interfaces within the tagged port-based VLAN, and you will have a fully routed backbone.

Dynamic port assignment (Layer 2 switches and Layer 3 switches)

All Switch ports are dynamically assigned to any Layer 3 VLAN on Brocade Layer 2 switches and any non-routable VLAN on Brocade Layer 3 switches. To maintain explicit control of the VLAN, you can explicitly exclude ports when configuring any Layer 3 VLAN on a Brocade Layer 2 switch or any non-routable Layer 3 VLAN on a Brocade Layer 3 switch.

If you do not want the ports to have dynamic membership, you can add them statically. This eliminates the need to explicitly exclude the ports that you do not want to participate in a particular Layer 3 VLAN.

Assigning a different VLAN ID to the default VLAN

When you enable port-based VLANs, all ports in the system are added to the default VLAN. By default, the default VLAN ID is "VLAN 1". The default VLAN is not configurable. If you want to use the VLAN ID "VLAN 1" as a configurable VLAN, you can assign a different VLAN ID to the default VLAN.

To reassign the default VLAN to a different VLAN ID, enter the following command.

```
Brocade(config)# default-vlan-id 4095
```

Syntax: [no] default-vlan-d <vlan-id>

You must specify a valid VLAN ID that is not already in use. For example, if you have already defined VLAN 10, do not try to use "10" as the new VLAN ID for the default VLAN. Valid VLAN IDs are numbers from 1 – 4095.

NOTE

does not change the properties of the default VLAN. Changing the name allows you to use the VLAN ID “1” as a configurable VLAN.

Assigning different VLAN IDs to reserved VLANs 4091 and 4092

If you want to use VLANs 4091 and 4092 as configurable VLANs, you can assign them to different VLAN IDs.

For example, to reassign reserved VLAN 4091 to VLAN 10, enter the following commands.

```
Brocade(config)# reserved-vlan-map vlan 4091 new-vlan 10
Reload required. Please write memory and then reload or power cycle.
Brocade(config)# write mem
Brocade(config)# exit
Brocade# reload
```

NOTE

You must save the configuration (write mem) and reload the software to place the change into effect.

The above configuration changes the VLAN ID of 4091 to 10. After saving the configuration and reloading the software, you can configure VLAN 4091 as you would any other VLAN.

Syntax: [no] reserved-vlan-map vlan 4091 | 4092 new-vlan <vlan-id>

For <vlan-id>, enter a valid VLAN ID that is not already in use. For example, if you have already defined VLAN 20, do not try to use “20 as the new VLAN ID. Valid VLAN IDs are numbers from 1 – 4090, 4093, and 4095. VLAN ID 4094 is reserved for use by the Single Spanning Tree feature.

Viewing reassigned VLAN IDs for reserved VLANs 4091 and 4092

To view the assigned VLAN IDs for reserved VLANs 4091 and 4092, use the **show reserved-vlan-map** command. The reassigned VLAN IDs also display in the output of the **show running-config** and **show config** commands.

The following shows example output for the **show reserved-vlan-map** command.

```
Brocade # show reserved-vlan-map
```

Reserved Purpose	Default	Re-assign	Current
CPU VLAN	4091	10	10
All Ports VLAN	4092	33	33

Syntax: show reserved-vlan-map

The following table defines the fields in the output of the **show reserved-vlan-map** command.

TABLE 140 Output of the **show reserved-vlan-map** command

Field	Description
Reserved Purpose	Describes for what the VLAN is reserved. Note that the description is for Brocade internal VLAN management.
Default	The default VLAN ID of the reserved VLAN.
Re-assign	The VLAN ID to which the reserved VLAN was reassigned. ¹
Current	The current VLAN ID for the reserved VLAN. ¹

1. If you reassign a reserved VLAN without saving the configuration and reloading the software, the reassigned VLAN ID will display in the **Re-assign** column. However, the previously configured or default VLAN ID will display in the **Current** column until the configuration is saved and the device reloaded.

Assigning trunk group ports

When a “lead” trunk group port is assigned to a VLAN, all other members of the trunk group are automatically added to that VLAN. A lead port is the first port of a trunk group port range; for example, “1” in 1 – 4 or “5” in 5 – 8.

Refer to [“Trunk group rules”](#) on page 699 for more information.

Enable spanning tree on a VLAN

The spanning tree bridge and port parameters are configurable using one CLI command set at the Global Configuration Level of each Port-based VLAN. Suppose you want to enable the IEEE 802.1D STP across VLAN 3. To do so, use the following method.

NOTE

When port-based VLANs are not operating on the system, STP is set on a system-wide level at the global CONFIG level of the CLI.

1. Access the global CONFIG level of the CLI on FESX-A by entering the following commands.

```
Brocade-A> enable
No password has been assigned yet...
Brocade-A# configure terminal
Brocade-A(config)#
```

2. Access the level of the CLI for configuring port-based VLAN 3 by entering the following command.

```
Brocade-A(config)#
Brocade-A(config)# vlan 3
Brocade-A(config-vlan-3)#
```

3. From VLAN 3 configuration level of the CLI, enter the following command to enable STP on all tagged and untagged ports associated with VLAN 3.

```
Brocade-B(config-vlan-3)#
Brocade-B(config-vlan-3)# spanning-tree
Brocade-B(config-vlan-3)#
```

4. Enter the following commands to exit the VLAN CONFIG mode and save the configuration to the system-config file on flash memory.

```
Brocade-B(config-vlan-3)#
Brocade-B(config-vlan-3)# end
Brocade-B# write memory
Brocade-B#
```

5. Repeat steps 1 – 4 on FESX-B.

NOTE

You do not need to configure values for the STP parameters. All parameters have default values as noted below. Additionally, all values will be globally applied to all ports on the system or on the port-based VLAN for which they are defined.

To configure a specific path-cost or priority value for a given port, enter those values using the key words in the brackets [] shown in the syntax summary below. If you do not want to specify values for any given port, this portion of the command is not required.

Syntax: `vlan <vlan-id> by port`

Syntax: `[no] spanning-tree`

Syntax: `spanning-tree [ethernet [<slotnum>/]<portnum> path-cost <value> priority <value>]
forward-delay <value> hello-time <value> maximum-age <time> priority <value>`

Bridge STP parameters (applied to all ports within a VLAN):

- Forward Delay – the period of time a bridge will wait (the listen and learn period) before forwarding data packets. Possible values: 4 – 30 seconds. Default is 15.
- Maximum Age – the interval a bridge will wait for receipt of a hello packet before initiating a topology change. Possible values: 6 – 40 seconds. Default is 20.
- Hello Time – the interval of time between each configuration BPDU sent by the root bridge. Possible values: 1 – 10 seconds. Default is 2.
- Priority – a parameter used to identify the root bridge in a network. The bridge with the lowest value has the highest priority and is the root. Possible values: 1 – 65,535. Default is 32,678.

Port parameters (applied to a specified port within a VLAN):

- Path Cost – a parameter used to assign a higher or lower path cost to a port. Possible values: 1 – 65535. Default is (1000/Port Speed) for Half-Duplex ports and is (1000/Port Speed)/2 for Full-Duplex ports.
- Priority – value determines when a port will be rerouted in relation to other ports. Possible values: 0 – 255. Default is 128.

Configuring IP subnet, IPX network and protocol-based VLANs

Protocol-based VLANs provide the ability to define separate broadcast domains for several unique Layer 3 protocols within a single Layer 2 broadcast domain. Some applications for this feature might include security between departments with unique protocol requirements. This feature enables you to limit the amount of broadcast traffic end-stations, servers, and routers need to accept.

IP subnet, IPX network, and protocol-based VLAN configuration example

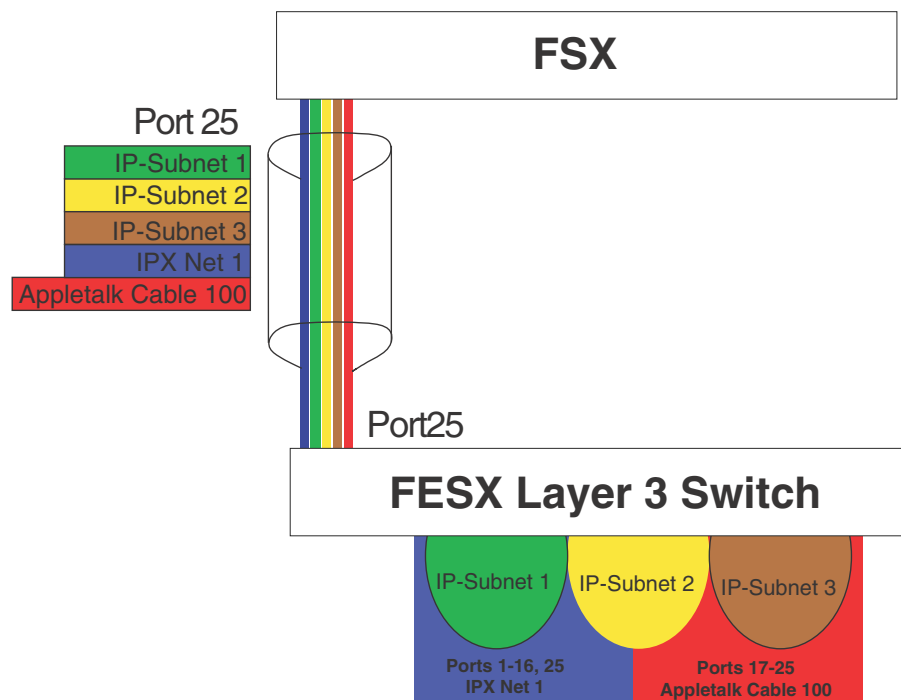
Suppose you want to create five separate Layer 3 broadcast domains within a single Layer 2 STP broadcast domain:

- Three broadcast domains, one for each of three separate IP subnets
- One for IPX Network 1
- One for the Appletalk protocol

Also suppose you want a single router interface to be present within all of these separate broadcast domains, without using IEEE 802.1Q VLAN tagging or any proprietary form of VLAN tagging.

Figure 81 shows this configuration.

FIGURE 81 Protocol-based (Layer 3) VLANs



To configure the VLANs shown in [Figure 81](#), use the following procedure.

1. To permanently assign ports 1 – 8 and port 25 to IP subnet VLAN 1.1.1.0, enter the following commands.

```
Brocade(config-vlan-2)# ip-subnet 1.1.1.0/24 name Green
Brocade(config-vlan-ip-subnet)# no dynamic
Brocade(config-vlan-ip-subnet)# static ethernet 1 to 8 ethernet 25
```

2. To permanently assign ports 9 – 16 and port 25 to IP subnet VLAN 1.1.2.0, enter the following commands.

```
Brocade(config-vlan-3)# ip-subnet 1.1.2.0/24 name Yellow
Brocade(config-vlan-ip-subnet)# no dynamic
Brocade(config-vlan-ip-subnet)# static ethernet 9 to 16 ethernet 25
```

3. To permanently assign ports 17 – 25 to IP subnet VLAN 1.1.3.0, enter the following commands.

```
Brocade(config-vlan-4)# ip-subnet 1.1.3.0/24 name Brown
Brocade(config-vlan-ip-subnet)# no dynamic
Brocade(config-vlan-ip-subnet)# static ethernet 17 to 25
```

4. To permanently assign ports 1 – 12 and port 25 to IPX network 1 VLAN, enter the following commands.

```
Brocade(config-ip-subnet)# ipx-network 1 ethernet_802.3 name Blue
Brocade(config-ipx-network)# no dynamic
Brocade(config-ipx-network)# static ethernet 1 to 12 ethernet 25
Brocade(config-ipx-network)#
```

5. To permanently assign ports 12 – 25 to Appletalk VLAN, enter the following commands.

```
Brocade(config-ipx-proto)# atalk-proto name Red
Brocade(config-atalc-proto)# no dynamic
Brocade(config-atalc-proto)# static ethernet 13 to 25
Brocade(config-atalc-proto)# end
Brocade# write memory
Brocade#
```

Syntax: `ip-subnet <ip-addr> <ip-mask> [name <string>]`

Syntax: `ipx-network <ipx-network-number> <frame-encapsulation-type> netbios-allow | netbios-disallow [name <string>]`

Syntax: `ip-proto | ipx-proto | atalk-proto | decnet-proto | netbios-proto | other-proto static | exclude | dynamic ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum>] [name <string>]`

IP subnet, IPX network, and protocol-based VLANs within port-based VLANs

If you plan to use port-based VLANs in conjunction with protocol-based VLANs, you must create the port-based VLANs first. Once you create a port-based VLAN, then you can assign Layer 3 protocol VLANs within the boundaries of the port-based VLAN. Generally, you create port-based VLANs to allow multiple separate STP domains.

Example

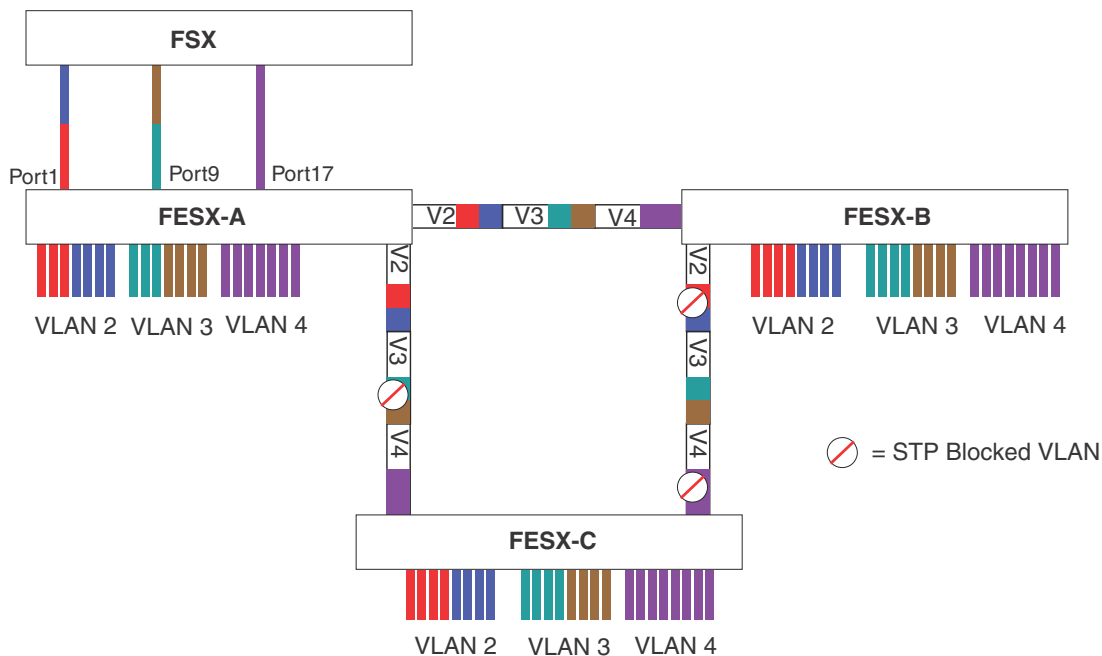
Suppose you need to provide three separate STP domains across an enterprise campus backbone. The first STP domain (VLAN 2) requires a set of ports at each Layer 2 switch location to be statically mapped to IP only. No other protocols can enter the switches on this set of ports.

A second set of ports within STP domain VLAN 2 will be restricted to only IPX traffic. The IP and IPX protocol VLANs will overlap on Port 1 of FESX-A to support both protocols on the same router interface. The IP subnets and IPX network that span the two protocol VLANs will be determined by the FastIron router configuration. The IP and IPX Protocol VLANs ensure that only the ports included in the each Layer 3 protocol VLAN will see traffic from the FastIron router.

The second STP domain (VLAN 3) requires that half the ports in the domain are dedicated to IP subnet 1.1.1.0/24 and the other ports are dedicated to IPX network 1. Similar to VLAN 2, Port 9 from VLAN 3 will be used to carry this IP subnet and IPX network to the FastIron router. No other protocols will be allowed to enter the network on VLAN 3. Also, no IP packets with a source address on subnet 1.1.1.0/24 or IPX packets with a source address on network 1 will be allowed to enter the switches on VLAN 3.

There is no need to segment Layer 3 broadcast domains within the STP broadcast domain (VLAN 4). The FastIron router will dictate the IP subnets and IPX network that are on VLAN 4. There are no Layer 3 protocol restrictions on VLAN 4; however, the FastIron router is configured to only forward IP and IPX between STP domains.

FIGURE 82 More protocol-based VLANs



To configure the Layer 3 VLANs on the FESX Layer 2 switches in [Figure 82](#), use the following procedure.

Configuring Layer 3 VLANs on FESX-A

Enter the following commands to configure FESX-A.

1. Create port-based VLAN 2 and assign the untagged and tagged ports that will participate in this VLAN.

```
Brocade-A >en
Brocade-A# config t
Brocade-A(config)# vlan 2 name IP_IPX_Protocol
Brocade-A(config-vlan-2)# untagged e1 to 8
Brocade-A(config-vlan-2)# tagged e25 to 26
```

2. Enable STP and set the priority to force FESX-A to be the root bridge for VLAN 2.

```
Brocade-A(config-vlan-2)# spanning-tree
Brocade-A(config-vlan-2)# spanning-tree priority 500
Brocade-A(config-vlan-2)#
```

3. Create the IP and IPX protocol-based VLANs and statically assign the ports within VLAN 2 that will be associated with each protocol-based VLAN.

```
Brocade-A(config-vlan-2)# ip-proto name Red
Brocade-A(config-vlan-ip-proto)# no dynamic
Brocade-A(config-vlan-ip-proto)# static e1 to 4 e25 to 26
Brocade-A(config-vlan-ip-proto)# exclude e5 to 8
Brocade-A(config-vlan-ip-proto)# ipx-proto name Blue
Brocade-A(config-vlan-ipx-proto)# no dynamic
Brocade-A(config-vlan-ipx-proto)# static e1 e5 to 8 e25 to 26
Brocade-A(config-vlan-ipx-proto)# exclude e2 to 4
```

4. To prevent machines with non-IP protocols from getting into the IP portion of VLAN 2, create another Layer 3 protocol VLAN to exclude all other protocols from the ports that contains the IP-protocol VLAN. To do so, enter the following commands.

```
Brocade-A(config-vlan-ipx-PROTO)# other-PROTO name Block_OTHER_PROTO
Brocade-A(config-vlan-OTHER-PROTO)# no dynamic
Brocade-A(config-vlan-OTHER-PROTO)# exclude e1 to 8
Brocade-A(config-vlan-OTHER-PROTO)#
```

5. Create port-based VLAN 3. Note that FESX-B will be the root for this STP domain, so you do not need to adjust the STP priority.

```
Brocade-A(config-vlan-OTHER-PROTO)# vlan 3 name IP-Sub_IPX-Net_Vlans
Brocade-A(config-vlan-3)# untagged e9 to 16
Brocade-A(config-vlan-3)# tagged e25 to 26
Brocade-A(config-vlan-3)# spanning-tree
Brocade-A(config-vlan-3)#
```

6. Create IP subnet VLAN 1.1.1.0/24, IPX network 1, and other-protocol VLANs

```
Brocade-A(config-vlan-3)# ip-subnet 1.1.1.0/24 name Green
Brocade-A(config-vlan-ip-subnet)# no dynamic
Brocade-A(config-vlan-ip-subnet)# static e9 to 12 e25 to 26
Brocade-A(config-vlan-ip-subnet)# exclude e13 to 16
Brocade-A(config-vlan-ip-subnet)# ipx-net 1 ethernet_802.3 name Brown
Brocade-A(config-vlan-ipx-network)# no dynamic
Brocade-A(config-vlan-ipx-network)# static e9 e13 to 16 e25 to 26
Brocade-A(config-vlan-ipx-network)# exclude e10 to 12
Brocade-A(config-vlan-ipx-network)# other-PROTO name Block_OTHER_PROTO
Brocade-A(config-vlan-OTHER-PROTO)# no dynamic
Brocade-A(config-vlan-OTHER-PROTO)# exclude e9 to 16
Brocade-A(config-vlan-OTHER-PROTO)#
```

7. Configure the last port-based VLAN 4. You need to set the STP priority for this VLAN because FESX-A will be the root bridge for this VLAN. Because you do not need to partition this STP domain into multiple Layer 3 broadcast domains, this is the only configuration required for VLAN 4.

```
Brocade-A(config-vlan-OTHER-PROTO)# vlan 4 name Purple_ALL-Protocols
Brocade-A(config-vlan-4)# untagged e17 to 24
Brocade-A(config-vlan-4)# tagged e25 to 26
Brocade-A(config-vlan-4)# spanning-tree
Brocade-A(config-vlan-4)# spanning-tree priority 500
Brocade-A(config-vlan-4)#
```

Configuring Layer 3 VLANs on FESX-B

Enter the following commands to configure FESX-B.

```
Brocade# config t
Brocade(config)# host Brocade-B
Brocade-B(config)#vlan 2 name IP_IPX_Protocol
Brocade-B(config-vlan-2)# untagged e1 to 8
Brocade-B(config-vlan-2)# tagged e25 to 26
Brocade-B(config-vlan-2)# spanning-tree
Brocade-B(config-vlan-2)# ip-PROTO name Red
Brocade-B(config-vlan-ip-PROTO)# # no dynamic
Brocade-B(config-vlan-ip-PROTO)# static e1 to 4 e25 to 26
Brocade-B(config-vlan-ip-PROTO)# exclude e5 to 8
Brocade-B(config-vlan-ip-PROTO)# ipx-PROTO name Blue
Brocade-B(config-vlan-ipx-PROTO)# no dynamic
```

```

Brocade-B(config-vlan-ipx-PROTO)# static e5 to 8 e25 to 26
Brocade-B(config-vlan-ipx-PROTO)# exclude e1 to 4
Brocade-B(config-vlan-other-PROTO)# vlan 3 name IP-Sub_IPX-Net_VLANS
Brocade-B(config-vlan-3)# untagged e9 to 16
Brocade-B(config-vlan-3)# tagged e25 to 26
Brocade-B(config-vlan-3)# spanning-tree
Brocade-B(config-vlan-3)# spanning-tree priority 500
Brocade-B(config-vlan-3)# ip-sub 1.1.1.0/24 name Green
Brocade-B(config-vlan-ip-subnet)# no dynamic
Brocade-B(config-vlan-ip-subnet)# static e9 to 12 e25 to 26
Brocade-B(config-vlan-ip-subnet)# exclude e13 to 16
Brocade-B(config-vlan-ip-subnet)# ipx-net 1 ethernet_802.3 name Brown
Brocade-B(config-vlan-ipx-network)# no dynamic
Brocade-B(config-vlan-ipx-network)# static e13 to 16 e25 to 26
Brocade-B(config-vlan-ipx-network)# exclude e9 to 12
Brocade-B(config-vlan-ipx-network)# vlan 4 name Purple_ALL-Protocols
Brocade-B(config-vlan-4)# untagged e17 to 24
Brocade-B(config-vlan-4)# tagged e25 to 26
Brocade-B(config-vlan-4)# spanning-tree

```

Configuring Layer 3 VLANs on FESX-C

Enter the following commands to configure FESX-C.

```

Brocade# config t
Brocade(config)# host Brocade-C
Brocade-C(config)# vlan 2 name IP_IPX_Protocol
Brocade-C(config-vlan-2)# untagged e1 to 8
Brocade-C(config-vlan-2)# tagged e25 to 26
Brocade-C(config-vlan-2)# spanning-tree
Brocade-C(config-vlan-2)# ip-PROTO name Red
Brocade-C(config-vlan-ip-PROTO)# no dynamic
Brocade-C(config-vlan-ip-PROTO)# static e1 to 4 e25 to 26
Brocade-C(config-vlan-ip-PROTO)# exclude e5 to 8
Brocade-C(config-vlan-ip-PROTO)# ipx-PROTO name Blue
Brocade-C(config-vlan-ipx-PROTO)# no dynamic
Brocade-C(config-vlan-ipx-PROTO)# static e5 to 8 e25 to 26
Brocade-C(config-vlan-ipx-PROTO)# exclude e1 to 4
Brocade-C(config-vlan-other-PROTO)# vlan 3 name IP-Sub_IPX-Net_VLANS
Brocade-C(config-vlan-3)# untagged e9 to 16
Brocade-C(config-vlan-3)# tagged e25 to 26
Brocade-C(config-vlan-3)# spanning-tree
Brocade-C(config-vlan-3)# ip-sub 1.1.1.0/24 name Green
Brocade-C(config-vlan-ip-subnet)# no dynamic
Brocade-C(config-vlan-ip-subnet)# static e9 to 12 e25 to 26
Brocade-C(config-vlan-ip-subnet)# exclude e13 to 16
Brocade-C(config-vlan-ip-subnet)# ipx-net 1 ethernet_802.3 name Brown
Brocade-C(config-vlan-ipx-network)# no dynamic
Brocade-C(config-vlan-ipx-network)# static e13 to 16 e25 to 26
Brocade-C(config-vlan-ipx-network)# exclude e9 to 12
Brocade-C(config-vlan-ipx-network)# vlan 4 name Purple_ALL-Protocols
Brocade-C(config-vlan-4)# untagged e17 to 24
Brocade-C(config-vlan-4)# tagged e25 to 26
Brocade-C(config-vlan-4)# spanning-tree

```

IPv6 protocol VLAN configuration

You can configure a protocol-based VLAN as a broadcast domain for IPv6 traffic. When the Layer 3 switch receives an IPv6 multicast packet (a packet with 06 in the version field and 0xFF as the beginning of the destination address), the Layer 3 switch forwards the packet to all other ports.

NOTE

The Layer 3 switch forwards all IPv6 multicast packets to all ports except the port that received the packet, and does not distinguish among subnet directed multicasts.

You can add the VLAN ports as static ports or dynamic ports. A static port is always an active member of the VLAN. Dynamic ports within any protocol VLAN age out after 10 minutes if no member protocol traffic is received on a port within the VLAN. The aged out port, however, remains as a candidate dynamic port for that VLAN. The port becomes active in the VLAN again if member protocol traffic is received on that port.

Once a port is re-activated, the aging out period for the port is reset to 20 minutes. Each time a member protocol packet is received by a candidate dynamic port (aged out port) the port becomes active again and the aging out period is reset for 20 minutes.

NOTE

You can disable VLAN membership aging of dynamically added ports. Refer to [“Disabling membership aging of dynamic VLAN ports”](#) on page 783).

To configure an IPv6 VLAN, enter commands such as the following.

```
Brocade(config)# vlan 2
Brocade(config-vlan-2)# untagged ethernet 1/1 to 1/8
Brocade(config-vlan-2)# ipv6-proto name V6
Brocade(config-ipv6-subnet)# static ethernet 1/1 to 1/6
Brocade(config-ipv6-subnet)# dynamic
```

The first two commands configure a port-based VLAN and add ports 1/1 – 1/8 to the VLAN. The remaining commands configure an IPv6 VLAN within the port-based VLAN. The **static** command adds ports 1/1 – 1/6 as static ports, which do not age out. The **dynamic** command adds the remaining ports, 1/7 – 1/8, as dynamic ports. These ports are subject to aging as described above.

Syntax: [no] ipv6-proto [name <string>]

Routing between VLANs using virtual routing interfaces (Layer 3 switches only)

Brocade Layer 3 switches offer the ability to create a virtual routing interface within a Layer 2 STP port-based VLAN or within each Layer 3 protocol, IP subnet, or IPX network VLAN. This combination of multiple Layer 2 or Layer 3 broadcast domains, or both, and virtual routing interfaces are the basis for Brocade's very powerful Integrated Switch Routing (ISR) technology. ISR is very flexible and can solve many networking problems. The following example is meant to provide ideas by demonstrating some of the concepts of ISR.

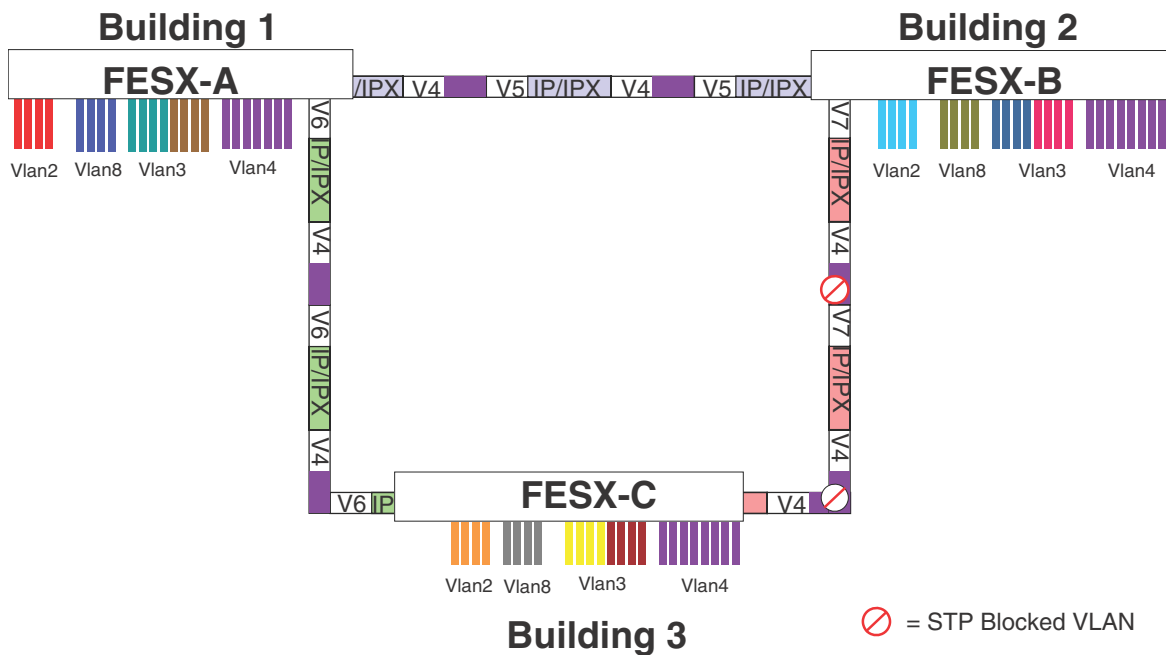
Example

Suppose you want to move routing out to each of three buildings in a network. Remember that the only protocols present on VLAN 2 and VLAN 3 are IP and IPX. Therefore, you can eliminate tagged ports 25 and 26 from both VLAN 2 and VLAN 3 and create new tagged port-based VLANs to support separate IP subnets and IPX networks for each backbone link.

You also need to create unique IP subnets and IPX networks within VLAN 2 and VLAN 3 at each building. This will create a fully routed IP and IPX backbone for VLAN 2 and VLAN 3. However, VLAN 4 has no protocol restrictions across the backbone. In fact there are requirements for NetBIOS and DecNet to be bridged among the three building locations. The IP subnet and IPX network that exists within VLAN 4 must remain a flat Layer 2 switched STP domain. You enable routing for IP and IPX on a virtual routing interface only on FESX-A. This will provide the flat IP and IPX segment with connectivity to the rest of the network. Within VLAN 4 IP and IPX will follow the STP topology. All other IP subnets and IPX networks will be fully routed and have use of all paths at all times during normal operation.

Figure 83 shows the configuration described above.

FIGURE 83 Routing between protocol-based VLANs



To configure the Layer 3 VLANs and virtual routing interfaces on the FESX Layer 3 switch in Figure 83, use the following procedure.

Configuring Layer 3 VLANs and virtual routing interfaces on the FESX-A

Enter the following commands to configure FESX-A. The following commands enable OSPF or RIP routing.

```
Brocade>en
No password has been assigned yet...
Brocade# configure terminal
Brocade(config)# hostname FESX-A
Brocade-A(config)# router ospf
Brocade-A(config-ospf-router)# area 0.0.0.0 normal
Please save configuration to flash and reboot.
Brocade-A(config-ospf-router)#
```

The following commands create the port-based VLAN 2. In the previous example, an external FESX defined the router interfaces for VLAN 2. With ISR, routing for VLAN 2 is done locally within each FESX. Therefore, there are two ways you can solve this problem. One way is to create a unique IP subnet and IPX network VLAN, each with its own virtual routing interface and unique IP or IPX address within VLAN 2 on each FESX. In this example, this is the configuration used for VLAN 3. The second way is to split VLAN 2 into two separate port-based VLANs and create a virtual router interface within each port-based VLAN. Later in this example, this second option is used to create a port-based VLAN 8 to show that there are multiple ways to accomplish the same task with ISR.

You also need to create the Other-Protocol VLAN within port-based VLAN 2 and 8 to prevent unwanted protocols from being Layer 2 switched within port-based VLAN 2 or 8. Note that the only port-based VLAN that requires STP in this example is VLAN 4. You will need to configure the rest of the network to prevent the need to run STP.

```
Brocade-A(config-ospf-router)# vlan 2 name IP-Subnet_1.1.2.0/24
Brocade-A(config-vlan-2)# untagged ethernet 1 to 4
Brocade-A(config-vlan-2)# no spanning-tree
Brocade-A(config-vlan-2)# router-interface ve1
Brocade-A(config-vlan-2)# other-proto name block_other_protocols
Brocade-A(config-vlan-other-proto)# no dynamic
Brocade-A(config-vlan-other-proto)# exclude ethernet 1 to 4
```

Once you have defined the port-based VLAN and created the virtual routing interface, you need to configure the virtual routing interface just as you would configure a physical interface.

```
Brocade-A(config-vlan-other-proto)# interface ve1
Brocade-A(config-vif-1)# ip address 1.1.2.1/24
Brocade-A(config-vif-1)# ip ospf area 0.0.0.0
```

Do the same thing for VLAN 8.

```
Brocade-A(config-vif-1)# vlan 8 name IPX_Network2
Brocade-A(config-vlan-8)# untagged ethernet 5 to 8
Brocade-A(config-vlan-8)# no spanning-tree
Brocade-A(config-vlan-8)# router-interface ve 2
Brocade-A(config-vlan-8)# ipx-network 2 ethernet_802.3
Brocade-A(config-vlan-8)# other-proto name block-other-protocols
Brocade-A(config-vlan-other-proto)# no dynamic
Brocade-A(config-vlan-other-proto)# exclude ethernet 5 to 8
Brocade-A(config-vlan-other-proto)# interface ve2
Brocade-A(config-vif-1)# ip address 1.1.2.2/24
Brocade-A(config-vif-1)# ip ospf area 0.0.0.0
```


The next thing you need to do is create VLAN 3. This is very similar to the previous example with the addition of virtual routing interfaces to the IP subnet and IPX network VLANs. Also there is no need to exclude ports from the IP subnet and IPX network VLANs on the router.

```
Brocade-A(config-vif-2)# vlan 3 name IP_Sub_&_IPX_Net_VLAN
Brocade-A(config-vlan-3)# untagged ethernet 9 to 16
Brocade-A(config-vlan-3)# no spanning-tree
Brocade-A(config-vlan-3)# ip-subnet 1.1.1.0/24
Brocade-A(config-vlan-ip-subnet)# static ethernet 9 to 12
Brocade-A(config-vlan-ip-subnet)# router-interface ve3
Brocade-A(config-vlan-ip-subnet)# ipx-network 1 ethernet_802.3
Brocade-A(config-vlan-ipx-network)# static ethernet 13 to 16
Brocade-A(config-vlan-ipx-network)# router-interface ve4
Brocade-A(config-vlan-ipx-network)# other-proto name block-other-protocols
Brocade-A(config-vlan-other-proto)# exclude ethernet 9 to 16
Brocade-A(config-vlan-other-proto)# no dynamic
Brocade-A(config-vlan-other-proto)# interface ve 3
Brocade-A(config-vif-3)# ip addr 1.1.1.1/24
Brocade-A(config-vif-3)# ip ospf area 0.0.0.0
```

Now configure VLAN 4. Remember this is a flat segment that, in the previous example, obtained its IP default gateway and IPX router services from an external FESX. In this example, FESX-A will provide the routing services for VLAN 4. You also want to configure the STP priority for VLAN 4 to make FESX-A the root bridge for this VLAN.

```
Brocade-A(config-vif-4)# vlan 4 name Bridged_ALL_Protocols
Brocade-A(config-vlan-4)# untagged ethernet 17 to 24
Brocade-A(config-vlan-4)# tagged ethernet 25 to 26
Brocade-A(config-vlan-4)# spanning-tree
Brocade-A(config-vlan-4)# spanning-tree priority 500
Brocade-A(config-vlan-4)# ipx network 3 ethernet_802.3
Brocade-A(config-vlan-ipx-network)# exit
Brocade-A(config-vlan-4)# router-interface ve5
Brocade-A(config-vlan-4)# interface ve5
Brocade-A(config-vif-5)# ip address 1.1.3.1/24
Brocade-A(config-vif-5)# ip ospf area 0.0.0.0
```

It is time to configure a separate port-based VLAN for each of the routed backbone ports (Ethernet 25 and 26).

If you do not create a separate tagged port-based VLAN for each point-to-point backbone link, you need to include tagged interfaces for Ethernet 25 and 26 within VLANs 2, 3, and 8. This type of configuration makes the entire backbone a single STP domain for each VLAN 2, 3, and 8. This is the configuration used in the example in [“Configuring IP subnet, IPX network and protocol-based VLANs”](#) on page 770. In this scenario, the virtual routing interfaces within port-based VLANs 2, 3, and 8 will be accessible using only one path through the network. The path that is blocked by STP is not available to the routing protocols until it is in the STP FORWARDING state.

```
Brocade-A(config-vif-5)# vlan 5 name Rtr_BB_to_Bldg.2
Brocade-A(config-vlan-5)# tagged ethernet 25
Brocade-A(config-vlan-5)# no spanning-tree
Brocade-A(config-vlan-5)# router-interface ve6
Brocade-A(config-vlan-5)# vlan 6 name Rtr_BB_to_Bldg.3
Brocade-A(config-vlan-6)# tagged ethernet 26
Brocade-A(config-vlan-6)# no spanning-tree
Brocade-A(config-vlan-6)# ipx-network 4 ethernet_802.3
```

```
Brocade-A(config-vlan-ipx-network)# exit
Brocade-A(config-vlan-6)# router-interface ve7
Brocade-A(config-vlan-6)# interface ve6
Brocade-A(config-vif-6)# ip addr 1.1.4.1/24
Brocade-A(config-vif-6)# ip ospf area 0.0.0.0
Brocade-A(config-vif-6)# interface ve7
Brocade-A(config-vif-7)# ip addr 1.1.5.1/24
Brocade-A(config-vif-7)# ip ospf area 0.0.0.0
Brocade-A(config-vif-7)#
```

This completes the configuration for FESX-A. The configuration for FESX-B and C is very similar except for a few issues which are as follows:

- IP subnets and IPX networks configured on FESX-B and FESX-C must be unique across the entire network, except for the backbone port-based VLANs 5, 6, and 7 where the subnet is the same but the IP address must change.
- There is no need to change the default priority of STP within VLAN 4.
- There is no need to include a virtual router interface within VLAN 4.
- The backbone VLAN between FESX-B and FESX-C must be the same at both ends and requires a new VLAN ID. The VLAN ID for this port-based VLAN is VLAN 7.

Configuring Layer 3 VLANs and virtual routing interfaces for FESX-B

Enter the following commands to configure FESX-B.

```
Brocade> en
No password has been assigned yet...
Brocade# config t
Brocade(config)# hostname FESX-B
Brocade-B(config)# router ospf
Brocade-B(config-ospf-router)# area 0.0.0.0 normal
Brocade-B(config-ospf-router)# vlan 2 name IP-Subnet_1.1.6.0/24
Brocade-B(config-vlan-2)# untagged ethernet 1 to 4
Brocade-B(config-vlan-2)# no spanning-tree
Brocade-B(config-vlan-2)# router-interface ve1
Brocade-B(config-vlan-2)# other-proto name block-other-protocols
Brocade-B(config-vlan-other-proto)# no dynamic
Brocade-B(config-vlan-other-proto)# exclude ethernet 1 to 4
Brocade-B(config-vlan-other-proto)# interface ve1
Brocade-B(config-vif-1)# ip addr 1.1.6.1/24
Brocade-B(config-vif-1)# ip ospf area 0.0.0.0
Brocade-B(config-vif-1)# vlan 8 name IPX_Network6
Brocade-B(config-vlan-8)# untagged ethernet 5 to 8
Brocade-B(config-vlan-8)# no span
Brocade-B(config-vlan-8)# router-interface ve2
Brocade-B(config-vlan-8)# ipx-net 6 ethernet_802.3
Brocade-B(config-vlan-ipx-network)# exit
Brocade-B(config-vlan-8)# other-proto name block-other-protocols
Brocade-B(config-vlan-other-proto)# no dynamic
Brocade-B(config-vlan-other-proto)# exclude ethernet 5 to 8
Brocade-B(config-vlan-other-proto)# interface ve2
Brocade-B(config-vif-2)# vlan 3 name IP_Sub_&_IPX_Net_VLAN
Brocade-B(config-vlan-3)# untagged ethernet 9 to 16
Brocade-B(config-vlan-3)# no spanning-tree
Brocade-B(config-vlan-3)# ip-subnet 1.1.7.0/24
Brocade-B(config-vlan-ip-subnet)# static ethernet 9 to 12
Brocade-B(config-vlan-ip-subnet)# router-interface ve3
```

Routing between VLANs using virtual routing interfaces (Layer 3 switches only)

```
Brocade-B(config-vlan-ip-subnet)# ipx-network 7 ethernet_802.3
Brocade-B(config-vlan-ipx-network)# static ethernet 13 to 16
Brocade-B(config-vlan-ipx-network)# router-interface ve4
Brocade-B(config-vlan-ipx-network)# other-proto name block-other-protocols
Brocade-B(config-vlan-other-proto)# exclude ethernet 9 to 16
Brocade-B(config-vlan-other-proto)# no dynamic
Brocade-B(config-vlan-other-proto)# interface ve 3
Brocade-B(config-vif-3)# ip addr 1.1.7.1/24
Brocade-B(config-vif-3)# ip ospf area 0.0.0.0
Brocade-B(config-vif-3)# interface ve4
Brocade-B(config-vif-4)# vlan 4 name Bridged_ALL_Protocols
Brocade-B(config-vlan-4)# untagged ethernet 17 to 24
Brocade-B(config-vlan-4)# tagged ethernet 25 to 26
Brocade-B(config-vlan-4)# spanning-tree
Brocade-B(config-vlan-4)# ipx-network 4 ethernet_802.3
Brocade-B(config-vlan-ipx-network)# exit
Brocade-B(config-vlan-4)# vlan 5 name Rtr_BB_to_Bldg.1
Brocade-B(config-vlan-5)# tagged ethernet 25
Brocade-B(config-vlan-5)# no spanning-tree
Brocade-B(config-vlan-5)# router-interface ve5
Brocade-B(config-vlan-5)# vlan 7 name Rtr_BB_to_Bldg.3
Brocade-B(config-vlan-7)# tagged ethernet 26
Brocade-B(config-vlan-7)# no spanning-tree
Brocade-B(config-vlan-7)# router-interface ve6
Brocade-B(config-vlan-7)# interface ve5
Brocade-B(config-vif-5)# ip addr 1.1.4.2/24
Brocade-B(config-vif-5)# ip ospf area 0.0.0.0
Brocade-B(config-vif-5)# interface ve6
Brocade-B(config-vif-6)# ip addr 1.1.8.1/24
Brocade-B(config-vif-6)# ip ospf area 0.0.0.0
Brocade-B(config-vif-6)#
```

Configuring Layer 3 VLANs and virtual routing interfaces for FESX-C

Enter the following commands to configure FESX-C.

```
Brocade> en
No password has been assigned yet...
Brocade# config t
Brocade(config)# hostname FESX-C
Brocade-C(config)# router ospf
Brocade-C(config-ospf-router)# area 0.0.0.0 normal
Brocade-C(config-ospf-router)# vlan 2 name IP-Subnet_1.1.9.0/24
Brocade-C(config-vlan-2)# untagged ethernet 1 to 4
Brocade-C(config-vlan-2)# no spanning-tree
Brocade-C(config-vlan-2)# router-interface ve1
Brocade-C(config-vlan-2)# other-proto name block-other-protocols
Brocade-C(config-vlan-other-proto)# no dynamic
Brocade-C(config-vlan-other-proto)# exclude ethernet 1 to 4
Brocade-C(config-vlan-other-proto)# interface ve1
Brocade-C(config-vif-1)# ip addr 1.1.9.1/24
Brocade-C(config-vif-1)# ip ospf area 0.0.0.0
Brocade-C(config-vif-1)# vlan 8 name IPX_Network9
Brocade-C(config-vlan-8)# untagged ethernet 5 to 8
Brocade-C(config-vlan-8)# no span
Brocade-C(config-vlan-8)# ipx-net 9 ethernet_802.3
```

```

Brocade-C(config-vlan-ipx-network)# exit
Brocade-C(config-vlan-8)# router-interface ve2
Brocade-C(config-vlan-8)# other-proto name block-other-protocols
Brocade-C(config-vlan-other-proto)# no dynamic
Brocade-C(config-vlan-other-proto)# exclude ethernet 5 to 8
Brocade-C(config-vlan-other-proto)# interface ve2
Brocade-C(config-vif-1)# ip addr 1.1.9.2/24
Brocade-C(config-vif-1)# ip ospf area 0.0.0.0
Brocade-C(config-vif-2)# vlan 3 name IP_Sub_&_IPX_Net_VLAN
Brocade-C(config-vlan-3)# untagged ethernet 9 to 16
Brocade-C(config-vlan-3)# no spanning-tree
Brocade-C(config-vlan-3)# ip-subnet 1.1.10.0/24
Brocade-C(config-vlan-ip-subnet)# static ethernet 9 to 12
Brocade-C(config-vlan-ip-subnet)# router-interface ve3
Brocade-C(config-vlan-ip-subnet)# ipx-network 10 ethernet_802.3
Brocade-C(config-vlan-ipx-network)# static ethernet 13 to 16
Brocade-C(config-vlan-ipx-network)# router-interface ve4
Brocade-C(config-vlan-ipx-network)# other-proto name block-other-protocols
Brocade-C(config-vlan-other-proto)# exclude ethernet 9 to 16
Brocade-C(config-vlan-other-proto)# no dynamic
Brocade-C(config-vlan-other-proto)# interface ve 3
Brocade-C(config-vif-3)# ip addr 1.1.10.1/24
Brocade-C(config-vif-3)# ip ospf area 0.0.0.0
Brocade-C(config-vif-3)# interface ve4
Brocade-C(config-vif-4)# vlan 4 name Bridged_ALL_Protocols
Brocade-C(config-vlan-4)# untagged ethernet 17 to 24
Brocade-C(config-vlan-4)# tagged ethernet 25 to 26
Brocade-C(config-vlan-4)# spanning-tree
Brocade-C(config-vlan-4)# vlan 7 name Rtr_BB_to_Bldg.2
Brocade-C(config-vlan-7)# tagged ethernet 25
Brocade-C(config-vlan-7)# no spanning-tree
Brocade-C(config-vlan-7)# ipx-network 8 ethernet_802.3
Brocade-C(config-vlan-ip-subnet)# exit
Brocade-C(config-vlan-7)# router-interface ve5
Brocade-C(config-vlan-7)# vlan 6 name Rtr_BB_to_Bldg.1
Brocade-C(config-vlan-6)# tagged ethernet 26
Brocade-C(config-vlan-6)# no spanning-tree
Brocade-C(config-vlan-6)# router-interface ve6
Brocade-C(config-vlan-6)# ipx network 5 ethernet_802.3
Brocade-C(config-vlan-6)# interface ve5
Brocade-C(config-vif-5)# ip addr 1.1.8.2/24
Brocade-C(config-vif-5)# ip ospf area 0.0.0.0
Brocade-C(config-vif-5)# interface ve6
Brocade-C(config-vif-6)# ip addr 1.1.5.2/24
Brocade-C(config-vif-6)# ip ospf area 0.0.0.0
Brocade-C(config-vif-6)#

```

Configuring protocol VLANs with dynamic ports

The configuration examples for protocol VLANs in the sections above show how to configure the VLANs using static ports. You also can configure the following types of protocol VLANs with dynamic ports:

- AppleTalk protocol
- IP protocol
- IPX protocol

- IP subnet
- IPX network

NOTE

The software does not support dynamically adding ports to AppleTalk cable VLANs. Conceptually, an AppleTalk cable VLAN consists of a single network cable, connected to a single port. Therefore, dynamic addition and removal of ports is not applicable.

NOTE

You cannot route to or from protocol VLANs with dynamically added ports.

Aging of dynamic ports

When you add the ports to the VLAN, the software automatically adds them all to the VLAN. However, dynamically added ports age out. If the age time for a dynamic port expires, the software removes the port from the VLAN. If that port receives traffic for the IP subnet or IPX network, the software adds the port to the VLAN again and starts the aging timer over. Each time the port receives traffic for the VLAN's IP subnet or IPX network, the aging timer starts over.

NOTE

You can disable VLAN membership aging of dynamically added ports. Refer to [“Disabling membership aging of dynamic VLAN ports”](#) on page 783).

Dynamic ports within any protocol VLAN age out after 10 minutes, if no member protocol traffic is received on a port within the VLAN. The aged out port, however, remains as a candidate dynamic port for that VLAN. The port becomes active in the VLAN again if member protocol traffic is received on that port.

Once a port is re-activated, the aging out period for the port is reset to 20 minutes. Each time a member protocol packet is received by a candidate dynamic port (aged out port) the port becomes active again and the aging out period is reset for 20 minutes.

Disabling membership aging of dynamic VLAN ports

You can disable VLAN membership aging of ports that are dynamically assigned to protocol or subnet-based VLANs. This feature resolves the connectivity issue that may occur in certain configurations when protocol or subnet VLANs are configured with dynamic port membership.

NOTE

This issue does not occur with statically assigned VLAN memberships. Thus, enable this feature only if your configuration includes dynamically assigned VLAN memberships for protocol or subnet VLANs.

To enable this feature, enter commands such as the following.

```
Brocade(config)# vlan 10 by port
Brocade(config-vlan-10)# interface ethernet 1/1 to 1/5
Brocade(config-vlan-10)# ip-proto name IP_Prot_VLAN
Brocade(config-vlan-ip-proto)# no-dynamic-aging
Brocade(config-vlan-ip-proto)# write memory
```

These commands create an IP protocol VLAN and disable the VLAN membership aging of ports that are dynamically assigned to the protocol VLAN.

Syntax: [no] no-dynamic-aging

Enter the **no** form of the command to disable this feature after it has been enabled.

By default, VLAN membership of dynamically assigned ports will age out after a period of time if no packets belonging to that protocol or subnet VLAN are received by the CPU.

The output of the **show running-config** command indicates if the no-dynamic-aging feature is enabled for a specific protocol or subnet VLAN.

Configuration guidelines for membership aging of dynamic VLAN ports

- You cannot dynamically add a port to a protocol VLAN if the port has any routing configuration parameters. For example, the port cannot have a virtual routing interface, IP subnet address, IPX network address, or AppleTalk network address configured on it.
- Once you dynamically add a port to a protocol VLAN, you cannot configure routing parameters on the port.
- Dynamic VLAN ports are not required or supported on AppleTalk cable VLANs.
- When protocol VLANs with dynamic ports are configured, the output of the **show running-config** command in the Router image will show the “dynamic” keyword. In the Switch image, the keyword is not shown in the output of the **show running-config** command.

NOTE

In the Switch image, all the ports are dynamic ports by-default, so the **dynamic** command does not appear in the **show running-config** command output. If you configure the **no dynamic** command, it will appear in the output of the **show running-config** command. Similarly in Router image, no ports are dynamic by-default, so the **no dynamic** command does not appear in the output of the **show running-config** command. If you configure the **dynamic** command, it will appear in the output of the **show running-config** command.

Configuring an IP, IPX, or AppleTalk Protocol VLAN with Dynamic Ports

To configure an IP, IPX, or AppleTalk protocol VLAN with dynamic ports, use the following method.

To configure port-based VLAN 10, then configure an IP protocol VLAN within the port-based VLAN with dynamic ports, enter the following commands such as the following.

```
Brocade(config)# vlan 10 by port
Brocade(config-vlan-10)# untagged ethernet 1/1 to 1/6
added untagged port ethe 1/1 to 1/6 to port-vlan 30.
Brocade(config-vlan-10)# ip-proto name IP_Prot_VLAN
Brocade(config-vlan-ip-proto)# dynamic
Brocade(config)# write memory
```

Syntax: **vlan** <vlan-id> **name** <string> [**by port**]

Syntax: **untagged ethernet** [<slotnum>/]<portnum> **to** [<slotnum>/]<portnum>

or

Syntax: **untagged ethernet** [<slotnum>/]<portnum> **ethernet** [<slotnum>/]<portnum>

NOTE

Use the first **untagged** command for adding a range of ports. Use the second command for adding separate ports (not in a range).

Syntax: **ip-proto** [**name** <string>]

Syntax: **ipx-proto** [**name** <string>]

Syntax: **appletalk-cable-vlan** <num> [**name** <string>]

Syntax: **dynamic**

The procedure is similar for IPX and AppleTalk protocol VLANs. Enter **ipx-proto** or **atalk-proto** instead of **ip-proto**.

Configuring an IP subnet VLAN with dynamic ports

To configure port-based VLAN 10, then configure an IP subnet VLAN within the port-based VLAN with dynamic ports, enter commands such as the following.

```
Brocade(config)# vlan 10 name IP_VLAN by port
Brocade(config-vlan-10)# untagged ethernet 1/1 to 1/6
added untagged port ethe 1/1 to 1/6 to port-vlan 10.
Brocade(config-vlan-10)# ip-subnet 1.1.1.0/24 name Mktg-LAN
Brocade(config-vlan-ip-subnet)# dynamic
Brocade(config)# write memory
```

These commands create a port-based VLAN on chassis ports 1/1 – 1/6 named “Mktg-LAN”, configure an IP subnet VLAN within the port-based VLAN, and then add ports from the port-based VLAN dynamically.

Syntax: **vlan** <vlan-id> **name** <string> [**by port**]

Syntax: **untagged ethernet** [<slotnum>/]<portnum> **to** [<slotnum>/]<portnum>

or

Syntax: **untagged ethernet** [<slotnum>/]<portnum> **ethernet** [<slotnum>/]<portnum>

NOTE

Use the first **untagged** command for adding a range of ports. Use the second command for adding separate ports (not in a range).

Syntax: **ip-subnet** <ip-addr> <ip-mask> [**name** <string>]

or

Syntax: **ip-subnet** <ip-addr>/<mask-bits> [**name** <string>]

Syntax: **dynamic**

Configuring an IPX network VLAN with dynamic ports

To configure port-based VLAN 20, then configure an IPX network VLAN within the port-based VLAN with dynamic ports, enter commands such as the following.

Configuring protocol VLANs with dynamic ports

```
Brocade(config)# vlan 20 name IPX_VLAN by port
Brocade(config-vlan-10)# untagged ethernet 2/1 to 2/6
added untagged port ethe 2/1 to 2/6 to port-vlan 20.
Brocade(config-vlan-10)# ipx-network abcd ethernet_ii name Eng-LAN
Brocade(config-vlan-ipx-network)# dynamic
Brocade(config)# write memory
```

These commands create a port-based VLAN on chassis ports 2/1 – 2/6 named “Eng-LAN”, configure an IPX network VLAN within the port-based VLAN, and then add ports from the port-based VLAN dynamically.

Syntax: `vlan <vlan-id> name <string> [by port]`

Syntax: `untagged ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>`

or

Syntax: `untagged ethernet [<slotnum>/]<portnum> ethernet [<slotnum>/]<portnum>`

NOTE

Use the first **untagged** command for adding a range of ports. Use the second command for adding separate ports (not in a range).

Syntax: `ipx-network <network-addr> ethernet_ii | ethernet_802.2 | ethernet_802.3 | ethernet_snap [name <string>]`

Syntax: `dynamic`

Configuring uplink ports within a port-based VLAN

You can configure a subset of the ports in a port-based VLAN as uplink ports. When you configure uplink ports in a port-based VLAN, the device sends all broadcast and unknown-unicast traffic from a port in the VLAN to the uplink ports, but not to other ports within the VLAN. Thus, the uplink ports provide tighter broadcast control within the VLAN.

This uplink port feature behaves the same as the private VLAN (PVLAN) feature, but with the ability to support tagged ports. This feature also supports two PVLAN modes: the Primary ports (uplink ports) and Isolated ports (host ports).

For example, if two ports within a port-based VLAN are Gbps ports attached to the network and the other ports are 10/100 ports attached to clients, you can configure the two ports attached to the network as uplink ports. In this configuration, broadcast and unknown-unicast traffic in the VLAN does not go to all ports. The traffic goes only to the uplink ports. The clients on the network do not receive broadcast and unknown-unicast traffic from other ports, including other clients.

Configuration considerations for uplink ports within a port-based VLAN

- When this feature is enabled, flooded traffic (unknown unicast, unregistered multicast, and broadcast traffic) is software forwarded.
- This feature should not be enabled with protocol VLANs or PVLANS in the same VLAN.

Configuration syntax for uplink ports within a port-based VLAN

To configure a port-based VLAN containing uplink ports, enter commands such as the following.

```
Brocade(config)# vlan 10 by port
Brocade(config-vlan-10)# untagged ethernet 1/1 to 1/24
Brocade(config-vlan-10)# untagged ethernet 2/1 to 2/2
Brocade(config-vlan-10)# uplink-switch ethernet 2/1 to 2/2
```

Syntax: `[no] uplink-switch ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]`

In this example, 24 ports on a 10/100 module and two Gbps ports on a Gbps module are added to port-based VLAN 10. The two Gbps ports are then configured as uplink ports.

IP subnet address on multiple port-based VLAN configuration

For a Brocade device to route between port-based VLANs, you must add a virtual routing interface to each VLAN. Generally, you also configure a unique IP subnet address on each virtual routing interface. For example, if you have three port-based VLANs, you add a virtual routing interface to each VLAN, then add a separate IP subnet address to each virtual routing interface. The IP address on each of the virtual routing interfaces must be in a separate subnet. The Brocade device routes Layer 3 traffic between the subnets using the subnet addresses.

NOTE

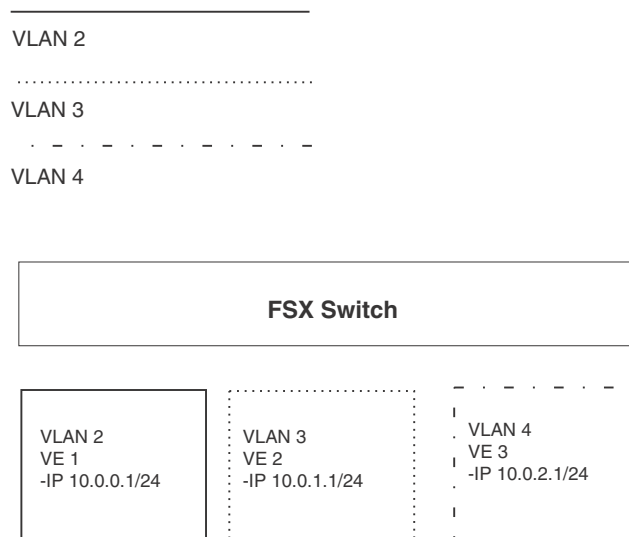
This feature applies only to Layer 3 switches.

NOTE

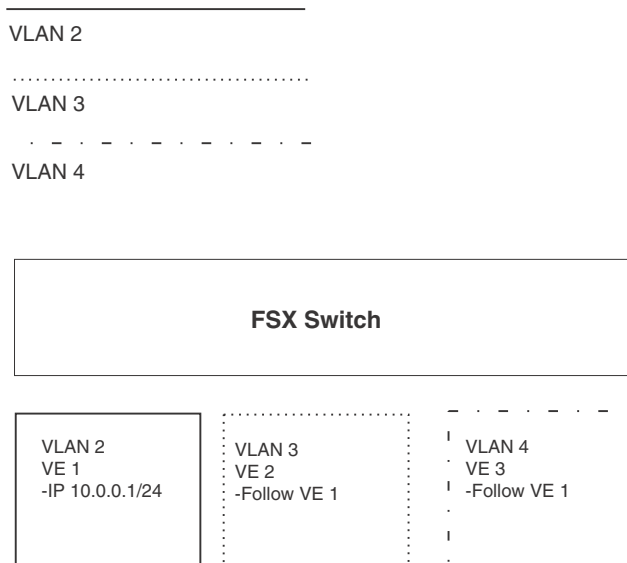
Before using the method described in this section, refer to [“VLAN groups and virtual routing interface group”](#) on page 791. You might be able to achieve the results you want using the methods in that section instead.

[Figure 84](#) shows an example of this type of configuration.

FIGURE 84 Multiple port-based VLANs with separate protocol addresses



As shown in this example, each VLAN has a separate IP subnet address. If you need to conserve IP subnet addresses, you can configure multiple VLANs with the same IP subnet address, as shown in [Figure 85](#).

FIGURE 85 Multiple port-based VLANs with the same protocol address

Each VLAN still requires a separate virtual routing interface. However, all three VLANs now use the same IP subnet address.

In addition to conserving IP subnet addresses, this feature allows containment of Layer 2 broadcasts to segments within an IP subnet. For ISP environments where the same IP subnet is allocated to different customers, placing each customer in a separate VLAN allows all customers to share the IP subnet address, while at the same time isolating them from one another Layer 2 broadcasts.

NOTE

You can provide redundancy to an IP subnet address that contains multiple VLANs using a pair of Brocade Layer 3 switches configured for Brocade VRRP (Virtual Router Redundancy Protocol).

The Brocade device performs proxy Address Resolution Protocol (ARP) for hosts that want to send IP traffic to hosts in other VLANs that are sharing the same IP subnet address. If the source and destination hosts are in the same VLAN, the Brocade device does not need to use ARP:

- If a host attached to one VLAN sends an ARP message for the MAC address of a host in one of the other VLANs using the same IP subnet address, the Brocade device performs a proxy ARP on behalf of the other host. The Brocade device then replies to the ARP by sending the virtual routing interface MAC address. The Brocade device uses the same MAC address for all virtual routing interfaces.

When the host that sent the ARP then sends a unicast packet addressed to the virtual routing interface MAC address, the device switches the packet on Layer 3 to the destination host on the VLAN.

NOTE

If the Brocade device ARP table does not contain the requested host, the Brocade device forwards the ARP request on Layer 2 to the same VLAN as the one that received the ARP request. Then the device sends an ARP for the destination to the other VLANs that are using the same IP subnet address.

- If the destination is in the same VLAN as the source, the Brocade device does not need to perform a proxy ARP.

To configure multiple VLANs to use the same IP subnet address:

- Configure each VLAN, including adding tagged or untagged ports.
- Configure a separate virtual routing interface for each VLAN, but do not add an IP subnet address to more than one of the virtual routing interfaces.
- Configure the virtual routing interfaces that do not have the IP subnet address to “follow” the virtual routing interface that does have the address.

To configure the VLANs shown in [Figure 85](#), you could enter the following commands.

```
Brocade(config)# vlan 1 by port
Brocade(config-vlan-1)# untagged ethernet 1/1
Brocade(config-vlan-1)# tagged ethernet 1/8
Brocade(config-vlan-1)# router-interface ve 1
```

Syntax: `router-interface ve <number>`

The commands above configure port-based VLAN 1. The VLAN has one untagged port (1/1) and a tagged port (1/8). In this example, all three VLANs contain port 1/8 so the port must be tagged to allow the port to be in multiple VLANs. You can configure VLANs to share a Layer 3 protocol interface regardless of tagging. A combination of tagged and untagged ports is shown in this example to demonstrate that sharing the interface does not change other VLAN features.

Notice that each VLAN still requires a unique virtual routing interface.

The following commands configure port-based VLANs 2 and 3.

```
Brocade(config-vlan-1)# vlan 2 by port
Brocade(config-vlan-2)# untagged ethernet 1/2
Brocade(config-vlan-2)# tagged ethernet 1/8
Brocade(config-vlan-2)# router-interface ve 2
Brocade(config-vlan-2)# vlan 3 by port
Brocade(config-vlan-3)# untagged ethernet 1/5 to 1/6
Brocade(config-vlan-3)# tagged ethernet 1/8
Brocade(config-vlan-3)# router-interface ve 3
```

The following commands configure an IP subnet address on virtual routing interface 1.

```
Brocade(config-vlan-3)# interface ve 1
Brocade(config-vif-1)# ip address 10.0.0.1/24
```

The following commands configure virtual routing interfaces 2 and 3 to “follow” the IP subnet address configured on virtual routing interface 1.

```
Brocade(config-vif-1)# interface ve 2
Brocade(config-vif-2)# ip follow ve 1
Brocade(config-vif-2)# interface ve 3
Brocade(config-vif-3)# ip follow ve 1
```

NOTE

Because virtual routing interfaces 2 and 3 do not have their own IP subnet addresses but instead are “following” virtual routing interface a IP address, you still can configure an IPX or AppleTalk interface on virtual routing interfaces 2 and 3.

VLAN groups and virtual routing interface group

To simplify configuration when you have many VLANs with the same configuration, you can configure VLAN groups and virtual routing interface groups.

NOTE

VLAN groups are supported on Layer 3 switches and Layer 2 switches. Virtual routing interface groups are supported only on Layer 3 switches.

When you create a VLAN group, the VLAN parameters you configure for the group apply to all the VLANs within the group. Additionally, you can easily associate the same IP subnet interface with all the VLANs in a group by configuring a virtual routing interface group with the same ID as the VLAN group.

- The VLAN group feature allows you to create multiple port-based VLANs with identical port members. Because the member ports are shared by all the VLANs within the group, you must add the ports as tagged ports. This feature not only simplifies VLAN configuration but also allows you to have a large number of identically configured VLANs in a startup-config file on the device flash memory module. Normally, a startup-config file with a large number of VLANs might not fit on the flash memory module. By grouping the identically configured VLANs, you can conserve space in the startup-config file so that it fits on the flash memory module.
- The virtual routing interface group feature is useful when you want to configure the same IP subnet address on all the port-based VLANs within a VLAN group. You can configure a virtual routing interface group only after you configure a VLAN group with the same ID. The virtual routing interface group automatically applies to the VLANs in the VLAN group that has the same ID and cannot be applied to other VLAN groups or to individual VLANs.

You can create up to 32 VLAN groups and 32 virtual routing interface groups. A virtual routing interface group always applies only to the VLANs in the VLAN group with the same ID.

NOTE

Depending on the size of the VLAN ID range you want to use for the VLAN group, you might need to allocate additional memory for VLANs. On Layer 3 switches, if you allocate additional memory for VLANs, you also need to allocate the same amount of memory for virtual routing interfaces. This is true regardless of whether you use the virtual routing interface groups. To allocate additional memory, refer to [“Allocating memory for more VLANs or virtual routing interfaces”](#) on page 795.

Configuring a VLAN group

To configure a VLAN group, enter commands such as the following.

```
Brocade(config)# vlan-group 1 vlan 2 to 257
Brocade(config-vlan-group-1)# tagged 1/1 to 1/2
```

The first command in this example begins configuration for VLAN group 1, and assigns VLANs 2 through 257 to the group. The second command adds ports 1/1 and 1/2 as tagged ports. Because all the VLANs in the group share the ports, you must add the ports as tagged ports.

Syntax: `vlan-group <num> vlan <vlan-id> to <vlan-id>`

Syntax: `tagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum>] | ethernet [<slotnum>/]<portnum>]`

The `vlan-group <num>` parameter specifies the VLAN group ID and can be from 1 – 32. The `vlan <vlan-id> to <vlan-id>` parameters specify a contiguous range (a range with no gaps) of individual VLAN IDs. Specify the low VLAN ID first and the high VLAN ID second. The command adds all of the specified VLANs to the VLAN group.

You can add up to 256 VLANs with the command at one time. To add more than 256 VLANs, enter separate commands. For example, to configure VLAN group 1 and add 512 VLANs to the group, enter the following commands.

```
Brocade(config)# vlan-group 1 vlan 2 to 257
Brocade(config-vlan-group-1)# add-vlan 258 to 513
```

NOTE

The device memory must be configured to contain at least the number of VLANs you specify for the higher end of the range. For example, if you specify 2048 as the VLAN ID at the high end of the range, you first must increase the memory allocation for VLANs to 2048 or higher. Additionally, on Layer 3 switches, if you allocate additional memory for VLANs, you also need to allocate the same amount of memory for virtual routing interfaces, before you configure the VLAN groups. This is true regardless of whether you use the virtual routing interface groups. The memory allocation is required because the VLAN groups and virtual routing interface groups have a one-to-one mapping. Refer to [“Allocating memory for more VLANs or virtual routing interfaces”](#) on page 795.

If a VLAN within the range you specify is already configured, or if the range contains more than 256 VLANs, the CLI does not add the group but instead displays an error message.

```
Brocade(config)#vlan-group 1 vlan 2 to 1000
VLAN group 1 is too big. Only 256 vlans are allowed at a time
```

In this case, create the group by specifying a valid contiguous range. Then add more VLANs to the group after the CLI changes to the configuration level for the group. See the following example.

```
Brocade(config)#vlan-group 2 vlan 1000 to 1250
Brocade(config-vlan-group-2)#add-vlan 1251 to 1500
Brocade(config-vlan-group-2)#add-vlan 1501 to 1750
Brocade(config-vlan-group-2)#add-vlan 1751 to 2000
```

You can add or remove individual VLANs or VLAN ranges from the VLAN group at configuration level. For example, if you want to add VLANs 1001 and 1002 to VLAN group 1 and remove VLANs 900 through 1000, enter the following commands.

```
Brocade(config-vlan-group-1)# add-vlan 1001 to 1002
Brocade(config-vlan-group-1)# remove-vlan 900 to 1000
```

Syntax: `add-vlan <vlan-id> [to <vlan-id>]`

Syntax: `remove-vlan <vlan-id> [to <vlan-id>]`

The `<vlan-id> to <vlan-id>` parameters specify a contiguous range (a range with no gaps) of individual VLAN IDs. Specify the low VLAN ID first and the high VLAN ID second. You can add or remove up to 256 VLANs at a time. To add or remove more than 256 VLANs, do so using separate commands. For example, to remove 512 VLANs from VLAN group 1, enter the following commands.

```
Brocade(config-vlan-group-1)# remove-vlan 400 to 654
Brocade(config-vlan-group-1)# remove-vlan 655 to 910
```

Displaying information about VLAN groups

To display VLAN group configuration information, use the **show vlan-group** command.

```
Brocade# show vlan-group
vlan-group 1 vlan 2 to 20
  tagged ethe 1/1 to 1/2
!
vlan-group 2 vlan 21 to 40
  tagged ethe 1/1 to 1/2
!
```

Syntax: **show vlan-group** [`<group-id>`]

The `<group-id>` specifies a VLAN group. If you do not use this parameter, the configuration information for all the configured VLAN groups is displayed.

Configuring a virtual routing interface group

A virtual routing interface group allows you to associate the same IP subnet interface with multiple port-based VLANs. For example, if you associate a virtual routing interface group with a VLAN group, all the VLANs in the group have the IP interface of the virtual routing interface group.

Configuration notes and feature limitations for virtual routing interface group

- When you configure a virtual routing interface group, all members of the group have the same IP subnet address. This feature is useful in collocation environments where the device has many IP addresses and you want to conserve the IP address space.
- The **group-router-interface** command creates router interfaces for each VLAN in the VLAN group by using the VLAN IDs of each of the VLANs as the corresponding virtual interface number. Therefore, if a VLAN group contains VLAN IDs greater than the maximum virtual interface number allowed, the **group-router-interface** command will be rejected.

CLI syntax for virtual routing interface group

To configure a virtual routing interface group, enter commands such as the following.

```
Brocade(config)# vlan-group 1
Brocade(config-vlan-group-1)# group-router-interface
Brocade(config-vlan-group-1)# exit
Brocade(config)# interface group-ve 1
Brocade(config-vif-group-1)# ip address 10.10.10.1/24
```

These commands enable VLAN group 1 to have a group virtual routing interface, then configure virtual routing interface group 1. The software always associates a virtual routing interface group only with the VLAN group that has the same ID. In this example, the VLAN group ID is 1, so the corresponding virtual routing interface group also must have ID 1.

Syntax: `group-router-interface`

Syntax: `interface group-ve <num>`

Syntax: `[no] ip address <ip-addr> <ip-mask> [secondary]`

or

Syntax: `[no] ip address <ip-addr>/<mask-bits> [secondary]`

The **router-interface-group** command enables a VLAN group to use a virtual routing interface group. Enter this command at the configuration level for the VLAN group. This command configures the VLAN group to use the virtual routing interface group that has the same ID as the VLAN group. You can enter this command when you configure the VLAN group for the first time or later, after you have added tagged ports to the VLAN and so on.

The `<num>` parameter in the **interface group-ve <num>** command specifies the ID of the VLAN group with which you want to associate this virtual routing interface group. The VLAN group must already be configured and enabled to use a virtual routing interface group. The software automatically associates the virtual routing interface group with the VLAN group that has the same ID. You can associate a virtual routing interface group only with the VLAN group that has the same ID.

NOTE

IPv6 is not supported with **group-ve**.

NOTE

FCX devices do not support ACLs with **group-ve**.

NOTE

FastIron devices support **group-ve** with OSPF, VRRP v2 and VRRP-E v2 protocols only.

The syntax and usage for the **ip address** command is the same as when you use the command at the interface level to add an IP interface.

Displaying the VLAN group and virtual routing interface group information

To verify configuration of VLAN groups and virtual routing interface groups, display the running-config file. If you have saved the configuration to the startup-config file, you also can verify the configuration by displaying the startup-config file. The following example shows the running-config information for the VLAN group and virtual routing interface group configured in the previous examples. The information appears in the same way in the startup-config file.

```
Brocade# show running-config
```

```
lines not related to the VLAN group omitted...
```

```
vlan-group 1 vlan 2 to 20
  add-vlan 1001 to 1002
  tagged ethe 1/1 to 1/2
```



```
router-interface-group

lines not related to the virtual routing interface group omitted...

interface group-ve 1
 ip address 10.10.10.1 255.255.255.0
```

NOTE

If you have enabled display of subnet masks in CIDR notation, the IP address information is shown as follows: 10.10.10.1/24.

Allocating memory for more VLANs or virtual routing interfaces

Brocade Layer 2 and Layer 3 Switches support up to 4095 VLANs. In addition, Layer 3 switches support up to 512 virtual routing interfaces.

The number of VLANs and virtual routing interfaces supported on your product depends on the device and, for Chassis devices, the amount of DRAM on the management module. [Table 141](#) lists the default and configurable maximum numbers of VLANs and virtual routing interfaces for Layer 2 and Layer 3 switches. Unless otherwise noted, the values apply to both types of switches.

TABLE 141 VLAN and virtual routing interface support

VLANs		Virtual routing interfaces	
Default maximum	Configurable maximum	Default maximum	Configurable maximum
64	4094	255	512

NOTE

If many of your VLANs will have an identical configuration, you might want to configure VLAN groups and virtual routing interface groups after you increase the system capacity for VLANs and virtual routing interfaces. Refer to [“VLAN groups and virtual routing interface group”](#) on page 791.

Increasing the number of VLANs you can configure

NOTE

Although you can specify up to 4095 VLANs, you can configure only 4094 VLANs. VLAN ID 4094 is reserved for use by the Single Spanning Tree feature.

To increase the maximum number of VLANs you can configure, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# system-max vlan 2048
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload
```

Syntax: `system-max vlan <num>`

The `<num>` parameter indicates the maximum number of VLANs. The range of valid values depends on the device you are configuring. Refer to [Table 141](#).

Increasing the number of virtual routing interfaces you can configure

To increase the maximum number of virtual routing interfaces you can configure, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# system-max virtual-interface 512
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload
```

Syntax: `system-max virtual-interface <num>`

The <num> parameter indicates the maximum number of virtual routing interfaces. The range of valid values depends on the device you are configuring. Refer to [Table 141](#).

Super-aggregated VLAN configuration

You can aggregate multiple VLANs within another VLAN. This feature allows you to construct Layer 2 paths and channels. This feature is particularly useful for Virtual Private Network (VPN) applications in which you need to provide a private, dedicated Ethernet connection for an individual client to transparently reach its subnet across multiple networks.

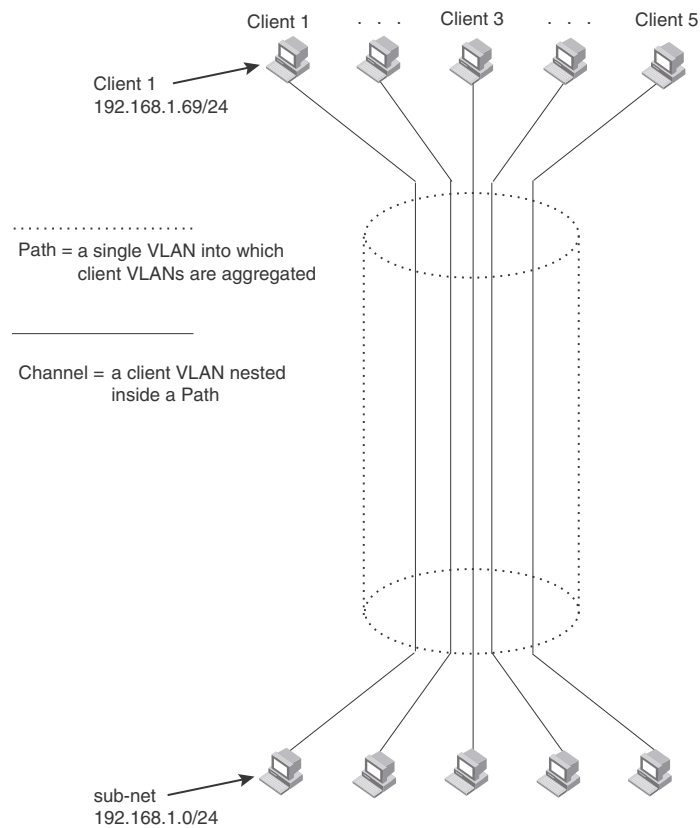
Conceptually, the paths and channels are similar to Asynchronous Transfer Mode (ATM) paths and channels. A path contains multiple channels, each of which is a dedicated circuit between two end points. The two devices at the end points of the channel appear to each other to be directly attached. The network that connects them is transparent to the two devices.

You can aggregate up to 4094 VLANs within another VLAN. This provides a total VLAN capacity on one Brocade device of 16,760,836 channels ($4094 * 4094$).

The devices connected through the channel are not visible to devices in other channels. Therefore, each client has a private link to the other side of the channel.

The feature allows point-to-point and point-to-multipoint connections.

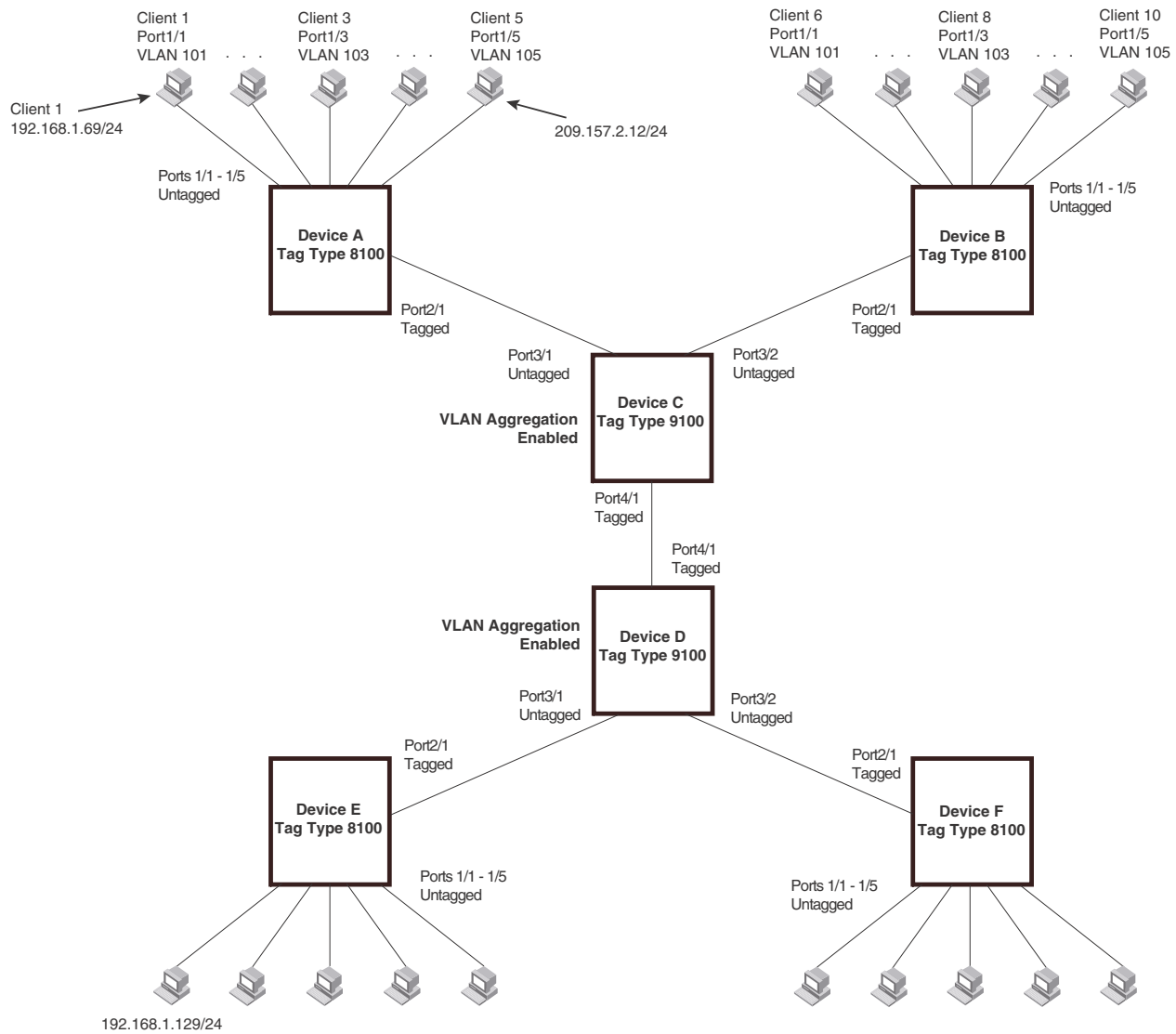
[Figure 86](#) shows a conceptual picture of the service that aggregated VLANs provide. Aggregated VLANs provide a path for multiple client channels. The channels do not receive traffic from other channels. Thus, each channel is a private link.

FIGURE 86 Conceptual model of the super aggregated VLAN application

Each client connected to the edge device is in its own port-based VLAN, which is like an ATM channel. All the clients' VLANs are aggregated by the edge device into a single VLAN for connection to the core. The single VLAN that aggregates the clients' VLANs is like an ATM path.

The device that aggregates the VLANs forwards the aggregated VLAN traffic through the core. The core can consist of multiple devices that forward the aggregated VLAN traffic. The edge device at the other end of the core separates the aggregated VLANs into the individual client VLANs before forwarding the traffic. The edge devices forward the individual client traffic to the clients. For the clients' perspective, the channel is a direct point-to-point link.

Figure 87 shows an example application that uses aggregated VLANs. This configuration includes the client connections shown in Figure 86.

FIGURE 87 Example of a super aggregated VLAN application

In this example, a collocation service provides private channels for multiple clients. Although the same devices are used for all the clients, the VLANs ensure that each client receives its own Layer 2 broadcast domain, separate from the broadcast domains of other clients. For example, client 1 cannot ping client 5.

The clients at each end of a channel appear to each other to be directly connected and thus can be on the same subnet and use network services that require connection to the same subnet. In this example, client 1 is in subnet 192.168.1.0/24 and so is the device at the other end of client 1 channel.

Because each VLAN configured on the core devices is an aggregate of multiple client VLANs, the aggregated VLANs greatly increase the number of clients a core device can accommodate.

This example shows a single link between the core devices. However, you can use a trunk group to add link-level redundancy.

Configuration notes for aggregated VLANs

- This feature is not supported on the 48-port 10/100/1000 Mbps (RJ45) Ethernet POE interface module (SX-FI48GPP).
- Super Aggregated VLANs and VSRP are not supported together on the same device.

Configuring aggregated VLANs

To configure aggregated VLANs, perform the following tasks:

- On each edge device, configure a separate port-based VLAN for each client connected to the edge device. In each client VLAN:
 - Add the port connected to the client as an untagged port.
 - Add the port connected to the core device (the device that will aggregate the VLANs) as a tagged port. This port must be tagged because all the client VLANs share the port as an uplink to the core device.
- On each core device:
 - Enable VLAN aggregation. This support allows the core device to add an additional tag to each Ethernet frame that contains a VLAN packet from the edge device. The additional tag identifies the aggregate VLAN (the path). However, the additional tag can cause the frame to be longer than the maximum supported frame size. The larger frame support allows Ethernet frames up to 1530 bytes long.
 - For FWS devices, the frame size is limited to 1522, not 1530, bytes. To allow frames larger than 1522, you must enable jumbo frames. To globally enable jumbo support, enter commands such as the following.

```
Brocade(config)# jumbo
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload
```

NOTE

Enable the VLAN aggregation option only on the core devices.

- Configure a VLAN tag type (tag ID) that is different than the tag type used on the edge devices. If you use the default tag type (8100) on the edge devices, set the tag type on the core devices to another value, such as 9100. The tag type must be the same on all the core devices. The edge devices also must have the same tag type but the type must be different from the tag type on the core devices.

NOTE

You can enable the Spanning Tree Protocol (STP) on the edge devices or the core devices, but not both. If you enable STP on the edge devices and the core devices, STP will prevent client traffic from travelling through the core to the other side.

Configuring aggregated VLANs on an edge device

To configure the aggregated VLANs on device A in [Figure 87](#) on page 798, enter the following commands.

```
Brocade(config)# vlan 101 by port
Brocade(config-vlan-101)# tagged ethernet 2/1
Brocade(config-vlan-101)# untagged ethernet 1/1
Brocade(config-vlan-101)# exit
Brocade(config)# vlan 102 by port
Brocade(config-vlan-102)# tagged ethernet 2/1
Brocade(config-vlan-102)# untagged ethernet 1/2
Brocade(config-vlan-102)# exit
Brocade(config)# vlan 103 by port
Brocade(config-vlan-103)# tagged ethernet 2/1
Brocade(config-vlan-103)# untagged ethernet 1/3
Brocade(config-vlan-103)# exit
Brocade(config)# vlan 104 by port
Brocade(config-vlan-104)# tagged ethernet 2/1
Brocade(config-vlan-104)# untagged ethernet 1/4
Brocade(config-vlan-104)# exit
Brocade(config)# vlan 105 by port
Brocade(config-vlan-105)# tagged ethernet 2/1
Brocade(config-vlan-105)# untagged ethernet 1/5
Brocade(config-vlan-105)# exit
Brocade(config)# write memory
```

Syntax: [no] vlan <vlan-id> [by port]

Syntax: [no] tagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Syntax: [no] untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Use the **tagged** command to add the port that the device uses for the uplink to the core device. Use the **untagged** command to add the ports connected to the individual clients.

Configuring aggregated VLANs on a core device

To configure the aggregated VLANs on device C in [Figure 87](#) on page 798, enter the following commands.

```
Brocade(config)# tag-type 9100
Brocade(config)# aggregated-vlan
Brocade(config)# vlan 101 by port
Brocade(config-vlan-101)# tagged ethernet 4/1
Brocade(config-vlan-101)# untagged ethernet 3/1
Brocade(config-vlan-101)# exit
Brocade(config)# vlan 102 by port
Brocade(config-vlan-102)# tagged ethernet 4/1
Brocade(config-vlan-102)# untagged ethernet 3/2
Brocade(config-vlan-102)# exit
Brocade(config)# write memory
```

Syntax: [no] tag-type <num>

Syntax: [no] aggregated-vlan

The `<num>` parameter specifies the tag type can be a hexadecimal value from 0 – ffff. The default is 8100.

Verifying the aggregated VLAN configuration

You can verify the VLAN, VLAN aggregation option, and tag configuration by viewing the running-config. To display the running-config, enter the **show running-config** command from any CLI prompt. After you save the configuration changes to the startup-config, you also can display the settings in that file by entering the **show configuration** command from any CLI prompt.

Complete CLI examples for aggregated VLANs

The following sections show all the Aggregated VLAN configuration commands on the devices in [Figure 87](#) on page 798.

NOTE

In these examples, the configurations of the edge devices (A, B, E, and F) are identical. The configurations of the core devices (C and D) also are identical. The aggregated VLAN configurations of the edge and core devices on one side must be symmetrical (in fact, a mirror image) to the configurations of the devices on the other side. For simplicity, the example in [Figure 87](#) on page 798 is symmetrical in terms of the port numbers. This allows the configurations for both sides of the link to be the same. If your configuration does not use symmetrically arranged port numbers, the configurations should not be identical but must use the correct port numbers.

Commands for configuring aggregated VLANs on device A

```
BrocadeA(config)# vlan 101 by port
BrocadeA(config-vlan-101)# tagged ethernet 2/1
BrocadeA(config-vlan-101)# untagged ethernet 1/1
BrocadeA(config-vlan-101)# exit
BrocadeA(config)# vlan 102 by port
BrocadeA(config-vlan-102)# tagged ethernet 2/1
BrocadeA(config-vlan-102)# untagged ethernet 1/2
BrocadeA(config-vlan-102)# exit
BrocadeA(config)# vlan 103 by port
BrocadeA(config-vlan-103)# tagged ethernet 2/1
BrocadeA(config-vlan-103)# untagged ethernet 1/3
BrocadeA(config-vlan-103)# exit
BrocadeA(config)# vlan 104 by port
BrocadeA(config-vlan-104)# tagged ethernet 2/1
BrocadeA(config-vlan-104)# untagged ethernet 1/4
BrocadeA(config-vlan-104)# exit
BrocadeA(config)# vlan 105 by port
BrocadeA(config-vlan-105)# tagged ethernet 2/1
BrocadeA(config-vlan-105)# untagged ethernet 1/5
BrocadeA(config-vlan-105)# exit
vA(config)# write memory
```

Commands for configuring aggregated VLANs on device B

The commands for configuring device B are identical to the commands for configuring device A. Notice that you can use the same channel VLAN numbers on each device. The devices that aggregate the VLANs into a path can distinguish between the identically named channel VLANs based on the ID of the path VLAN.

```
BrocadeB(config)# vlan 101 by port
BrocadeB(config-vlan-101)# tagged ethernet 2/1
BrocadeB(config-vlan-101)# untagged ethernet 1/1
BrocadeB(config-vlan-101)# exit
BrocadeB(config)# vlan 102 by port
BrocadeB(config-vlan-102)# tagged ethernet 2/1
BrocadeB(config-vlan-102)# untagged ethernet 1/2
BrocadeB(config-vlan-102)# exit
BrocadeB(config)# vlan 103 by port
BrocadeB(config-vlan-103)# tagged ethernet 2/1
BrocadeB(config-vlan-103)# untagged ethernet 1/3
BrocadeB(config-vlan-103)# exit
BrocadeB(config)# vlan 104 by port
BrocadeB(config-vlan-104)# tagged ethernet 2/1
BrocadeB(config-vlan-104)# untagged ethernet 1/4
BrocadeB(config-vlan-104)# exit
BrocadeB(config)# vlan 105 by port
BrocadeB(config-vlan-105)# tagged ethernet 2/1
BrocadeB(config-vlan-105)# untagged ethernet 1/5
BrocadeB(config-vlan-105)# exit
BrocadeB(config)# write memory
```

Commands for configuring aggregated VLANs on device C

Because device C is aggregating channel VLANs from devices A and B into a single path, you need to change the tag type and enable VLAN aggregation.

```
BrocadeC(config)# tag-type 9100
BrocadeC(config)# aggregated-vlan
BrocadeC(config)# vlan 101 by port
BrocadeC(config-vlan-101)# tagged ethernet 4/1
BrocadeC(config-vlan-101)# untagged ethernet 3/1
BrocadeC(config-vlan-101)# exit
BrocadeC(config)# vlan 102 by port
BrocadeC(config-vlan-102)# tagged ethernet 4/1
BrocadeC(config-vlan-102)# untagged ethernet 3/2
BrocadeC(config-vlan-102)# exit
BrocadeC(config)# write memory
```

Commands for configuring aggregated VLANs on device D

Device D is at the other end of path and separates the channels back into individual VLANs. The tag type must be the same as tag type configured on the other core device (Device C). In addition, VLAN aggregation also must be enabled.

```
BrocadeD(config)# tag-type 9100
BrocadeD(config)# aggregated-vlan
BrocadeD(config)# vlan 101 by port
BrocadeD(config-vlan-101)# tagged ethernet 4/1
BrocadeD(config-vlan-101)# untagged ethernet 3/1
BrocadeD(config-vlan-101)# exit
```



```

BrocadeD(config)# vlan 102 by port
BrocadeD(config-vlan-102)# tagged ethernet 4/1
BrocadeD(config-vlan-102)# untagged ethernet 3/2
BrocadeD(config-vlan-102)# exit
BrocadeD(config)# write memory

```

Commands for configuring aggregated VLANs on device E

Because the configuration in [Figure 87](#) on page 798 is symmetrical, the commands for configuring device E are identical to the commands for configuring device A.

```

BrocadeE(config)# vlan 101 by port
BrocadeE(config-vlan-101)# tagged ethernet 2/1
BrocadeE(config-vlan-101)# untagged ethernet 1/1
BrocadeE(config-vlan-101)# exit
BrocadeE(config)# vlan 102 by port
BrocadeE(config-vlan-102)# tagged ethernet 2/1
BrocadeE(config-vlan-102)# untagged ethernet 1/2
BrocadeE(config-vlan-102)# exit
BrocadeE(config)# vlan 103 by port
BrocadeE(config-vlan-103)# tagged ethernet 2/1
BrocadeE(config-vlan-103)# untagged ethernet 1/3
BrocadeE(config-vlan-103)# exit
BrocadeE(config)# vlan 104 by port
BrocadeE(config-vlan-104)# tagged ethernet 2/1
BrocadeE(config-vlan-104)# untagged ethernet 1/4
BrocadeE(config-vlan-104)# exit
BrocadeE(config)# vlan 105 by port
BrocadeE(config-vlan-105)# tagged ethernet 2/1
BrocadeE(config-vlan-105)# untagged ethernet 1/5
BrocadeE(config-vlan-105)# exit
BrocadeE(config)# write memory

```

Commands for configuring aggregated VLANs on device F

The commands for configuring device F are identical to the commands for configuring device E. In this example, Because the port numbers on each side of the configuration in [Figure 87](#) on page 798 are symmetrical, the configuration of device F is also identical to the configuration of device A and device B.

```

BrocadeF(config)# vlan 101 by port
BrocadeF(config-vlan-101)# tagged ethernet 2/1
BrocadeF(config-vlan-101)# untagged ethernet 1/1
BrocadeF(config-vlan-101)# exit
BrocadeF(config)# vlan 102 by port
BrocadeF(config-vlan-102)# tagged ethernet 2/1
BrocadeF(config-vlan-102)# untagged ethernet 1/2
BrocadeF(config-vlan-102)# exit
BrocadeF(config)# vlan 103 by port
BrocadeF(config-vlan-103)# tagged ethernet 2/1
BrocadeF(config-vlan-103)# untagged ethernet 1/3
BrocadeF(config-vlan-103)# exit
BrocadeF(config)# vlan 104 by port
BrocadeF(config-vlan-104)# tagged ethernet 2/1
BrocadeF(config-vlan-104)# untagged ethernet 1/4
BrocadeF(config-vlan-104)# exit
BrocadeF(config)# vlan 105 by port

```

```

BrocadeF(config-vlan-105)# tagged ethernet 2/1
BrocadeF(config-vlan-105)# untagged ethernet 1/5
BrocadeF(config-vlan-105)# exit
BrocadeF(config)# write memory

```

802.1ad tagging configuration

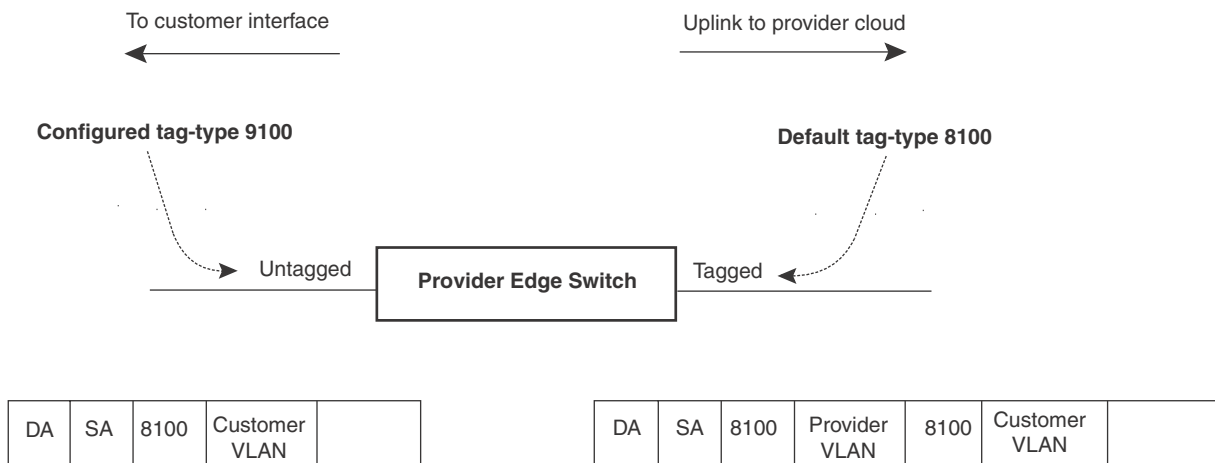
802.1ad tagging provides finer granularity for configuring 802.1Q tagging, enabling you to configure 802.1Q tag-types on a group of ports. This feature allows you to create two identical 802.1Q tags (802.1ad tagging) on a single device. This enhancement improves SAV interoperability between Brocade devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

NOTE

Brocade devices treat a double-tagged Ethernet frame as a Layer 2 only frame. The packets are not inspected for Layer 3 and Layer 4 information, and operations are not performed on the packet utilizing Layer 3 or Layer 4 information.

Figure 88 shows an example application with 802.1ad tagging.

FIGURE 88 802.1ad configuration example



In Figure 88, the untagged ports (to customer interfaces) accept frames that have any 802.1Q tag other than the configured tag-type 9100. These packets are considered untagged on this incoming port and are re-tagged when they are sent out of the uplink towards the provider. The 802.1Q tag-type on the uplink port is 8100, so the Brocade device will switch the frames to the uplink device with an additional 8100 tag, thereby supporting devices that only support this method of VLAN tagging.

Configuration rules for 802.1ad tagging

- Because the uplink (to the provider cloud) and the edge link (to the customer port) must have different 802.1Q tags, make sure the uplink and edge link are in different port regions. Refer to [“About port regions”](#) on page 556 for a list of valid port regions.
- On devices that support port regions, if you configure a port with an 802.1Q tag-type, the Brocade device automatically applies the 802.1Q tag-type to all ports within the same port region. Likewise, if you remove the 802.1Q tag-type from a port, the Brocade device automatically removes the 802.1Q tag-type from all ports within the same port region.
- Q-in-Q and SAV are not supported on the following :
 - 48-port 10/100/1000 Mbps (RJ45) Ethernet POE interface module (SX-FI48GPP)
 - 2-port 10-Gigabit Ethernet interface module (SX-FI-2XG)
 - 8-port 10-Gigabit Ethernet interface module (SX-FI-8XG)
 - 24-port Gigabit Ethernet fiber interface module (SX-FI-24HF)
 - 24-port Gigabit Ethernet copper interface module with PoE+ (SX-FI-24GPP)
- FastIron X Series devices support one configured tag-type per device along with the default tag-type of 8100. For example, if you configure an 802.1Q tag of 9100 on ports 1 – 12, then later configure an 802.1Q tag of 5100 on port 15, the device automatically applies the 5100 tag to all ports in the same port region as port 15, and also changes the 802.1Q tag-type on ports 1 – 12 to 5100.
- 802.1ad tagging and VSRP are not supported together on the same device.
- In addition to **tag-type**, FastIron WS and Brocade FCX Series devices support **tag-profile**. For more information, refer to [“Configuring 802.1ad tag profiles”](#) on page 808.

Enabling 802.1ad tagging

To enable 802.1ad tagging, configure an 802.1Q tag on the untagged edge links (the customer ports) to any value other than the 802.1Q tag for incoming traffic. For example, in [Figure 89](#), the 802.1Q tag on the untagged edge links (ports 11 and 12) is 9100, whereas, the 802.1Q tag for incoming traffic is 8100.

To configure 802.1 ad tagging as shown in [Figure 89](#), enter commands such as the following on the untagged edge links of devices C and D.

```
Brocade(config)# tag-type 9100 ethernet 11 to 12
Brocade(config)# aggregated-vlan
```

Note that because ports 11 and 12 belong to the port region 1 – 12, the 802.1Q tag actually applies to ports 1 – 12.

Syntax: [no] tag-type <num> [ethernet <port> [to <port>]]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

802.1ad tagging configuration

The **ethernet** *<port>* to *<port>* parameter specifies the ports that will use the defined 802.1Q tag. This parameter operates with the following rules:

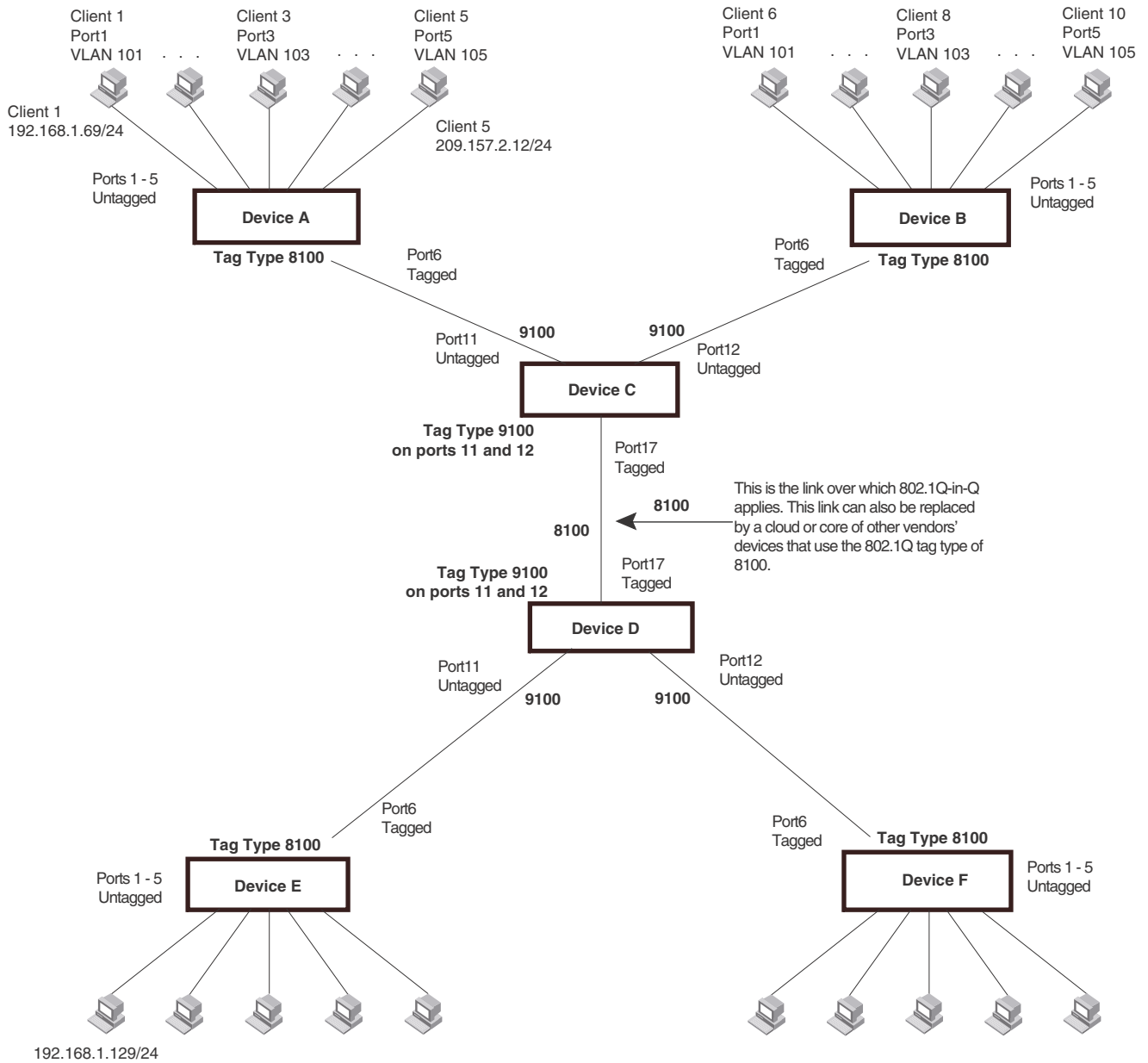
- If you specify a single port number, the 802.1Q tag applies to all ports within the port region. For example, if you enter the command **tag-type 9100 ethernet 1**, the Brocade device automatically applies the 802.1Q tag to ports 1 – 12 because all of these ports are in the same port region. You can use the **show running-config** command to view how the command has been applied.
- If you do not specify a port or range of ports, the 802.1Q tag applies to all Ethernet ports on the device.

Example 802.1ad configuration

[Figure 89](#) shows an example 802.1ad configuration.

FIGURE 89 Example 802.1ad configuration

802.1ad tagging configuration



Configuring 802.1ad tag profiles

NOTE

802.1ad tag profiles are not supported on FastIron X Series devices.

The 802.1ad tagging feature supports a **tag-profile** command that allows you to add a tag profile with a value of 0 to 0xffff in addition to the default tag-type 0x8100. This enhancement also allows you to add a tag profile for a single port, or to direct a group of ports to a globally-configured tag profile.

Configuration notes for 802.1ad tagging

- One global tag profile with a number between 0 and 0xffff can be configured on stackable devices.
- On individual ports, if **tag-profile** is enabled, it points to the global tag profile.
- **Tag-profile** can also be enabled for provisional ports.
- **Tag-type** and **tag-profile** cannot be configured at the same time. You will see the message “un-configure the tag-type to set the tag-profile.” If **tag-type** is already configured, you will need to unconfigure it and then add the **tag-profile**.
- Do not use the **tag-type** command in conjunction with the **tag-profile** command. If a **tag-type** has already been configured and you try to use the **tag-profile** command, you will see an error message telling you to remove the tag-type before you add the tag-profile.
- For devices operating in an IronStack topology, when a tag-type for a port is changed, the tag-type for all of the ports on a stack unit also changes. Because of this limitation, SAV and Q-in-Q cannot be used at the same time on stacking devices.

CLI Syntax for 802.1ad tagging

To add a global tag-profile enter the following command.

```
Brocade(config)# tag-profile 9500
```

This command adds a profile in addition to the default profile of 0x8100.

Syntax: [no] **tag-profile** <tag-no>

where <tag-no> can be 0x8100 (the default) or 0xffff.

To enable the new profile on individual ports, enter commands similar to the following.

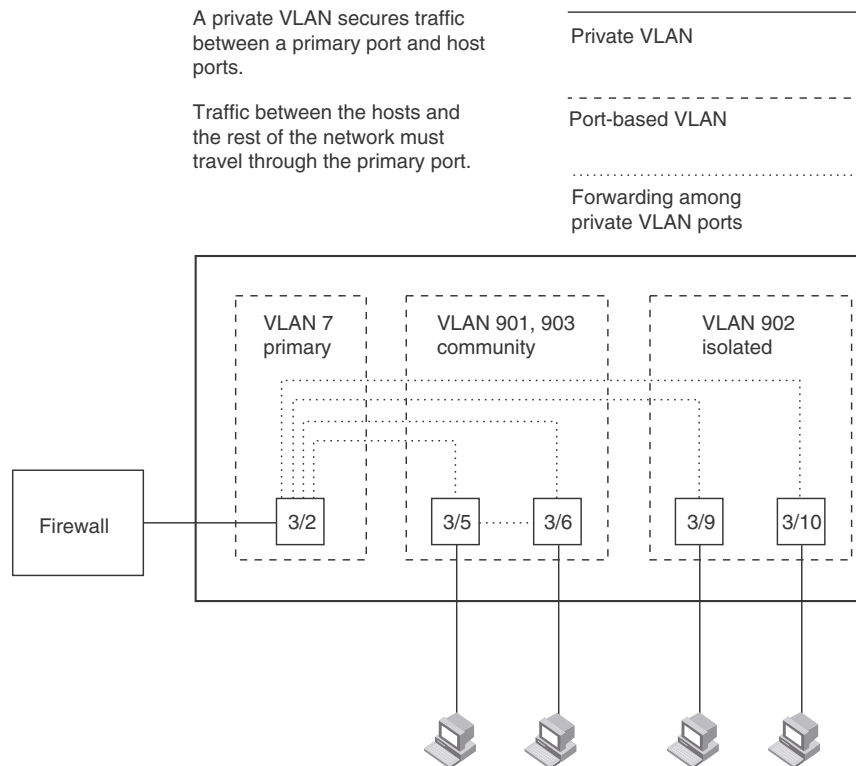
```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e1000-1/1/1)# tag-profile enable
Brocade(config-mif-1/1/1,1/2/1)# tag-profile enable
```

Syntax: [no] **tag-profile enable**

Private VLAN configuration

A private VLAN (PVLAN) is a VLAN that has the properties of standard Layer 2 port-based VLANs but also provides additional control over flooding packets on a VLAN. [Figure 90](#) shows an example of an application using a PVLAN.

FIGURE 90 PVLAN used to secure communication between a workstation and servers



This example uses a PVLAN to secure traffic between hosts and the rest of the network through a firewall. Five ports in this example are members of a PVLAN. The first port (port 3/2) is attached to a firewall. The next four ports (ports 3/5, 3/6, 3/9, and 3/10) are attached to hosts that rely on the firewall to secure traffic between the hosts and the rest of the network. In this example, two of the hosts (on ports 3/5 and 3/6) are in a community PVLAN, and thus can communicate with one another as well as through the firewall. The other two hosts (on ports 3/9 and 3/10), are in an isolated VLAN and thus can communicate only through the firewall. The two hosts are secured from communicating with one another even though they are in the same VLAN.

By default, in the FCX platform, the device will forward broadcast, unregistered multicast, and unknown unicast packets from outside sources into the PVLAN.

By default, in FastIron platforms other than the FCX, the device will not forward broadcast, unregistered multicast, and unknown unicast packets from outside sources into the PVLAN. If needed, you can override this behavior for broadcast packets, unknown-unicast packets, or both. (Refer to [“Displaying PVLAN information”](#) on page 825.)

You can configure a combination of the following types of PVLANS:

- Primary – The primary PVLAN ports are “promiscuous”. They can communicate with all the isolated PVLAN ports and community PVLAN ports in the isolated and community VLANs that are mapped to the promiscuous port.
- Isolated – Broadcasts and unknown unicasts received on isolated ports are sent only to the promiscuous ports and switch – switch ports. They are not flooded to other ports in the isolated VLAN.

NOTE

On ICX 6430 and ICX 6450 devices, however, private VLANs will act as a normal VLAN and will flood unknown destinations, broadcast and multicast traffic to all ports in the VLAN if the primary VLAN does not have the PVLAN mapping that defines the uplink port for the isolated VLAN.

- Community – Broadcasts and unknown unicasts received on community ports are sent to the primary port and also are flooded to the other ports in the community VLAN.

Each PVLAN must have a primary VLAN. The primary VLAN is the interface between the secured ports and the rest of the network. The PVLAN can have any combination of community and isolated VLANs.

As with regular VLANs, PVLANS can span multiple switches. The PVLAN is treated like any other VLAN by the PVLAN-trunk ports. The PVLAN-trunk port is added to both the primary and the secondary VLANs as a tagged member through the **pvlan-trunk** command. [Figure 91](#) shows an example of a PVLAN network across switches:

- Broadcast, unknown unicast or unregistered multicast traffic from the primary VLAN port is forwarded to all ports in isolated and community VLANs in both the switches.
- Broadcast, unknown unicast or unregistered multicast traffic from an isolated port in switch A is not forwarded to an isolated port in switch A. It will not be forwarded to an isolated port in switch B across the PVLAN-trunk port.
- Broadcast, unknown unicast or unregistered multicast traffic from a community port in switch A will be forwarded to a community port in switch B through the PVLAN-trunk port. It is forwarded to the promiscuous ports and switch – switch ports of the primary VLAN.

FIGURE 91 PVLAN across switches

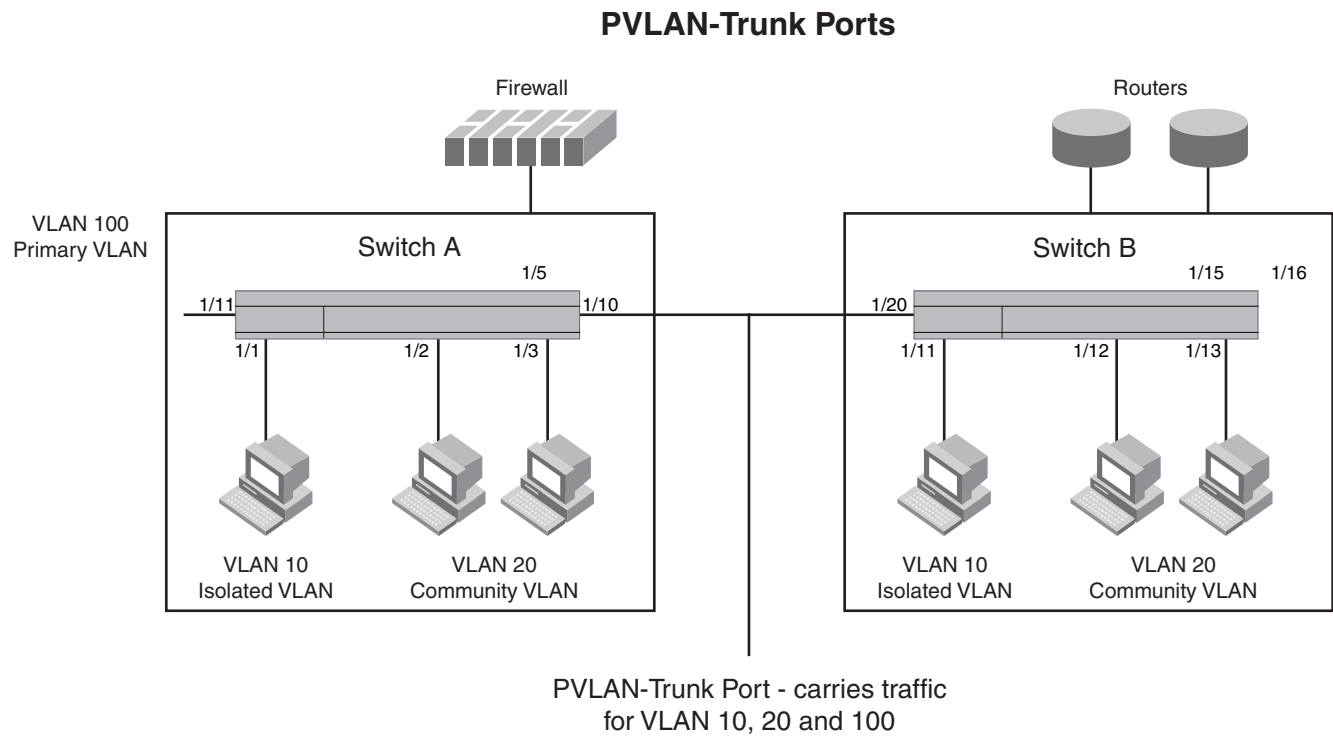


Figure 92 shows an example PVLAN network with tagged switch-switch link ports.

FIGURE 92 Example PVLAN network with tagged ports

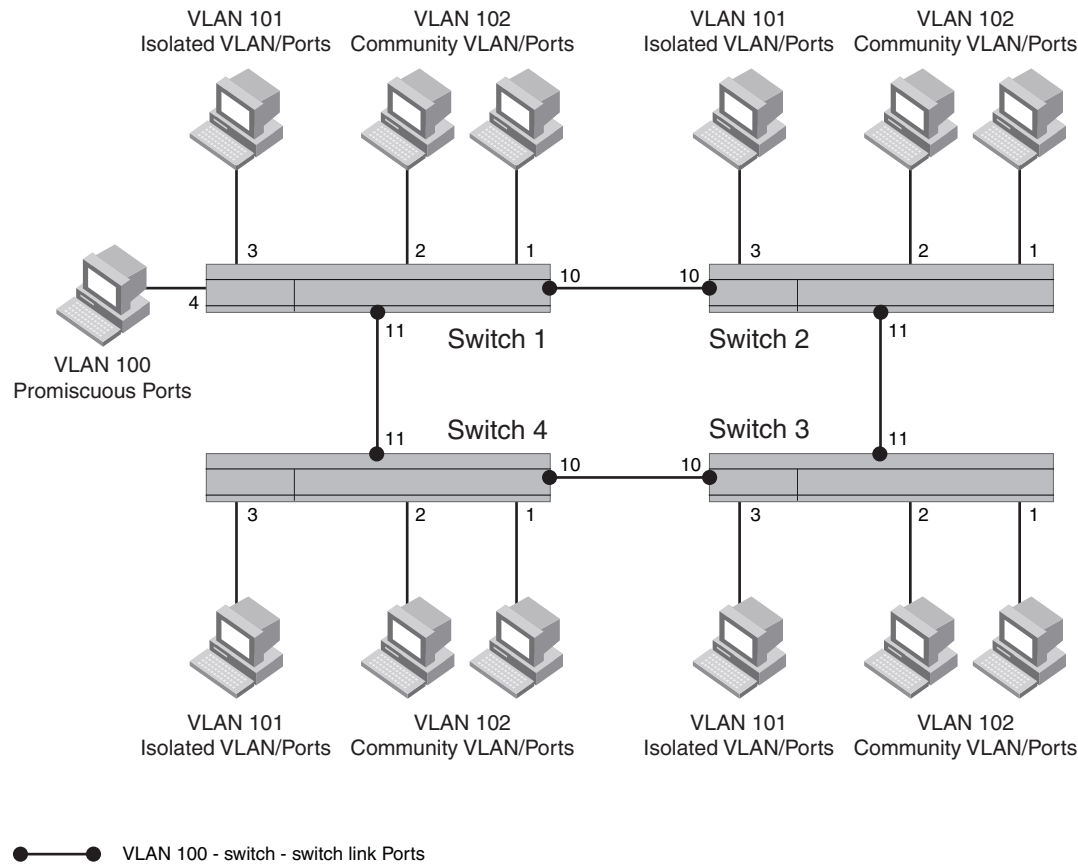


Table 142 lists the differences between PVLANS and standard VLANs.

TABLE 142 Comparison of PVLANS and standard port-based VLANs

Forwarding behavior	Private VLANs	Standard VLANs
All ports within a VLAN constitute a common layer broadcast domain	No	Yes
Broadcasts and unknown unicasts are forwarded to all the VLAN ports by default	No (isolated VLAN) Yes (community VLAN)	Yes
Known unicasts	Yes	Yes

Configuration notes for PVLANS and standard VLANs

- PVLANS are supported on untagged ports on all FastIron platforms. PVLANS are also supported on tagged ports on the FCX platform only.
- Normally, in any port-based VLAN, the Brocade device floods unknown unicast, unregistered multicast, and broadcast packets in hardware, although selective packets, such as IGMP, may be sent only to the CPU for analysis, based on the IGMP snooping configuration. When protocol or subnet VLANs are enabled, or if PVLAN mappings are enabled, the Brocade device will flood unknown unicast, unregistered multicast, and broadcast packets in software. The flooding of broadcast or unknown unicast from the community or isolated VLANs to other secondary VLANs will be governed by the PVLAN forwarding rules. The switching is done in hardware and thus the CPU does not enforce packet restrictions. The hardware forwarding behavior is supported on the FCX and Turbolron platforms only.
- There is currently no support for IGMP snooping within PVLANS. In order for clients in PVLANS to receive multicast traffic, IGMP snooping must be disabled so that all multicast packets are treated as unregistered packets and are flooded in software to all the ports.
- The FastIron forwards all known unicast traffic in hardware. This differs from the way the BigIron implements PVLANS, in that the BigIron uses the CPU to forward packets on the primary VLAN "promiscuous" port. In addition, on the BigIron, support for the hardware forwarding sometimes results in multiple MAC address entries for the same MAC address in the device MAC address table. On the FastIron, multiple MAC entries do not appear in the MAC address table because the FastIron transparently manages multiple MAC entries in hardware.
- To configure a PVLAN, configure each of the component VLANs (isolated, community, and public) as a separate port-based VLAN:
 - Use standard VLAN configuration commands to create the VLAN and add ports.
 - Identify the PVLAN type (isolated, community, or public)
 - For the primary VLAN, map the other PVLANS to the ports in the primary VLAN
- A primary VLAN can have multiple ports. All these ports are active, but the ports that will be used depends on the PVLAN mappings. Also, secondary VLANs (isolated and community VLANs) can be mapped to multiple primary VLAN ports.
- You can configure PVLANS and dual-mode VLAN ports on the same device. However, the dual-mode VLAN ports cannot be members of PVLANS.
- VLAN identifiers configured as part of a PVLAN (primary, isolated, or community) should be consistent across the switched network. The same VLAN identifiers cannot be configured as a normal VLAN or a part of any other PVLAN.
- Promiscuous and switch-switch link ports are member ports of the primary VLAN only. All switch-switch link ports are tagged ports.
- Member ports of isolated and community VLANs cannot be member ports of any other VLAN.
- All member ports that are part of the PVLAN (isolated or secondary) will perform VLAN classification based on the PVLAN ID (PVID) only (no VLAN classification by port, protocol, ACL and so on, if any).
- PVST, when needed in PVLANS, should be enabled on all (primary and secondary) private VLANs.

Configuring the primary VLAN

To configure a primary VLAN, enter commands such as the following.

```
Brocade(config)# vlan 7
Brocade(config-vlan-7)# untagged ethernet 3/2
Brocade(config-vlan-7)# pvlan type primary
Brocade(config-vlan-7)# pvlan mapping 901 ethernet 3/2
```

These commands create port-based VLAN 7, add port 3/2 as an untagged port, identify the VLAN as the primary VLAN in a PVLAN, and map the other secondary VLANs to the ports in this VLAN.

To map the secondary VLANs to the primary VLAN and to configure the tagged switch link port, enter commands such as the following.

```
Brocade(config)# vlan 100
Brocade(config-vlan-100)# tagged ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# untagged ethernet 1/1/4
Brocade(config-vlan-100)# pvlan type primary
Brocade(config-vlan-100)# pvlan mapping 101 ethernet 1/1/4
Brocade(config-vlan-100)# pvlan mapping 102 ethernet 1/1/4
Brocade(config-vlan-100)# pvlan pvlan-trunk 101 ethernet 1/1/10 to 1/1/11
```

These commands create port-based VLAN 100, add port 1/1/10 to 1/1/11 as a tagged port, identify the VLAN as the primary VLAN in a PVLAN, map the other secondary VLANs to the ports in this VLAN, and configure the tagged switch link port.

Syntax: **untagged ethernet** [<stack-unit>/<slotnum>/]<portnum> [**to** [<stack-unit>/<slotnum>/]<portnum> | **ethernet** [<stack-unit>/<slotnum>/]<portnum>]

or

Syntax: **tagged ethernet** [<stack-unit>/<slotnum>/]<portnum> [**to** [<stack-unit>/<slotnum>/]<portnum> | **ethernet** [<stack-unit>/<slotnum>/]<portnum>]

Syntax: [**no**] **pvlan type community** | **isolated** | **primary**

Syntax: [**no**] **pvlan mapping** <vlan-id> **ethernet** [<stack-unit>/<slotnum>/]<portnum>

Syntax: [**no**] **pvlan pvlan-trunk** <vlan-id> **ethernet** [<stack-unit>/<slotnum>/]<portnum> [**to** [<stack-unit>/<slotnum>/]<portnum>]

The **untagged** or **tagged** command adds the ports to the VLAN.

The **pvlan type** command specifies that this port-based VLAN is a PVLAN. Specify **primary** as the type.

The **pvlan mapping** command identifies the other PVLANS for which this VLAN is the primary. The command also specifies the primary VLAN ports to which you are mapping the other secondary VLANs. The mapping command is not allowed on the secondary VLANs. The parameters of the **pvlan mapping** command are as follows:

- The <vlan-id> parameter specifies another PVLAN. The other PVLAN you want to specify must already be configured.
- The **ethernet** <portnum> parameter specifies the primary VLAN port to which you are mapping all the ports in the other PVLAN (the one specified by <vlan-id>).

The **pvlan pvlan-trunk** command identifies the switch-switch link for the PVLAN. There can be more than one switch-switch link for a single community VLAN.

Configuring an isolated or community PVLAN

You can use the **pvlan type** command to configure the PVLAN as an isolated or community PVLAN. The following are some configuration considerations to be noted for configuring isolated and community PVLANs.

Isolated VLANs

- A port being added to the isolated VLAN can be either a tagged port or an untagged port.
- A member port of an isolated VLAN classifies a frame based on PVID only.
- An isolated port (member of an isolated VLAN) communicates only with the promiscuous port, if a promiscuous port is configured.
- An isolated VLAN must be associated with the primary VLAN for traffic from the isolated port to be switched. An isolated VLAN is associated with only one primary VLAN and to the same primary VLAN in the entire switched network.
- An isolated port communicates only with the configured switch-switch link port if there are no promiscuous ports configured for the isolated VLAN.
- A primary VLAN is associated with only one isolated VLAN. An isolated VLAN can only be mapped to a promiscuous port and a switch-switch link port that belong to the same primary VLAN.
- Link Aggregation Group (LAG) ports are not allowed as member ports of an isolated VLAN.

Community VLANs

- A port being added to the community VLAN can be either a tagged port or an untagged port.
- A member port of a community VLAN classifies a frame based on PVID only.
- A community VLAN is associated with only one primary VLAN and to the same primary VLAN in the entire switched network. A primary VLAN is associated with multiple community VLANs.
- A community VLAN must be associated with the primary VLAN for traffic from the community port to be switched.
- LAG ports are not allowed as member ports of a community VLAN.

To configure a community PVLAN, enter commands such as the following.

```
Brocade(config)# vlan 901
Brocade(config-vlan-901)# untagged ethernet 3/5 to 3/6
Brocade(config-vlan-901)# pvlan type community
```

These commands create port-based VLAN 901, add ports 3/5 and 3/6 to the VLAN as untagged ports, then specify that the VLAN is a community PVLAN.

Syntax: **untagged ethernet** [<slotnum>/]<portnum> [**to** [<slotnum>/]<portnum> | **ethernet** [<slotnum>/]<portnum>]

or

Syntax: **tagged ethernet** [<slotnum>/]<portnum> [**to** [<slotnum>/]<portnum> | **ethernet** [<slotnum>/]<portnum>]

Syntax: [**no**] **pvlan type community | isolated | primary**

The **untagged ethernet** or **tagged ethernet** command adds the ports to the VLAN.

The **pvlan type** command specifies that this port-based VLAN is a PVLAN and can be of the following types:

- **community** – Broadcasts and unknown unicasts received on community ports are sent to the primary port and also are flooded to the other ports in the community VLAN.
- **isolated** – Broadcasts and unknown unicasts received on isolated ports are sent only to the primary port. They are not flooded to other ports in the isolated VLAN.
- **primary** – The primary PVLAN ports are “promiscuous”. They can communicate with all the isolated PVLAN ports and community PVLAN ports in the isolated and community VLANs that are mapped to the promiscuous port.

Changing from one PVLAN type to another (for example, from primary to community or vice versa) is allowed but the mapping will be removed.

Enabling broadcast or unknown unicast traffic to the PVLAN

To enhance PVLAN security, the primary PVLAN does not forward broadcast or unknown unicast packets to its community and isolated VLANs, and other ports in the primary VLAN. For example, if port 3/2 in [Figure 90](#) on page 809 receives a broadcast packet from the firewall, the port does not forward the packet to the other PVLAN ports (3/5, 3/6, 3/9, and 3/10).

This forwarding restriction does not apply to traffic from the PVLAN. The primary port does forward broadcast and unknown unicast packets that are received from the isolated and community VLANs. For example, if the host on port 3/9 sends an unknown unicast packet, port 3/2 forwards the packet to the firewall.

If you want to remove the forwarding restriction, you can enable the primary port to forward broadcast or unknown unicast traffic, if desired, using the following CLI method. You can enable or disable forwarding of broadcast or unknown unicast packets separately.

NOTE

On Layer 2 switches and Layer 3 switches, you also can use MAC address filters to control the traffic forwarded into and out of the PVLAN. In addition, if you are using a Layer 2 switch, you also can use ACLs.

NOTE

FCX devices do not support ACLs on interface groups.

CLI example for a general PVLAN network

To configure the PVLANs shown in [Figure 90](#) on page 809, enter the following commands.

```
Brocade(config)# vlan 901
Brocade(config-vlan-901)# untagged ethernet 3/5 to 3/6
Brocade(config-vlan-901)# pvlan type community
Brocade(config-vlan-901)# exit
Brocade(config)# vlan 902
Brocade(config-vlan-902)# untagged ethernet 3/9 to 3/10
Brocade(config-vlan-902)# pvlan type isolated
Brocade(config-vlan-902)# exit
Brocade(config)# vlan 903
Brocade(config-vlan-903)# untagged ethernet 3/7 to 3/8
```

```

Brocade(config-vlan-903)# pvlan type community
Brocade(config-vlan-903)# exit
Brocade(config)# vlan 7
Brocade(config-vlan-7)# untagged ethernet 3/2
Brocade(config-vlan-7)# pvlan type primary
Brocade(config-vlan-7)# pvlan mapping 901 ethernet 3/2
Brocade(config-vlan-7)# pvlan mapping 902 ethernet 3/2
Brocade(config-vlan-7)# pvlan mapping 903 ethernet 3/2

```

CLI example for a PVLAN network with switch-switch link ports

To configure the PVLANs with tagged switch-switch link ports as shown in [Figure 92](#) on page 812, enter the following commands.

FCX Switch 1

```

Brocade(config)# vlan 101 by port
Brocade(config-vlan-101)# untagged ethernet 1/1/3
Brocade(config-vlan-101)# pvlan type isolated

Brocade(config)# vlan 102 by port
Brocade(config-vlan-102)# untagged ethernet 1/1/1 to 1/1/2
Brocade(config-vlan-102)# pvlan type community

Brocade(config)# vlan 100 by port
Brocade(config-vlan-100)# tagged ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# untagged ethernet 1/1/4
Brocade(config-vlan-100)# pvlan type primary
Brocade(config-vlan-100)# pvlan mapping 101 ethernet 1/1/4
Brocade(config-vlan-100)# pvlan mapping 102 ethernet 1/1/4
Brocade(config-vlan-100)# pvlan pvlan-trunk 101 ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# pvlan pvlan-trunk 102 ethernet 1/1/10 to 1/1/11

```

FCX Switch 2

```

Brocade(config)# vlan 101 by port
Brocade(config-vlan-101)# untagged ethernet 1/1/3
Brocade(config-vlan-101)# pvlan type isolated

Brocade(config)# vlan 102 by port
Brocade(config-vlan-102)# untagged ethernet 1/1/1 to 1/1/2
Brocade(config-vlan-102)# pvlan type community

Brocade(config)# vlan 100 by port
Brocade(config-vlan-100)# tagged ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# pvlan type primary
Brocade(config-vlan-100)# pvlan pvlan-trunk 102 ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# pvlan pvlan-trunk 101 ethernet 1/1/10 to 1/1/11

```

FCX Switch 3

```

Brocade(config)# vlan 101 by port
Brocade(config-vlan-101)# untagged ethernet 1/1/3
Brocade(config-vlan-101)# pvlan type isolated

Brocade(config)# vlan 102 by port

```

Dual-mode VLAN ports

```
Brocade(config-vlan-102)# untagged ethernet 1/1/1 to 1/1/2
Brocade(config-vlan-102)# pvlan type community

Brocade(config)# vlan 100 by port
Brocade(config-vlan-100)# tagged ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# pvlan type primary
Brocade(config-vlan-100)# pvlan pvlan-trunk 102 ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# pvlan pvlan-trunk 101 ethernet 1/1/10 to 1/1/11
```

FCX Switch 4

```
Brocade(config)# vlan 101 by port
Brocade(config-vlan-101)# untagged ethernet 1/1/3
Brocade(config-vlan-101)# pvlan type isolated

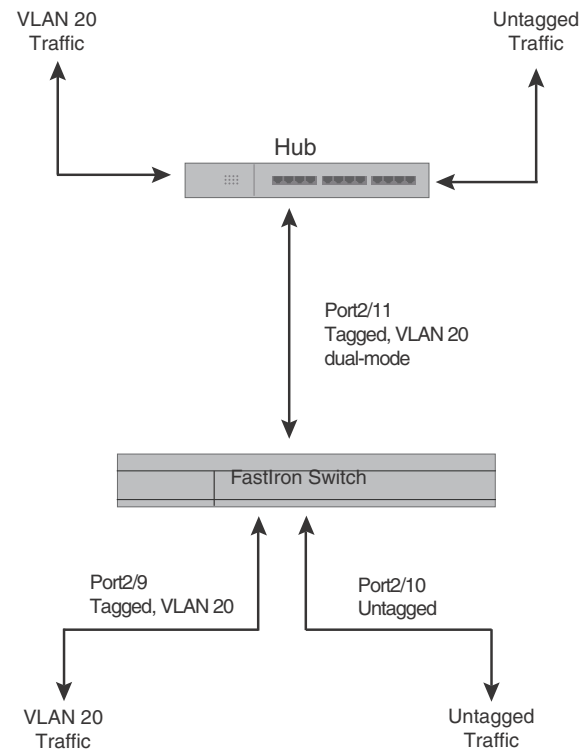
Brocade(config)# vlan 102 by port
Brocade(config-vlan-102)# untagged ethernet 1/1/1 to 1/1/2
Brocade(config-vlan-102)# pvlan type community
Brocade(config)# vlan 100 by port
Brocade(config-vlan-100)# tagged ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# pvlan type primary
Brocade(config-vlan-100)# pvlan pvlan-trunk 102 ethernet 1/1/10 to 1/1/11
Brocade(config-vlan-100)# pvlan pvlan-trunk 101 ethernet 1/1/10 to 1/1/11
```

Dual-mode VLAN ports

Configuring a tagged port as a **dual-mode** port allows it to accept and transmit both tagged traffic and untagged traffic at the same time. A dual-mode port accepts and transmits frames belonging to VLANs configured for the port, as well as frames belonging to the default VLAN (that is, untagged traffic).

For example, in [Figure 93](#), port 2/11 is a dual-mode port belonging to VLAN 20. Traffic for VLAN 20, as well as traffic for the default VLAN, flows from a hub to this port. The dual-mode feature allows traffic for VLAN 20 and untagged traffic to go through the port at the same time.

FIGURE 93 Dual-mode VLAN port example

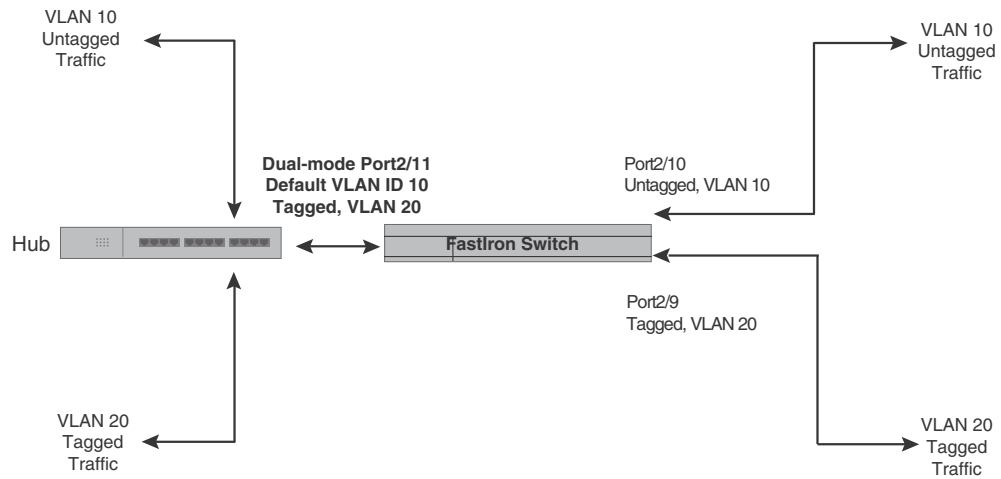


To enable the dual-mode feature on port 2/11 in [Figure 93](#), enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# tagged ethernet 2/11
Brocade(config-vlan-20)# tagged ethernet 2/9
Brocade(config-vlan-20)# interface ethernet 2/11
Brocade(config-if-e1000-2/11)# dual-mode
Brocade(config-if-e1000-2/11)# exit
```

Syntax: [no] dual-mode

You can configure a dual-mode port to transmit traffic for a specified VLAN (other than the DEFAULT-VLAN) as untagged, while transmitting traffic for other VLANs as tagged. [Figure 94](#) illustrates this enhancement.

FIGURE 94 Specifying a default VLAN ID for a dual-mode port

In [Figure 94](#), tagged port 2/11 is a dual-mode port belonging to VLANs 10 and 20. The default VLAN assigned to this dual-mode port is 10. This means that the port transmits tagged traffic on VLAN 20 (and all other VLANs to which the port belongs) and transmits untagged traffic on VLAN 10.

The dual-mode feature allows tagged traffic for VLAN 20 and untagged traffic for VLAN 10 to go through port 2/11 at the same time. A dual-mode port transmits only untagged traffic on its default VLAN (that is, either VLAN 1, or a user-specified VLAN ID), and only tagged traffic on all other VLANs.

The following commands configure VLANs 10 and 20 in [Figure 94](#). Tagged port 2/11 is added to VLANs 10 and 20, then designated a dual-mode port whose specified default VLAN is 10. In this configuration, port 2/11 transmits only untagged traffic on VLAN 10 and only tagged traffic on VLAN 20.

```
Brocade(config)# vlan 10 by port
Brocade(config-vlan-10)# untagged ethernet 2/10
Brocade(config-vlan-10)# tagged ethernet 2/11
Brocade(config-vlan-10)# exit
Brocade(config)# vlan 20 by port
Brocade(config-vlan-20)# tagged ethernet 2/9
Brocade(config-vlan-20)# tagged ethernet 2/11
Brocade(config-vlan-20)# exit
Brocade(config)# interface ethernet 2/11
Brocade(config-if-e1000-2/11)# dual-mode 10
Brocade(config-if-e1000-2/11)# exit
```

Syntax: [no] dual-mode [<vlan-id>]

Notes:

- If you do not specify a <vlan-id> in the **dual mode** command, the port default VLAN is set to 1. The port transmits untagged traffic on the DEFAULT-VLAN.
- The dual-mode feature is disabled by default. Only tagged ports can be configured as dual-mode ports.
- In trunk group, either all of the ports must be dual-mode, or none of them can be.

The **show vlan** command displays a separate row for dual-mode ports on each VLAN.

Example

```
Brocade# show vlan
Total PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 16

legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree Off
  Untagged Ports: (S1) 1 2 3 4 5 6 7 8
  Untagged Ports: (S2) 1 2 3 4 5 6 7 8 12 13 14 15 16 17 18 19
  Untagged Ports: (S2) 20 21 22 23 24
  Tagged Ports: None
  Uplink Ports: None
DualMode Ports: None
PORT-VLAN 10, Name [None], Priority level0, Spanning tree Off
  Untagged Ports: (S2) 10
  Tagged Ports: None
  Uplink Ports: None
DualMode Ports: (S2) 11
PORT-VLAN 20, Name [None], Priority level0, Spanning tree Off
  Untagged Ports: None
  Tagged Ports: (S2) 9
  Uplink Ports: None
DualMode Ports: (S2) 11
```

Displaying VLAN information

After you configure the VLANs, you can verify the configuration using the **show** commands described in this section.

NOTE

If a VLAN name begins with "GVRP_VLAN_", the VLAN was created by the GARP VLAN Registration Protocol (GVRP). If a VLAN name begins with "STATIC_VLAN_", the VLAN was created by GVRP and then was converted into a statically configured VLAN.

Displaying VLANs in alphanumeric order

By default, VLANs are displayed in alphanumeric order, as shown in the following example.

```
Brocade# show run
...
vlan 2 by port
...
vlan 10 by port
...
vlan 100 by port
...
```

Displaying system-wide VLAN information

Use the **show vlans** command to display VLAN information for all the VLANs configured on the device.

The following example shows the display for the IP subnet and IPX network VLANs configured in the examples in [“Configuring an IP subnet VLAN with dynamic ports”](#) on page 785 and [“Configuring an IPX network VLAN with dynamic ports”](#) on page 785.

```
Brocade# show vlans
Total PORT-VLAN entries: 2
Maximum PORT-VLAN entries: 8
legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree Off
  Untagged Ports: (S2) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
  Untagged Ports: (S2) 17 18 19 20 21 22 23 24
  Untagged Ports: (S4) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
  Untagged Ports: (S4) 17 18 19 20 21 22 23 24
  Tagged Ports: None

PORT-VLAN 10, Name IP_VLAN, Priority level0, Spanning tree Off
  Untagged Ports: (S1) 1 2 3 4 5 6
  Tagged Ports: None

IP-subnet VLAN 1.1.1.0 255.255.255.0, Dynamic port enabled
  Name: Mktg-LAN
  Static ports: None
  Exclude ports: None
  Dynamic ports: (S1) 1 2 3 4 5 6
PORT-VLAN 20, Name IPX_VLAN, Priority level0, Spanning tree Off
  Untagged Ports: (S2) 1 2 3 4 5 6
  Tagged Ports: None

IPX-network VLAN 0000ABCD, frame type ethernet_ii, Dynamic port enabled
  Name: Eng-LAN
  Static ports: None
  Exclude ports: None
  Dynamic ports: (S2) 1 2 3 4 5 6
```

In the **show vlans** output, ports that are tagged but are not dual-mode ports are listed as tagged ports. In the following example display output, ports 7 and 8 are dual-mode ports in port-based VLAN 4. Ports 7 and 8 also belong to port-based VLAN 3, but they are tagged ports only in VLAN 3 and are not configured as dual-mode ports.

```

Brocade# show vlan 4
Total PORT-VLAN entries: 5
Maximum PORT-VLAN entries: 3210
PORT-VLAN 4, Name [None], Priority level0, Spanning tree Off
  Untagged Ports: None
  Tagged Ports:   6   9  10  11
  Uplink Ports: None
  DualMode Ports:  7   8
ESX624FE+2XG Router# show vlan 3
Total PORT-VLAN entries: 5
Maximum PORT-VLAN entries: 3210
PORT-VLAN 3, Name [None], Priority level0, Spanning tree Off
  Untagged Ports: None
  Tagged Ports:   6   7   8   9  10
  Uplink Ports: None
  DualMode Ports: None

```

Syntax: `show vlans [<vlan-id> | ethernet [<slotnum>/]<portnum>]`

The `<vlan-id>` parameter specifies a VLAN for which you want to display the configuration information.

The `<slotnum>` parameter is required on chassis devices.

The `<portnum>` parameter specifies a port. If you use this parameter, the command lists all the VLAN memberships for the port.

Displaying global VLAN information

The **show vlan brief** command displays the following information:

- The system-max VLAN values (maximum, default, and current)
- The default VLAN ID number
- The total number of VLANs configured on the device
- The VLAN ID numbers of the VLANs configured on the device

The following shows example output.

```

Brocade# show vlan brief
System-max vlan Params: Max(4095) Default(64) Current(3210)
Default vlan Id :1
Total Number of Vlan Configured :5
VLANs Configured :1 to 4 10

```

Syntax: `show vlan brief`

Displaying VLAN information for specific ports

Use one of the following methods to display VLAN information for specific ports.

To display VLAN information for all the VLANs of which port 7/1 is a member, enter the following command.

Displaying VLAN information

```
Brocade# show vlans ethernet 7/1
Total PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 8

legend: [S=Slot]

PORT-VLAN 100, Name [None], Priority level0, Spanning tree Off
Untagged Ports: (S7) 1 2 3 4
Tagged Ports: None
```

Syntax: `show vlans [<vlan-id> | ethernet [<slotnum>/]<portnum>]`

The `<vlan-id>` parameter specifies a VLAN for which you want to display the configuration information.

The `<slotnum>` parameter is required on chassis devices.

The `<portnum>` parameter specifies a port. If you use this parameter, the command lists all the VLAN memberships for the port.

Displaying a port VLAN membership

To display VLAN membership for a specific port on the device, enter a command such as the following.

```
Brocade# show vlan brief ethernet 7
Port 7 is a member of 3 VLANs
VLANs 3 to 4 10
```

Syntax: `show vlan brief ethernet [<slotnum>/]<portnum>`

The `<slotnum>` parameter is required on chassis devices.

Displaying a port dual-mode VLAN membership

The output of the **show interfaces** command lists dual-mode configuration and corresponding VLAN numbers. The following shows an example output.

```
Brocade# show interfaces ethernet 7
GigabitEthernet7 is down, line protocol is down
  Hardware is GigabitEthernet, address is 0012.f2a8.4706 (bia 0012.f2a8.4706)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown
  Configured mdi mode AUTO, actual unknown
  Member of 3 L2 VLANs, port is dual mode in Vlan 4, port state is BLOCKING
```

Syntax: `show interfaces ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum>] [ethernet [<slotnum>/]<portnum>...]`

The `<slotnum>` parameter is required on chassis devices.

Displaying port default VLAN IDs (PVIDs)

The output of the **show interfaces brief** command lists the port default VLAN IDs (PVIDs) for each port. PVIDs are displayed as follows:

- For untagged ports, the PVID is the VLAN ID number.
- For dual-mode ports, the PVID is the dual-mode VLAN ID number.
- For tagged ports without dual-mode, the PVID is always Not Applicable (NA).

```
Brocade# show interfaces brief
Port  Link      State   Dupl Speed Trunk Tag Pvid Pri MAC      Name
1      Up         Forward Full 1G    None No  1   0   0012.f2a8.4700 a12345678901
2      Up         Forward Full 1G    None Yes 1   0   0012.f2a8.4701
3      Up         Forward Full 1G    None Yes NA  0   0012.f2a8.4702
4      Up         Forward Full 1G    None Yes NA  0   0012.f2a8.4703
5      Up         Forward Full 1G    None No  2   0   0012.f2a8.4704
6      Down       None    None None None Yes NA  0   0012.f2a8.4705
7      Down       None    None None None Yes 4   0   0012.f2a8.4706
8      Down       None    None None None Yes 4   0   0012.f2a8.4707
9      Down       None    None None None Yes NA  0   0012.f2a8.4708
10     Down       None    None None None Yes NA  0   0012.f2a8.4709
```

Syntax: **show interfaces brief** [**ethernet** [<slotnum>/]<portnum> [**to** [<slotnum>/]<portnum> [**ethernet** [<slotnum>/]<portnum>...]]]

The <slotnum> parameter is required on chassis devices.

Displaying PVLAN information

To display the PVLAN configuration with respect to the primary VLAN and its associated secondary VLANs and to display the member ports, promiscuous ports, and the switch-switch link ports of a PVLAN, enter a command such as the following.

```
Brocade# show pvlan
PVLAN: primary VLAN 100
Port 1/1/4 1/1/10 1/1/11
Community VLAN 102
Port 1/1/1 1/1/2 1/1/10 1/1/11
Promiscuous Port: 1/1/4
Inter switch link Port: 1/1/10 1/1/11
BpduGuard enabled Port: 1/1/1 1/1/2
Isolate VLAN 101
Port 1/1/3 1/1/10 1/1/11
Promiscuous Port: 1/1/4
Inter switch link Port: 1/1/10 1/1/11
BpduGuard enabled Port: 1/1/1 1/1/2
```

Syntax: **show pvlan** <vid>

The <vid> variable specifies the VLAN ID of the PVLAN. If the VLAN ID is not specified, the command displays the default VLAN ID.

Displaying VLAN information

Multi-Chassis Trunking

Table 143 lists the individual Brocade FastIron switches and the Multi-Chassis Trunking (MCT) features they support.

TABLE 143 Supported MCT features

Feature	FSX 800 FSX 1600 ^a	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
MCT ^b	Yes	No	No	No	No
Cluster client automatic configuration	Yes	No	No	No	No
Cluster operation features	Yes	No	No	No	No
xSTP BPDU forwarding	Yes	No	No	No	No

- a. MCT ICL and CCEP connectivity is supported only on the following FSX 800 and FSX 1600 modules.
 - SX-FI-24GPP, 24-port Gigabit Ethernet copper interface module with PoE+
 - SX-FI-24HF, 24-port Gigabit Ethernet fiber interface module
 - SX-FI-2XG, 2-port 10 Gigabit Ethernet interface module
 - SX-FI-8XG, 8-port 10 Gigabit Ethernet interface module
 - SX-FI-48GPP, 48-port Gigabit Ethernet copper interface module with POE+ (2 slot high, 2:1 oversubscribed)
- b. MCT can operate on an FSX chassis that contains the modules listed in the previous footnote plus legacy modules, provided that CCEP and ICL are configured on the above-mentioned modules. MCT peering is supported only with FSX switches.

Multi-Chassis Trunking Overview

Multi-Chassis Trunking (MCT) is an alternative to spanning tree protocols. Spanning tree is a technology that protects the network against loops by blocking necessary ports, and having the network span to relearn topologies when one link fails in a network. MCT is a technology that allows two FSX switches to cluster together and appear as a single logical device. Trunking is a technology that allows multiple links of a device to appear as one logical link. The combination of MCT and trunking allows for creating a resilient network topology that utilizes all links in the network, creating an ideal network topology for latency sensitive applications.

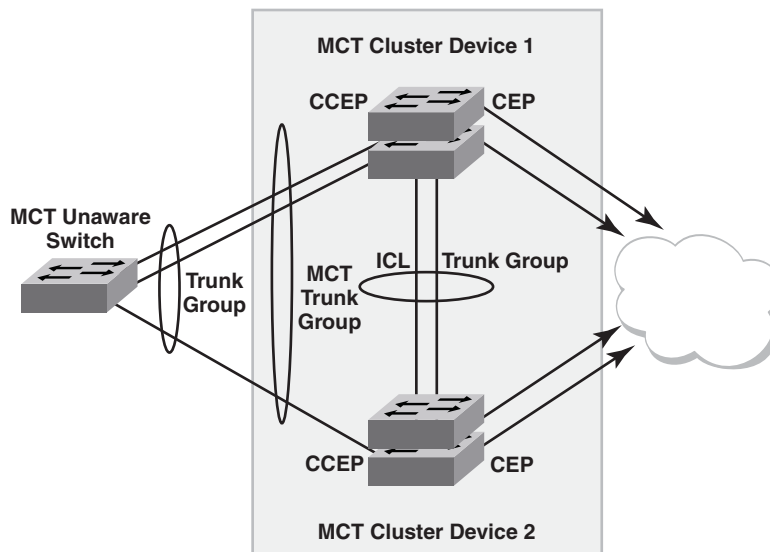
Standard static or dynamic LACP trunks provide link-level redundancy and increased capacity. However, trunks do not provide device-level redundancy. If the device to which the trunk is attached fails, the entire trunk loses network connectivity. Two devices are needed for network resiliency with trunked links to both devices. With spanning tree, one of these trunks would be blocked from use until the failure of the other trunk is detected, taking from 1 to 30 seconds potentially adding latency and jitter, not only on the affected devices locally, but through-out the span topology. With MCT, member links of the trunk are split and connected to two clustered FSX chassis. MCT has integrated loop detections allowing all links to be active. If a failure is detected, traffic is dynamically allocated across the remaining links with the failure detection and allocation of traffic occurring in sub-second time without impacting the rest of the network.

MCT inherits all of the benefits of a trunk group by providing multiple physical links to act as a single logical link. The new available bandwidth is an aggregate of all the links in the group. The traffic is shared across the links in the group using dynamic flow-based load balancing and traffic is moved to a remaining link group in sub-seconds in the event of a failure in one of the links. MCT eliminates the single point of failure that exists at a device level when all links of a trunk terminate on the same device without the overhead associated with spanning tree. MCT diverts a subset of the links to a second device to provide redundancy and sub-second fault detection at the device level.

How MCT works

Figure 95 shows a basic MCT configuration. The MCT initiates at a single MCT-unaware server or switch and terminates at two MCT-aware devices.

FIGURE 95 How MCT works



The MCT process involves the following processes:

- Sub-second failover occurs in the event of a link, module, switch fabric, control plane, or device failure.
- Sub-second failover operates at the physical level.
- Layer 2 and Layer 3 forwarding (when using fast path forwarding) is done at the first hop regardless of VRRP-E state.
- Load balancing is flow-based (does not involve VLANs sharing across network links).
- Resiliency is supported regardless of the traffic type (Layer 3, Layer 2 or non-IP legacy protocols).
- Interaction with Metro Ring Protocol (MRP) builds larger resilient Layer 2 domains.
- Device level redundancy is provided in addition to link and modular redundancy.
- Traffic received from an ICL port is not forwarded to the Cluster Client Edge Ports (CCEPs) if the MCT peer device has the reach ability to the same cluster client.

- Traffic received from non-ICL ports is forwarded the same way as non-MCT devices.
- Known unicast, multicast, and broadcast traffic received on Cluster Edge Ports (CEP) or ICL ports is forwarded to the destination port.
- For unknown unicast, multicast, and broadcast traffic received on ICL ports, the forwarding behavior depends on the peer MCT device's ability to reach the same client.
- Unknown unicast, multicast, and broadcast traffic received from CCEP is forwarded as usual, with the default behavior to flood the entire VLAN.

MCT terminology

- MCT cluster: A pair of devices (switches) that is clustered together using MCT to appear as a single logical device. The devices are connected as peers through an Inter-Chassis Link (ICL).
- MCT cluster device: One of the two devices in an MCT cluster.
- MCT peer device: From the perspective of an MCT cluster device, the other device in the MCT cluster.
- MCT cluster client: A device that connects with MCT cluster devices through static or dynamic trunks. It can be a switch or an endpoint server host in the single-level MCT topology or another pair of MCT devices in a multi-tier MCT topology.
- Inter-Chassis Link (ICL): A single-port or multi-port 1 GbE or 10 GbE interface between the two MCT cluster devices. It provides the control path for CCP for the cluster and also serves as the data path between the two devices.
- MCT VLANs: VLANs on which MCT cluster clients are operating. Any VLAN that has an ICL port is an MCT VLAN, even though it does not have any clients.
- MCT session VLANs: The VLAN used by the MCT cluster for control operations. CCP protocol runs over this VLAN. The interface can be a single link or a trunk group port. If it is a trunk group port, it should be the primary port of the trunk group. The MCT session VLAN subnet is not distributed in routing protocols using redistribute commands.
- MCT keep-alive VLAN: The VLAN that provides a backup control path in the event that ICL goes down.
- Cluster Communication Protocol (CCP): A Brocade proprietary protocol that provides reliable, point-to-point transport to synchronize information between MCT cluster devices. It is the default MCT control path between the two peer devices. CCP comprises two main components: CCP peer management and CCP client management. CCP peer management deals with establishing, and maintaining a TCP transport session between peers, while CCP client management provides event-based, reliable packet transport to CCP peers.
- Cluster Client Edge Port (CCEP): A physical port or trunk group interface on an MCT cluster device that is connected to client devices.
- Cluster Edge Port (CEP): A port on an MCT cluster device that belongs to the MCT VLAN and connects to an upstream core switch/router, but is neither a CCEP nor an ICL.
- RBridgeID: RBridgeID is a value assigned to MCT cluster devices and clients to uniquely identify them, and helps in associating the source MAC address with an MCT device.

MCT data flow

MCT can be deployed in a single-level configuration involving two MCT cluster devices or in a cascading configuration with a pair of MCT cluster devices operating as switches and another pair operating as routers. Refer to “[Single-level MCT example](#)” on page 867 for a single-level illustration and configuration example, and “[Two-level MCT example](#)” on page 871 for a two-level or cascading configuration example.

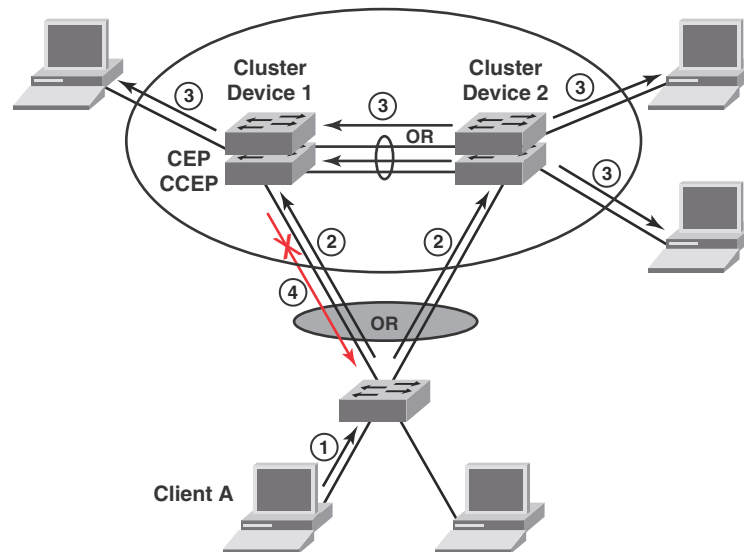
Basic MCT data flow works as follows.

Broadcast, unknown unicast, and multicast (BUM) traffic from a client through a CCEP

1. Traffic originates at the client.
2. Because the link between the client switch and the MCT cluster is a trunk, the traffic travels over one physical link. In the example in [Figure 96](#), the traffic travels over the link towards cluster device 2. The traffic enters the MCT cluster through the CCEP of cluster device 2.
3. The traffic is sent to any local CEPs and CCEPs. It passes to the peer cluster device over the ICL link, where it is sent to the peer device's local CEPs.
4. Traffic does not pass back down to the client through the CCEP.

Refer to [Figure 96](#).

FIGURE 96 MCT data flow - BUM traffic from CCEP

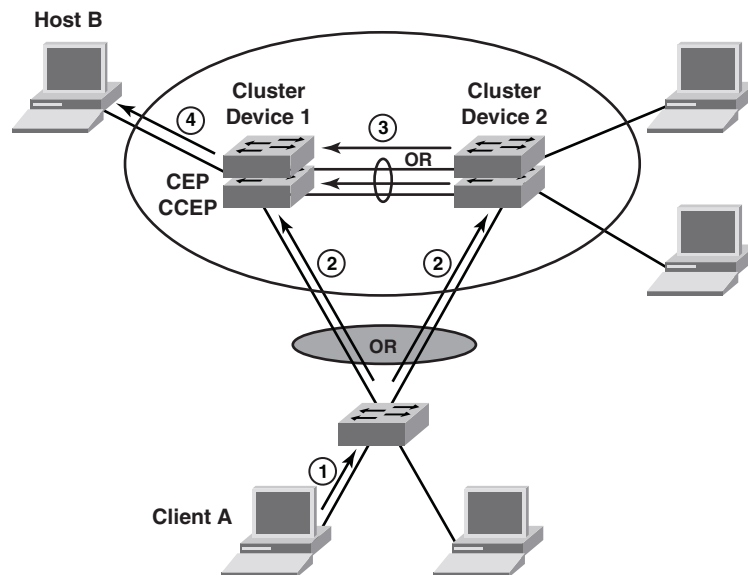


Unicast traffic from a client through a CCEP to a CEP

1. Traffic originates at the client.
2. Because the link between the client switch and the MCT cluster is a trunk, the traffic travels over one physical link. In the example in [Figure 97](#), the traffic travels over the link towards cluster device 2. The traffic enters the MCT cluster through the CCEP of cluster device 2.
3. Depending on the destination, the traffic may pass over the ICL link to the other cluster device. In the example in [Figure 97](#), the destination is on cluster device 1, so the traffic is forwarded out to the ICL port.
4. The traffic passes out to the destination.

Refer to [Figure 97](#).

FIGURE 97 MCT data flow - unicast traffic from CCEP

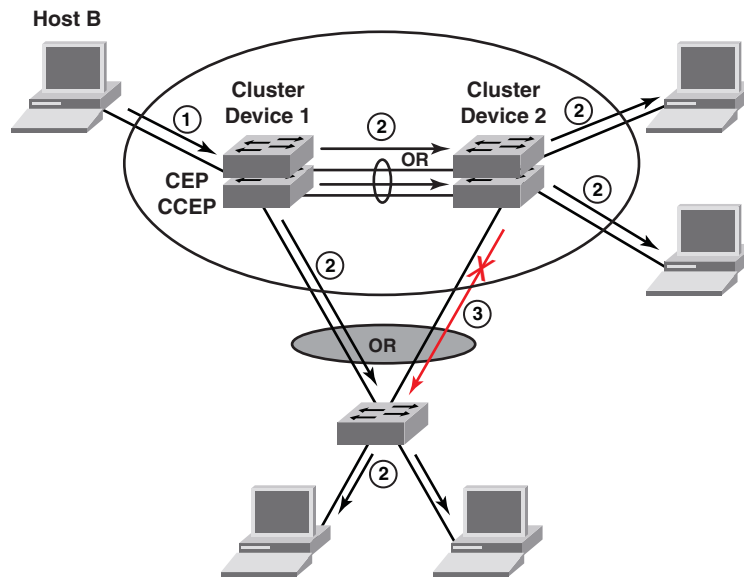


Broadcast, unknown unicast, and multicast (BUM) traffic from a client through a CEP

1. Traffic originates at the client and enters one of the MCT cluster devices through a CEP.
2. The traffic is sent to the peer cluster device through the ICL link and also sent to any local CCEPs and CEPs. Once traffic is received on the peer cluster device, it will be sent to its local CEPs.
3. Traffic does not pass back down to the client through the CCEP.

Refer to [Figure 98](#).

FIGURE 98 MCT data flow - BUM traffic from a CEP

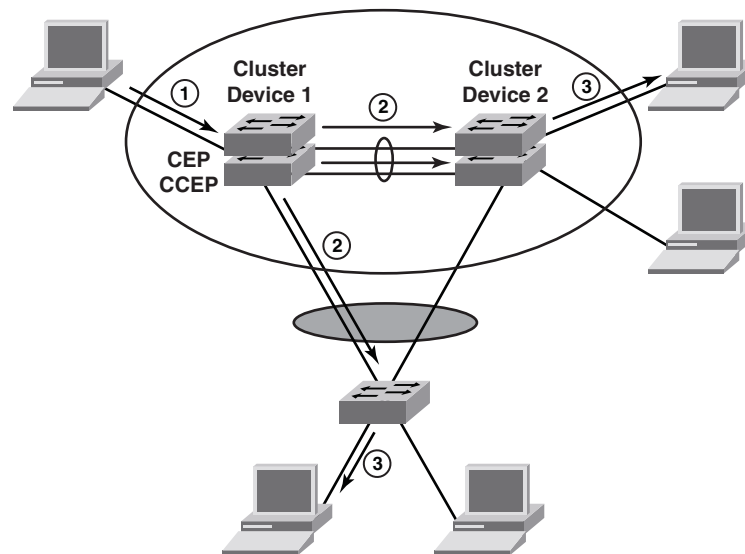


Unicast traffic from a client through a CEP to another CEP or a CCEP

1. Traffic originates at the client and enters one of the cluster devices through the CEP.
2. Depending on the destination, the traffic may pass over the ICL link to the other cluster device or sent to a local CCEP.
3. The traffic passes out to the destination.

Refer to [Figure 97](#).

FIGURE 99 MCT data flow - unicast traffic from a CEP

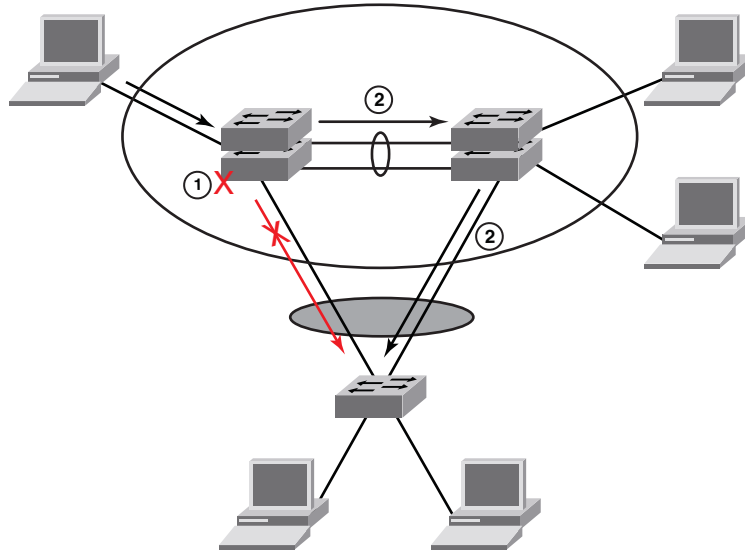


Port failure on the cluster device

1. A CCEP on the cluster device that received the unicast or BUM traffic fails.
2. The traffic is automatically redirected to the other MCT cluster device over the ICL and on to its destinations through CCEPs.

Refer to [Figure 100](#).

FIGURE 100 MCT data flow with port failure

**MCT and VLANs**

MCT relies on the following VLAN types:

- Session VLAN: Provides the control channel for CCP. Brocade recommends keeping only ICL ports in the session VLAN. A virtual interface is required to be configured on the session VLAN for the router image.
- Keep-alive VLAN: Provides a backup control path if the ICL goes down (optional, but strongly recommended).
- MCT VLAN: Serves the customer data traffic. An ICL must belong to every MCT VLAN to provide a data path between two cluster devices. When an ICL is added to a VLAN, it becomes an MCT VLAN.

Cluster client automatic configuration

Client configuration includes setting the client name, client RBridgeID (unique identification for each client), client interface (CCEP), and deployment settings on both MCT cluster devices. With up to 150 clients per cluster, manual configuration can take a considerable amount of time.

Cluster client automatic configuration saves the time that would be required to complete the entire configuration manually.

The following limitations apply to cluster client automatic configuration:

- Cluster client automatic configuration is designed for generating new clients, not for updating an existing client.
- A single client span across multiple devices is not supported (cascading MCT). For example, the configuration of cascading MCT through cluster client automatic configuration is not supported.
- Multiple clients on the same device are not supported.
- LACP client interface auto-detection is supported only for devices running release 7.4 software and later on FastIron platforms.
- RBridgeID collision: When hash collisions occur, cluster client automatic configuration reports errors, and manual intervention is required.

For cluster client automatic configuration to work, the following prerequisites are required on the cluster side:

- The cluster must be configured on both MCT cluster devices.
- An MCT VLAN must be configured on both MCT cluster devices.
- The trunk group configuration must be removed from the client interfaces.
- The client interfaces must be up and operational.

The following prerequisites are required on the client side:

- VLAN and trunk group configuration must be completed.
- Link Level Discovery Protocol (LLDP) must be enabled.

Refer to [“Setting up cluster client automatic configuration”](#) on page 841 for detailed instructions on the cluster client automatic configuration process.

MCT feature interaction

The following FastIron features are supported with MCT. Note that all security features are locally significant and are not synchronized across an MCT cluster.

- LACP on the CCEP.
- VRRP on the CCEP.
- MRP and MRP II, with the restriction that the ICL port cannot be the secondary port of the MRP ring.
- Flooding features (VLAN CPU protection, multicast flooding, and so on) on MCT VLANs.
- Unidirectional Link Detection (UDLD) as independent boxes (configured independently).
- ARP as independent boxes (configured independently).
- STP and RSTP.

- Ingress ACLs on all MCT ports. Egress ACLs are supported only on MCT CEPs or ICL ports. Egress ACLs are not supported on MCT CCEPs.
- QoS and MAC filters and profiles with the same configuration on both cluster devices.
- IPv4 ACLs and rate limits. If the rules are applied on the CCEPs, the same rules must be applied to the CCEP ports on both cluster devices.
- Layer 3 Routing. VE with IP address assignment is supported on CCEPs for VRRP. However, routing protocols are not enabled on CCEPs.
- Static multi-port MAC.
- Port MAC security, multi-port authentication, and 802.1X, only on CEPs.
- Static MAC address configuration. Static MAC addresses are programmed on both local and remote peers as static entries.
- DAI and DHCP snooping for clients connected through CEPs. They must be configured independently on both cluster devices.
 - If the trusted ports are off the CCEP, the **arp inspection trust** or **dhcp snoop trust** command must be used on the CCEPs and ICL ports.
 - DHCP and ARP entries are created on both MCT cluster devices if the flow traverses both the CCEP and ICL.
- Hitless failover. If the failover operation is performed with a cluster configuration, the TCP session is reestablished. The MAC addresses from the cluster peer devices will be revalidated and programmed accordingly.
- Hitless upgrade. If the upgrade operation is performed with a cluster configuration, the TCP session is reestablished. The MAC addresses from the cluster peer devices will be revalidated and programmed accordingly.

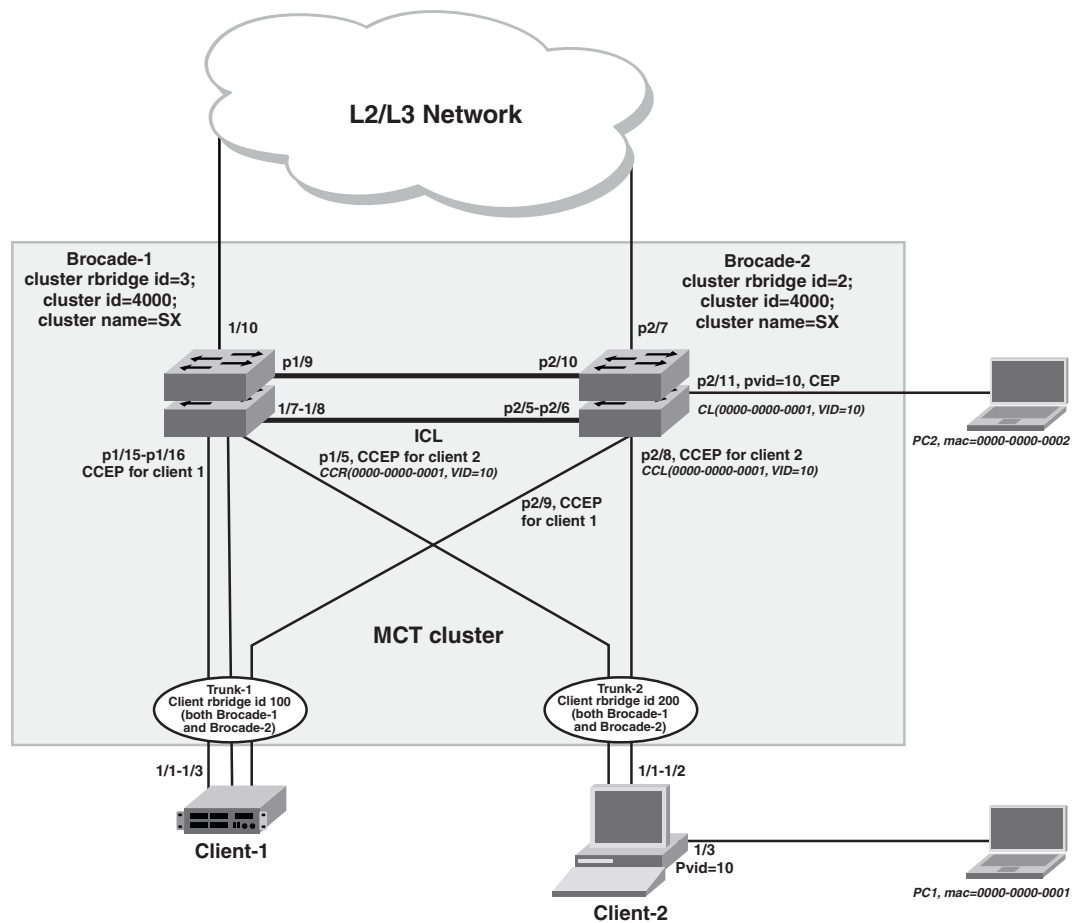
The following FastIron features are not supported with MCT:

- LACP on ICL.
- MSTP, VSRP, RIP, OSPF, IS-IS, and BGP.
- IPv6, VRRP-E (IPv6), and VRRPv3.
- GRE on the ICL VE interfaces.
- DAI on the CCEPs.
- Host security features (port MAC security, multi-port authentication, 802.1X, DAI, DHCP snooping) on CCEPs.
- Multi-port ARP on ICL or CCEPs.
- Web authentication on MCT VLANs.

Basic MCT configuration

This section describes how to set up a basic MCT configuration. [Figure 101](#) shows a basic MCT topology, which applies to Layer 2 and Layer 3. MCT can also be supported with VRRP or VRRP-E. Refer to [“MCT for VRRP or VRRP-E”](#) on page 857.

FIGURE 101 Basic MCT configuration



MCT configuration considerations

- When running STP, the STP state should be the same on both cluster devices. For additional information on running STP with MCT, refer to [“STP/RSTP”](#) on page 850.
- One ICL can be configured per device, and a device can be in only one cluster.
- The software version in both cluster devices must be exactly the same for the cluster to function.
- An ICL port should not be an untagged member of any VLAN. An ICL is preferably a static trunk that provides port level redundancy and higher bandwidth for cluster communication.
- ICL ports must be part of MCT VLANs and session VLANs.

- An ICL cannot be an LACP trunk (must be either a static trunk or single port).
- MAC learning is disabled on ICL ports for all VLANs.
- MDUP synchronizes all MAC entries for VLANs served by an ICL link.
- The cluster ID should be same on both cluster devices.
- The cluster RBridgeID should not conflict with any client RBridgeID or the peer RBridgeID.
- The client RBridgeID is unique and should be the same on the cluster devices.
- Brocade recommends keeping only ICL ports in the session VLAN during operation.
- MCT can support up to 12 members per trunk group.
- An ICL interface cannot be configured as the CCEP in any client.
- BPDU guard and root guard configuration should be identical on both cluster devices.
- As Egress PCL is configured on CCEPs, Egress ACL cannot be configured on them. All types of ingress ACLs, DDOS attacks, and so on can still be configured on those ports.
- Brocade recommends that you configure a keep-alive VLAN as a separate link (not ICL). The keep-alive VLAN provides a backup control path when CCP goes down.

Differences in configuring MCT for the switch and router image

There are some differences in the MCT configuration for the switch image versus the router image:

- On a switch image, STP is by default enabled for all the VLANs; however, for MCT, Layer 2 protocols such as STP and RSTP should not be enabled on the session VLAN. Therefore, STP must be disabled explicitly for the session VLAN. STP is automatically disabled in the router image.
- Virtual Ethernet (VE) cannot be configured on a session VLAN in a switch image, but an IP address is needed for the cluster devices to communicate via CCP. Therefore, in a switch image, the configured management IP address is used to establish communication between the cluster devices.
- The management IP addresses should be configured in each of the cluster devices in the same subnet. If the IP addresses are in different subnets, ARP does not resolve the addresses and MCT may not work. The ARP for the peer cluster devices is always learned on the ICL port or trunk, so any management traffic between the two devices will always go through the ICL ports.

Configuring MCT

This section provides basic configuration steps, which should be completed in the specified order.

Step 1: Configure trunks (if needed)

Step 2: Configure the session VLAN and recommended keep-alive VLAN

Step 3: Configure the cluster

Step 4: Configure clients

After completing these steps, you can verify the configuration by running the **show cluster** command. Refer to [“Displaying peer and client states”](#) on page 860.

Step 1: Configure trunks (if needed)

An ICL is typically a trunk group that provides port level redundancy and higher bandwidth for cluster communication. The ICL can be a single interface or a static trunk. LACP on ICL is not supported.

If needed, configure the ICL trunk as follows on each cluster device (configuration shown for Brocade-1 in [Figure 101](#)).

```
Brocade-1(config)#trunk ethernet 1/15 to 1/16
Brocade-1(config)#trunk deploy
```

On the client side, trunk configuration is required for a static trunk only before assigning interfaces as CCEP. It is not necessary to configure trunks for a single client interface or LACP client interface. If needed, configure client side trunks on each cluster device (configuration shown for Client-1 in [Figure 101](#)).

```
Client-1(config)#trunk ethernet 1/1 to 1/3
Client-1(config)#trunk deploy
```

Step 2: Configure the session VLAN and recommended keep-alive VLAN

To create the session VLAN and recommended keep-alive VLAN for Brocade-1 in the [Figure 101](#) topology, enter the following commands.

```
Brocade-1(config)#vlan 3001 name MCT-keep-alive
Brocade-1(config-vlan-3001)#tagged ethernet 1/9
Brocade-1(config-vlan-3001)#exit
Brocade-1(config)#vlan 3000 name Session-VLAN
Brocade-1(config-vlan-3000)#tagged ether 1/7 to 1/8
Brocade-1(config-vlan-3000)#no spanning-tree
```

For routers, add the following commands.

```
Brocade-1(config-vlan-3000)#router-interface ve 3000
Brocade-1(config)# interface ve 3000
Brocade-1(config-vif-3000)#ip address 1.1.1.3/24
```

For switches, add the following commands.

```
Brocade-1(config)#ip address 1.1.1.3/24
```

To create a session VLAN and keep-alive VLAN for Brocade-2, enter the following commands.

```
Brocade-2(config)#vlan 3001 name MCT-keep-alive
Brocade-2(config-vlan-3001)#tagged ethernet 2/10
Brocade-2(config-vlan-3001)#exit
Brocade-2(config)# vlan 3000 name Session-VLAN
Brocade-2(config-vlan-3000)#tagged ether 2/5 to 2/6
Brocade-2(config-vlan-3000)#no spanning-tree
```

For routers, add the following commands.

```
Brocade-2(config-vlan-3000)#router-interface ve 3000
Brocade-2(config)#interface ve 3000
Brocade-2(config-vif-3000)#ip address 1.1.1.2/24
```

For switches, add the following commands.

```
Brocade-2(config)#ip address 1.1.1.2/24
```

To implicitly configure the session VLAN and add the ICL as a tagged member of the VLAN, enter the following commands.

```
Brocade-1(config)#vlan 1000 name MCT-VLAN-example
Brocade-1(config-vlan-1000)#tagged ether 1/4 to 1/5 e 1/7 to 1/8
```

Step 3: Configure the cluster

Cluster local configuration uses the cluster ID and RBridgeID for the local switch or router.

Syntax: [no] cluster [<cluster-name>] <cluster-id>

Syntax: [no] rbridge-id <id>

Configuration of the peer device involves the peer's IP address, RBridgeID, and ICL specification. The <cluster-name> variable is optional; the device auto-generates the cluster name as CLUSTER-X when only the cluster ID is specified. The <cluster-id> variable must be the same on both cluster devices.

Syntax: [no] peer <peer-ip> rbridge-id <peer-rbridge> icl <map-icl>

The RBridgeID must be different from the cluster RBridge and any other client in the cluster. The MCT member VLAN is defined as any VLAN of which the ICL is a member.

To configure Brocade-1 for the cluster in the [Figure 101](#) topology, enter the following commands.

```
Brocade-1(config)#cluster SX 4000
Brocade-1(config-cluster-SX)#rbridge-id 3
Brocade-1(config-cluster-SX)#session-vlan 3000
Brocade-1(config-cluster-SX)#keep-alive-vlan 3001
Brocade-1(config-cluster-SX)#icl SX-MCT ethernet 1/7
Brocade-1(config-cluster-SX)#peer 1.1.1.2 rbridge-id 2 icl SX-MCT
Brocade-1(config-cluster-SX)#deploy
```

To configure Brocade-2 for the cluster in the [Figure 101](#) topology, enter the following commands.

```
Brocade-2(config)# cluster SX 4000
Brocade-2(config-cluster-SX)#rbridge-id 2
Brocade-2(config-cluster-SX)#session-vlan 3000
Brocade-2(config-cluster-SX)#keep-alive-vlan 3001
Brocade-2(config-cluster-SX)#icl SX-MCT ethernet 2/5
Brocade-2(config-cluster-SX)#peer 1.1.1.3 rbridge-id 3 icl SX-MCT
Brocade-2(config-cluster-SX)#deploy
```

Step 4: Configure clients

This section describes how to configure clients manually. For instructions on automatic client configuration, refer to [“Setting up cluster client automatic configuration”](#) on page 841.

Client configuration requires the client name, RBridgeID, and CCEP. In the network shown in [Figure 101](#), Client-1 has a three-port LACP trunk (1/1-1/3), while Client-2 has a two-port static trunk (1/1-1/2) towards the MCT cluster.

The client name can be different on the different cluster devices. To configure the client name, enter the following command.

Syntax: [no] client <client-name>

The client RBridgeID must be identical on both of the cluster devices. To configure the client RBridgeID, use the following command.

Syntax: `[no] rbridge-id <id>`

To configure the physical port or static trunk as the client CCEP, use the following command.

Syntax: `[no] client-interface ethernet <slot/port>`

To configure the LACP client CCEP port (you must specify all LACP ports with this command), use the following command.

Syntax: `client-interface link-aggregation ethernet <slot/port> to <slot/port>`

When LACP is automatically enabled on the specified client interfaces, it operates in active mode with the LACP long timeout mode by default.

To change the mode for link aggregation, enter the following command.

Syntax: `[no] client-interface link-aggregation <passive>`

To change the timeout mode for link aggregation, enter the following command.

Syntax: `[no] client-interface link-aggregation <timeout-short>`

To configure Client-1 on Brocade-1 in the [Figure 101](#) topology, enter the following command.

```
Brocade-1(config-cluster-SX)# client client-1
Brocade-1(config-cluster-SX-client-1)#rbridge-id 100
Brocade-1(config-cluster-SX-client-1)#client-interface link-aggregation ether
1/15 to 1/16
Brocade-1(config-cluster-SX-client-1)#deploy
```

To configure Client-1 on Brocade-2 in the [Figure 101](#) topology, enter the following command.

```
Brocade-2(config-cluster-SX)# client client-1
Brocade-2(config-cluster-SX-client-2)#rbridge-id 100
Brocade-2(config-cluster-SX-client-2)#client-interface link-aggregation ether 2/9
Brocade-2(config-cluster-SX-client-2)#deploy
```

To configure Client-2 on Brocade-1 in the [Figure 101](#) topology, enter the following command.

```
Brocade-1(config-cluster-SX)# client client-2
Brocade-1(config-cluster-SX-client-1)#rbridge-id 200
Brocade-1(config-cluster-SX-client-1)#client-interface ether 1/5
Brocade-1(config-cluster-SX-client-1)#deploy
```

To configure Client-2 on Brocade-2 in the [Figure 101](#) topology, enter the following command.

```
Brocade-2(config-cluster-SX)# client client-2
Brocade-2(config-cluster-SX-client-2)#rbridge-id 200
Brocade-2(config-cluster-SX-client-2)#client-interface ether 2/8
Brocade-2(config-cluster-SX-client-2)#deploy
```

Setting up cluster client automatic configuration

Complete the following steps to configure cluster client automatic configuration.

1. Enable the client auto-detect ports on both MCT devices.

```
Brocade-1(config-cluster-SX)#client-auto-detect ethernet 1/15-1/16
```

In the port list, specify all the CCEPs for all potential clients.

2. Start the client auto-detect process on both cluster devices.

```
Brocade-1(config-cluster-SX)#client-auto-detect start
```

Within one minute, the system reports information and errors (if there are mismatches such as an LACP configuration mismatch). You can fix the mismatch while the process is running.

3. Check and fix the automatically detected clients.

```
Brocade-1(config-cluster-SX)#show cluster cluster-SX client-auto-detect
cluster cluster-SX 4000
  rbridge-id 3
  session-vlan 3000
  icl SX-MCT ethernet 1/7
peer 1.1.1.2 rbridge-id 2 icl SX-MCT
client-auto-config ethe 8/1 to 8/2 ethe 8/5 ethe 8/7 eth 8/9
  client-auto-config start
deploy
  client AUTO-FCX624-Router002438769e00
  rbridge-id 3593
  client-interface link-aggregation ethe 8/1 to 8/2 ethe 8/5
  client AUTO-FCX624-Switch008738766700
  rbridge-id 2468
  client-interface static-trunk ethe 8/7 ethe 8/9
!
```

NOTE

At this point, the client configuration does not appear in the running configuration and cannot be modified. Static trunk and LACP configuration are not effective yet.

4. Configure automatically detected clients into the running configuration.

```
Brocade-1(config-cluster-SX)#client-auto-detect config
```

All automatically configured client information is now published into the running configuration and the static trunk configuration will be generated, created, and deployed. LACP will start. By default, clients are in the non-deployed state and the CCEPs will be put into the disable state. Ports that are successfully programmed as CCEP will be removed from the autoconfig-enabled port list. If the port list is empty, which means all ports are configured into clients successfully, the automatic configuration process will be stopped. The original LLDP configuration will be restored. Otherwise, the automatic configuration process will continue only on the ports still left in the list.

Other cluster client automatic configuration commands

You can use the following commands as an alternative to the step-by-step procedure in [“Cluster client automatic configuration”](#) on page 835.

Use the following command to enable or disable cluster client automatic configuration on a range of ports.

Syntax: [no] client-auto-detect Ethernet <x> [to <y>]

Use the following command as an alternate to **client-auto-detect config**. This command also configures automatically detected clients into the running configuration and deploys all of the automatically detected clients.

Syntax: client-auto-detect config deploy-all

Use the following command to start the cluster client automatic configuration. Within one minute of the time that each client is discovered, the client is automatically configured and deployed into the running configuration.

Make sure that the network connection and configuration are in place before using this command.

Syntax: `client-auto-detect start` [config-deploy-all]

Use the following command to stop the current running cluster client automatic configuration process. All auto-detected but unconfigured clients will be cleared.

Syntax: `client-auto-detect stop`

MCT failover scenarios

The following scenarios describe what happens if specific elements in the MCT configuration fail.

- Client interface on one of the MCT cluster devices goes down.
Traffic switches to the other cluster device with minimal traffic loss.
- MCT cluster device goes down.
When an MCT cluster device goes down (for example, due to a power failure), the traffic will fail over to the other MCT cluster device.
- Hitless failover occurs.
The MCT CCEPs stay up during hitless switchover, failover, or upgrade. Link protocols such as UDLD and LACP on CCEPs do not flap. Traffic disruption is minimal (sub-second). The MCT CCP connection will flap once, and MAC is re-synced between the peer devices.
The CCP will go down and come back up again once the hitless failover is completed.
- ICL interface or CCP goes down (keep-alive configured)
If a keep-alive VLAN is used, the devices in the cluster can communicate even if the ICL goes down. If the peer device is reachable over the keep-alive VLAN, the MCT peers perform the master/slave negotiation per client. After negotiation, the slave shuts down its client ports, and the master client ports continue to forward the traffic.

The master/slave negotiation is performed per MCT client on the basis of RBridgeID and client Local or Remote reachability. If the client is reachable from both MCT devices, the higher RBridgeID becomes the master. If the client is reachable from one of the MCT devices only, then the cluster device on which it is reachable becomes the master.

If the peer device is not reachable over the keep-alive VLAN, then both cluster devices will keep forwarding.

NOTE

Brocade recommends using keep-alive VLANs with the MCT configurations. This will provide an alternative reachability if the ICL interface goes down. However, a keep-alive VLAN should not be configured when **bpdu-flood-enable** is configured. Refer to [“BPDU forwarding”](#) on page 850.

- ICL interface or CCP goes down (keep-alive not configured)
When the keep-alive VLAN is not configured, both cluster devices will keep forwarding. Use the **client-isolation strict** command to remove the client interface as soon as the ICL link goes down and completely isolates the client.

- Double failures (for example, when the ICL goes down and the client interface goes down on one of the MCT cluster devices)

Multiple failures could drop traffic in this scenario, even if there is a physical path available.

Cluster failover mode

The following failover modes can be configured with MCT:

- Fast-failover (default) - As soon as the ICL interface goes down the CCP goes down. All the remote MAC addresses are flushed.
- Slow-failover - Even if the ICL interface goes down, the CCP waits for the hold-time before taking the CCP down. Remote MAC addresses are flushed only when the CCP is down.

To disable the fast-failover mode, enter a command such as the following.

```
Brocade-1(config-cluster-SX)#peer 1.1.1.3 disable-fast-failover
```

Syntax: [no] peer <peer-ip> disable-fast-failover

Client isolation mode

NOTE

The CLI will allow modification of the client isolation mode on MCT cluster devices even when the cluster is deployed. You must create the same isolation mode on both cluster devices.

MCT cluster devices can operate in two modes. Both peer devices should be configured in the same mode.

Loose mode (default): When the CCP goes down, the peer device performs the master/slave negotiation. After negotiation, the slave shuts down its peer ports, whereas the master peer ports continue to forward the traffic (keep-alive VLAN configured).

If the keep-alive VLAN is not configured, both peer devices become master and both of the client ports stay up.

```
Brocade-1(config-cluster-SX)#client-isolation loose
```

Strict mode: When the CCP goes down, the interfaces on both the cluster devices are administratively shut down. In this mode, the client is completely isolated from the network if the CCP is not operational.

```
Brocade-1(config-cluster-SX)#client-isolation strict
```

Syntax: [no] client-isolation <loose | strict>

Shutting down all client interfaces

Use the **client-interfaces shutdown** command when performing a hitless upgrade operation. This command can be used to shut down all the local client interfaces in the cluster. This would result in failover of traffic to the peer device.

```
Brocade-1(config-cluster-SX)#client-interfaces shutdown
```

Syntax: [no] client-interfaces shutdown

Using the keep-alive VLAN

CCRR messages are used to exchange information between peer devices. When the CCP is up, CCRR messages are sent over the CCP. When the CCP client reachability is down, you can use the **keep-alive-vlan** command under the cluster context so CCRR messages are periodically sent over the keep-alive VLAN. Only one VLAN can be configured as a keep-alive VLAN. The keep-alive VLAN cannot be a member VLAN of the MCT and this VLAN can be tagged or untagged.

NOTE

Keep-alive VLAN configuration is not allowed when the client isolation mode is strict, and when the keep-alive VLAN is configured, client isolation mode cannot be configured as strict.

```
Brocade-1(config-cluster-SX)#keep-alive-vlan 10
```

Syntax: **[no] keep-alive-vlan** <vlan-id>

The <vlan_id> variable specifies the VLAN range. Possible values are from 1 to 4089.

When the CCP is down, the following results occur.

- If the keep-alive VLAN is configured, then CCRR messages are sent every second over that VLAN.
- When CCP is down and the keep-alive VLAN is configured, master/slave selection is based on following criteria:
 - If one device's CCEPs are up and the peer's CCEPs are down, then the peer with the local CCEPs down becomes the slave.
 - Otherwise, the device with the higher RBridgeID becomes the slave.
- If no packets are received from the peer device for a period of three seconds, then the peer is considered down.
- If the keep-alive VLAN is not configured and both the peer devices are up, then both peers keep forwarding the traffic independently.

Setting keep-alive timers and hold-time

To specify the keep-alive timers and hold time for the peer devices, enter a command such as the following.

```
Brocade-1(config-cluster-SX)# peer 1.1.1.3 timers keep-alive 40 hold-time 120
```

Syntax: **[no] peer** <peer-ip> **timers keep-alive** <keep-alive time> **hold-time** <hold-time>

The <peer-ip> parameter should be in the same subnet as the cluster management interface.

The <keep-alive time> variable can be from 0 to 21845. The default is 10 seconds.

The <hold-time> variable can be from 3 to 65535 and must be at least 3 times the keep-alive time. The default is 90 seconds.

NOTE

The keep-alive VLAN and keep-alive timers are not related. The keep-alive timer is used by CCP.

Layer 2 behavior with MCT

This section describes the Layer 2 behavior when MCT is configured.

MAC operations

This section describes MAC address-related configuration operations.

MAC Database Update

The MAC addresses that are learned locally are given the highest priority or the cost of 0 so they are always selected as the best MAC address.

Each MAC address is advertised with a cost. Low cost MAC addresses are given preference over high cost addresses.

If a MAC address moves from a CCEP port to a CEP port, a MAC move message is sent to the peer and the peer moves the MAC address from its CCEP ports to the ICL links.

If the cost of a MAC address is the same, then the address learned from the Lower RBridgeID wins and is installed in the FDB.

MAC addresses in MCT VLANs are updated across the cluster using MDUP messages.

Cluster MAC types

Cluster Local MAC (CL): MAC addresses that are learned on the MCT VLAN and on CEPs locally. MAC addresses are synchronized to the cluster peer device and are subject to aging.

Cluster Remote MAC (CR): MAC addresses that are learned via MDUP messages from the peer device (CL on the peer). The MAC addresses are always programmed on the ICL port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for these MAC addresses with a cost of 1, and associated with the peer RBridgeID.

Cluster Client Local MAC (CCL): MAC addresses that are learned on the MCT VLAN and on CCEPs.

The MAC addresses are synchronized to the cluster peer device and are subject to aging. A MDB entry is created for these addresses with a cost of 0 and are associated with the client and cluster RBridgeIDs.

Cluster Client Remote MAC (CCR): MAC addresses that are learned via MDUP message from the peer device (CCL on the peer). The MAC addresses are always programmed on the corresponding CCEP port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for the MAC addresses with the cost of 1, and are associated with the client and peer RBridgeIDs.

Cluster Multi-Destination Local MAC (CML): A static MAC entry is configured locally on the MCT VLAN. Any static MAC address configured on MCT VLAN will have ICL added by default so it will automatically become a multi-destination MAC entry. The local configuration generates a local MDB, any CML entry can still have up to 2 MDBs associated with, one local and one remote. The remote MDB comes with the remote static configuration for the same (MAC, VLAN). If the dynamic MAC and static configuration co-exist, the dynamic MAC address will be removed whether it is locally learnt or learned from MDUP. The port list of a CML entry has a ICL port, the client ports from the client list in the local and remote (if exists) configuration, and all locally configured CEP ports.

Cluster Multi-Destination Remote MAC (CMR): A static MAC entry is configured on MCT VLAN on the peer side and there is no associated local configuration. The CMR entry has only the remote MDB. The port list of a CMR entry has an ICL port, and all the client ports from the client list in the remote configuration. When there is local configuration for the same entry, the CMR is converted to CML.

MAC aging

Only the local MAC entries are aged on a cluster device. The remote MAC address entries are aged based on explicit MDUP messages only.

The remote MAC addresses learned through MDUP messages are dynamic addresses with the exception that they never age from FDB.

MAC flush

If the CEP is down, the MAC addresses are flushed and individual MAC deletion messages are sent to the peer device.

If the CCEP local port is down, the MAC addresses are flushed locally and individual MAC deletion messages are sent to the peer device.

If the **clear mac** command is given, all the MDB and FDB are rebuilt.

If the **clear mac vlan** command is given, all the local MDB and FDB are rebuilt for that VLAN.

MAC movement happens normally on the local device.

CEP to CCEP MAC movement – MAC movement normally happens on the local device, and deletes all the other MDBs from the peer to create a new local MDB.

MAC show commands

To display all the cluster local MAC address entries for a cluster, use the **show mac cluster** command.

```
Brocade#show mac cluster 1000
Total Cluster Enabled(CL+CR+CCL+CCR) MACs: 1
Total Cluster Local(CL) MACs: 1
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
Total active entries from all ports = 1
Total static entries from all ports = 3
MAC-Address      Port      Type      Index  MCT-Type  VLAN
1111.2222.3333   8/1       Static    4254    CML       20
1111.2222.3333   8/3       Static    4254    CML       20
1111.2222.3333   8/13      Static    4254    CML       20
```

Syntax: **show mac** [**cluster** <id> | <name> <local> | <remote>]

MAC clear commands

To clear all MAC addresses in the system, enter the following command.

```
Brocade#clear mac
```

Syntax: **clear mac**

Clearing cluster-specific MAC addresses

To clear cluster-specific MAC addresses in the system, enter a command such as the following.

```
Brocade#clear mac cluster AGG-1 local
```

Syntax: clear mac cluster <cluster-id> | <cluster-name> { local | remote }

Clearing client-specific MAC addresses

To clear client-specific MAC addresses in the system, enter a command such as the following.

```
Brocade#clear mac cluster AGG-1 client 1 local
```

Syntax: clear mac cluster <cluster-id> | <cluster-name> client <client-name> { local | remote }

Clearing VLAN-specific MAC addresses

To clear VLAN-specific MAC addresses in the system, enter a command such as the following.

```
Brocade#clear mac vlan 2
```

Syntax: clear mac vlan <vlan_id>

Clearing MCT VLAN-specific MAC addresses

To clear MCT VLAN-specific MAC addresses in the system, enter a command such as the following.

```
Brocade#clear mac cluster AGG-1 vlan 1 local
```

Syntax: clear mac cluster <cluster_id> | <cluster-name> vlan <vlan_id> {local | remote}

Clearing cluster client vlan-specific MACs

To clear cluster client-specific MAC addresses in the system, enter a command such as the following.

```
Brocade#clear mac cluster AGG-1 vlan 2 client 1 local
```

Syntax: clear mac cluster <cluster_id> | <cluster-name> vlan <vlan_id> client <client_name> {local | remote}

Displaying MDUP packet statistics

To display the statistics of MDUP packets, enter a command such as the following.

```
Brocade#show mac mdup-stats
MDUP Information
=====
MDUP Data buffers in queue : 0
MDUP Statistics
=====
MDUP Update Messages sent: 7
Add Mac sent: 20
Del Mac sent: 0
Move Mac sent: 0
MDUP Mac Info Messages sent: 1
MDUP Flush Messages sent: 1
MDUP Synch Messages sent: 0
MDUP Update Messages received: 3
Add Mac received: 40
Del Mac received: 0
Move Mac received: 0
MDUP Mac Info Messages received: 0
MDUP Flush Messages received: 0
MDUP Synch Messages received: 0
```

Syntax: show mac mdup-stats

Syncing router MAC addresses to peer MCT devices

The MCT cluster device uses a router MAC address to identify the packets that are addressed to the switch. Such packets may be received by a peer cluster device. The peer device switches packets over the ICL to the local MCT device to be routed properly.

Dynamic trunks

MCT client creates a single dynamic trunk group towards the MCT cluster devices. The dynamic trunk group consists of two trunk groups, each of which is configured on one of the MCT devices. A dynamic trunk group runs Link Aggregation Control Protocol (LACP).

For the two dynamic trunk groups of the MCT to behave as a single trunk group from the MCT client's perspective, both of the dynamic trunk groups should have the same LACP system ID and key, referred to as the MCT system ID and MCT key.

The LACP system ID in the FSX product normally comes from the port MAC address. To support LACP over MCT, it is necessary to obtain the ID in another way. To do so, MCT uses a pre-defined algorithm.

NOTE

Each MCT cluster device has a unique cluster ID, and one MCT client ID. The LACP key is predefined from the client ID and cluster ID. The user cannot change the key.

MCT does not involve stacking, and control protocol synchronization is minimal. The LACP runs independently on the cluster devices.

Port loop detection

Loop detection can be used in an MCT topology to detect Layer 2 loops due to misconfigurations, for example, on the client side when MCT links are not configured as trunk links on the MCT-unaware client.

In MCT, the ICL link should be up at all times to prevent the cluster from going down. They should not be shut down when a loop is detected in a network. Instead other available ports (CCEPs) should be shut down. If loop detection BDPUs are received on the ICL port, then instead of shutting down the ICL links, all the CCEPs will be error-disabled and the user will be notified with the following log message.

```
Loop-detection:Packet received on ICL port <port_number> for vlan
<vlan_id>.Errdisable CCEPs.
```

Strict mode loop detection can be enabled on ICL ports. This is because in strict mode, a port is disabled only if a packet is looped back to that same port. Strict mode overcomes specific hardware issues where packets are echoed back to the input port. This process assists in detecting hardware faults on ICL ports.

Loop-detection can be enabled on MCT and non-MCT VLANs simultaneously. There is no change in loop detection behavior when enabled on non-MCT VLANs.

MCT Layer 2 protocols

Keep the following information in mind when configuring Layer 2 protocols with MCT.

MRP

- An ICL interface cannot be configured as an MRP secondary interface or vice versa, because the ICL cannot be BLOCKING.
- MRP cannot be enabled on MCT CCEP port and vice versa.

STP/RSTP

- STP is not recommended to be configured on MCT VLANs at MCT cluster devices. By default, the spanning tree is disabled in the MCT VLANs. If the network topology may be creating Layer 2 loops through external connections, STP could be enabled on switches outside the MCT cluster to prevent the Layer 2 loop. The MCT cluster devices will perform a pass-through forwarding of STP BPDUs received through its ports in the MCT VLAN.
- In rare cases in which the network topology consists of Layer 2 loops outside the MCT cluster that require STP/RSTP to be enabled on MCT VLANs in the cluster, the CCEPs will always be in the spanning tree disabled state.
- The STP/RSTP algorithms have been modified such that ICL never goes to blocking. The ICL guard mechanism ensures that if ICL is going into a blocking state, then the port on which the superior BPDUs are being received is moved to blocking state and the ICL guard timer starts running on it. This timer runs as long as superior BPDUs are received on this interface. As long as this timer runs on an interface, the superior BPDUs are dropped.
- The new BLK_BY_ICL STP state indicates that the superior BPDUs were received on this interface, which could have led to blocking of the ICL interface, with the result that the CL port guard mechanism has been triggered on this port.
- In a 802.1s MSTP deployment, Brocade recommends disabling spanning tree on the MCT cluster devices at the global level. MSTP cannot be configured on individual cluster devices.
- An MCT cluster can support up to 32 spanning tree instances.

BPDU forwarding

If the network deploys single STP or IEEE 802.1s (MSTP), both the MCT cluster devices must be configured using the **bpdu-flood-enable** command to flood the single STP/MSTP BPDUs in the SSTP/MSTP domain (forward to all of the ports in the cluster switch irrespective of VLAN.)

Syntax: [no] **bpdu-flood-enable**

When **bpdu-flood-enable** is configured, there should not be any links other than the ICL (including the keep-alive VLAN link) connecting the two MCT cluster devices. If there is an additional link, then the flooded BPDU will cause a loop and high CPU utilization.

Protocol-based VLANs

Protocol and subnet VLANs can be configured on MCT VLANs, however, ICL and CCEPs cannot be configured as dynamic members of protocol based VLANs (and vice versa). ICL and CCEP can either be excluded or static members of protocol based VLANs. CCEPs can be configured as dynamic or static, or exclude members of protocol based VLANs.

In a cluster, both cluster devices should have exactly same protocol VLAN membership with respect to ICL and CCEP. ICL and CCEPs should be configured with same type of protocol/VLAN membership, although there is no such restriction from the CLI.

Uplink switch

Uplink switch is supported on MCT VLANs. ICLs and CCEPs can be configured as uplink-switch ports. Both cluster devices should have exactly same uplink-switch port memberships with respect to the ICL and CCEPs.

Layer 2 multicast snooping over MCT

To support multicast snooping over MCT, the ICL port is used to synchronize the following information between the cluster devices using MDUP:

- MAC - forward entries (mcache entries on MCT VLAN).
- IGMP/MLD Join/Leave (control packets on MCT VLAN).
- PIM-SM Join/Prune (control packets on MCT VLAN).

IGMP/MLD snooping

Snooping can be configured globally or at the VLAN level. Each cluster device in the MCT VLAN can be configured either as active or passive. There is no restriction for cluster devices to run active-active or passive-passive configuration.

The following commands show configuration commands for the VLAN level (IGMP), VLAN level (MLD), global level (IGMP/MLD), and for PIM-SM.

VLAN level (IGMP)

```
Brocade(config)#vlan 100
Brocade(config-vlan-100)#multicast active/passive
```

VLAN level (MLD)

```
Brocade(config-vlan-100)# mld-snooping active/passive
```

Global Level (IGMP/MLD)

```
Brocade(config)#ip multicast active/passive
Brocade(config)#ipv6 mld-snooping active/passive
```

PIM-SM snooping (configured only on a VLAN and requires IGMP snooping to run in a passive mode):

```
Brocade(config)#vlan 100
Brocade(config-vlan-100)#multicast passive
Brocade(config-vlan-100)# multicast pimsm-snooping
```

IGMP/MLD snooping behavior on MCT cluster devices

- Local information is synchronized to the MCT peer device using CCP. The information includes Mcache/FDB entry (on arrival of data traffic), joins/leaves, dynamic router ports, and PIM-SM snooping joins/prunes.
- Native control packets (joins/leaves) that are received are processed by protocol code, and also forwarded out if required.
- All control/data traffic is received on ICL. The traffic is forwarded out of CCEP only if the remote CCEP is down; otherwise, it is dropped by the egress filters on CCEP.
- ICL is added as OIF by default whenever the CCEP is involved as either source or receiver. This provides faster convergence during MCT failover.
- For IGMP/MLD joins/leaves:
 - The control packets only received on CCEP are synced to the MCT peer using CCP.
 - The control packets received on CEP are not synced to MCT peer using CCP.
- Static groups and static router ports configured on CCEP are not synced across to the MCT peer. For these features to work correctly, they must be manually configured on the respective CCEP of both the cluster nodes.

MCT failover handling for Layer 2 multicast over MCT

The following failover scenarios may occur. Refer to [“MCT failover scenarios”](#) on page 843 for other types of failover scenarios.

- Local CCEP Down EVENT:
 - Outgoing traffic on local CCEP will now go through ICL and out of the remote CCEP.
 - Incoming traffic on local CCEP will now ingress through the remote CCEP, and then ingress through ICL locally.
- Local CCEP Up EVENT:
 - Outgoing traffic on remote CCEP (after egressing through local ICL) will now start going out of local CCEP.
 - Incoming traffic from client through ICL (after ingressing on remote CCEP) will now switch back to local CCEP (this is true only if the client trunk hashing sends the traffic towards local CCEP).
- CCP (Cluster communication protocol) Down EVENT:
 - All related information (i.e. IGMP/MLD group, mcache, dynamic router port, pim-sm snooping entry) that were synced from the peer device will now be marked for aging locally.
- CCP (Cluster communication protocol) Up EVENT:
 - All related information (i.e. IGMP/MLD group, mcache, dynamic router port, pim-sm snooping entry) that were locally learned will be synced to the peer device.

PIM-SM snooping over MCT

- PIM-SM snooping can be configured only on a VLAN. It requires IGMP snooping to be running in passive mode. IPv6 snooping is not supported.
- Router ports can be configured on a VLAN or globally. They can be learned dynamically on the port where the query is received or configured statically.

- Both MCT1 devices must run pimsm-snoop.
- PIM messages are forwarded by way of hardware.
- PIM join/prune is synced to the peer cluster device using CCP.
- PIM prune is processed only if indicated by the peer cluster device.
- PIM join/prune received natively on ICL is ignored.
- PIM hello is not synced, but is received natively on ICL.
- PIM port/source information is refreshed on both cluster devices by syncing PIM messages and ages out if not refreshed.

Forwarding entries for PIM-SM multicast snooping

Table 144 and Table 145 list the forwarding entries for PIM-SM multicast snooping.

TABLE 144 Forwarding entries (*,G)^a

Event	MCT-1	MCT-2
No-Join	(*,G)->blackhole	(*,G)->blackhole
(S,G)Join on (MCT-1)CEP	(*,G)->CEP [s] ^b	(*,G)->ICL [s]
(S,G)Join on (MCT-2)CEP	(*,G)->ICL [s]	(*,G)->CEP [s]
(S,G)Join on (MCT-1)CCEP	(*,G)->CCEP [s], ICL [s]	(*,G)->CCEP [s], ICL [s]
(S,G)Join on (MCT-2)CCEP	(*,G)->CCEP[s], ICL [s]	(*,G)->CCEP [s], ICL [s]

a. *ICL: The ICL port is added as default whenever CCEP is in OIF. The data traffic receiving from ICL port will be filtered out by egress filter (dynamically programmed) on CCEPs.

b. [s]: denotes sources maintained on port hash-list.

TABLE 145 Forwarding entries (S,G)^a

Event	MCT-1	MCT-2
No-Join	(S,G)->blackhole	(S,G)->blackhole
Join (MCT-1)CEP	(S,G)->CEP	(S,G)->ICL
Join (MCT-2)CEP	(S,G)->ICL	(S,G)->CEP
Join (MCT-1)CCEP	(S,G)->CCEP, ICL	(S,G)->CCEP, ICL
Join (MCT-2)CCEP	(S,G)->CCEP, ICL	(S,G)->CCEP, ICL

a. *ICL: The ICL port is added as default whenever CCEP is in OIF. The data traffic receiving from ICL port will be filtered out by egress filter (dynamically programmed) on CCEPs.

Layer 3 behavior with MCT

Table 146 lists the type of Layer 3 support available with MCT. Note that routing protocols are not supported on ICL and CCEPs. At the edge network, it is highly recommended to configure VRRP/VRRP-E when MCT is enabled.

TABLE 146 Layer 3 Feature Support with MCT

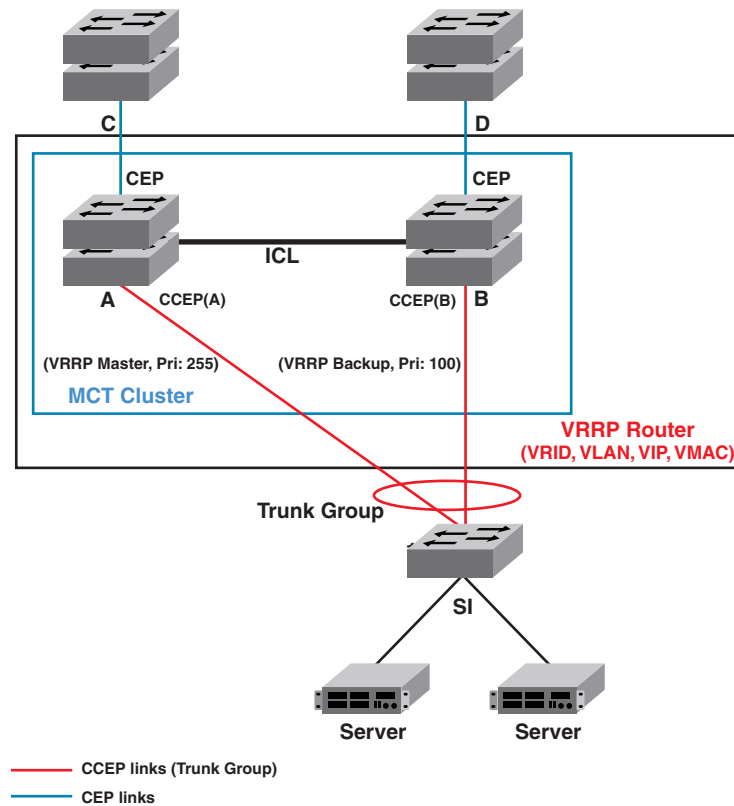
Feature	Sub-feature	Session VLAN VE	Member VLAN VE	Design Philosophy
ip	access-group ^a	Yes	Yes	<ul style="list-style-type: none"> Only features that are relevant for MCT management are supported on session VLAN VE Layer 3 unicast dynamic routing not supported on member VLAN VE
	address	Yes	Yes	
	arp-age	Yes	Yes	
	bootp-gateway	Yes	Yes	
	directed-broadcast	Yes	Yes	
	dvmrp	No	No	
	encapsulation	Yes	Yes	
	follow	No	No	
	helper-address	Yes	Yes	
	icmp	Yes	Yes	
	igmp	No	No	
	irdp	No	Yes	
	local-proxy-arp	No	Yes	
	metric	No	Yes	
	mtu	Yes	Yes	
	multicast-boundary	No	No	
	ospf	No	No	
	pim	No	No	
	pim-sparse	No	No	
	policy	No	Yes	
	proxy-arp	No	Yes	
	redirect	No	Yes	
	rip	No	No	
	tcp	Yes	Yes	
	tunnel	No	No	
	use-acl-on-arp	Yes	Yes	
	vrrp	No	Yes	
	vrrp-extended	No	Yes	
ipv6		No	No	<ul style="list-style-type: none"> IPv6 is not supported for MCT management IPv6 not supported on member VLAN VE

a. Only Ingress ACLs are supported on VLAN VE that have CCEPs. Egress ACLs are not supported on VLAN VE with CCEPs.

Layer 3 unicast over MCT

The following examples show a sample configuration for the Layer 3 unicast configuration shown in [Figure 102](#).

FIGURE 102 Configuration for Layer 3 unicast



Device A

MCT Configuration

```
!
vlan 10 by port
    tagged ethe 3/1
    router-interface ve 10
!
interface ve 10
    ip address 10.1.1.1 255.255.255.0
!
cluster L3UC 1
    rbridge-id 101
    session-vlan 10
    icl L3icl ethernet 3/1
    peer 10.1.1.2 rbridge-id 102 icl L3icl
    deploy
    client s1
    rbridge-id 300
```

```

client-interface ethernet 3/3
deploy
!

```

VRRP-E Configuration

```

!
vlan 100 by port
    tagged ethe 3/1 ethe 3/3
    router-interface ve 100
!
router vrrp-extended
!
interface ve 100
    ip address 100.1.1.1 255.255.255.0
    ip vrrp-extended vrid 1
    backup priority 255
    ip-address 100.1.1.254
    enable
!

```

Device B

MCT Configuration

```

!
vlan 10 by port
    tagged ethe 3/1
    router-interface ve 10
!
interface ve 10
    ip address 10.1.1.2 255.255.255.0
!
cluster L3UC 1
    rbridge-id 102
    session-vlan 10
    icl L3icl ethernet 3/1
    peer 10.1.1.1 rbridge-id 101 icl L3icl
    deploy
    client s1
        rbridge-id 300
        client-interface ethernet 3/25
        deploy
!

```

VRRP-E Configuration

```

!
vlan 100 by port
    tagged ethe 3/1 ethe 3/25
    router-interface ve 100
!
router vrrp-extended
!
interface ve 100
    ip address 100.1.1.2 255.255.255.0
    ip vrrp-extended vrid 1
    backup
    ip-address 100.1.1.254
    enable
!

```

Switch S1

```

!
trunk ethe 3 to 4
!
vlan 100 by port
  tagged ethe 3 to 4
  router-interface ve 100
!
interface ve 100
  ip address 100.1.1.100 255.255.255.0
!

```

MCT for VRRP or VRRP-E

A simple MCT topology addresses resiliency and efficient load balancing in Layer 2 network topologies. To interface with a Layer 3 network, MCT is configured with Virtual Router Redundancy Protocol (VRRP) to add redundancy in Layer 3. The standard VRRP mode is master-backup and all traffic is forwarded through the master. In VRRP-E server virtualization, multiple VRRP standby devices are supported and each device can be configured to route to an upstream Layer 3 network. This provides efficient deployment for both Layer 2 and Layer 3 networks.

The MCT device that acts as backup router needs to ensure that packets sent to a VRRP or VRRP-E virtual IP address can be Layer 2-switched to the VRRP-E master router for routing. The VRRP or VRRP-E backup learns the VMAC while processing the VRRP hello message from the VRRP master. Both data traffic and VRRP or VRRP-E control traffic travel through the ICL unless the short-path forwarding feature is enabled (VRRP-E only).

Layer 3 traffic forwarding from CCEPs to CCEPs

Traffic destined to the CCEPs from the client or CEPs follow the normal IP routing on both master and backup devices. By default, the best route should not involve the ICL link. Only when the local CCEP is down will the traffic be re-routed to pass through ICL link.

Layer 3 traffic forwarding from CCEPs to CEPs

For Layer 3 forwarding to work on MCT devices, a dynamic trunk must be configured on the MCT client. VRRP/VRRP-E and VRRP-E2 SPF should be enabled, if required. Routes should be statically configured on the MCT cluster devices for all member VLANs.

If VRRP is deployed or VRRP-E is deployed without the short path forwarding feature on the VRRP-E backup, it is likely that almost 50% (and 100% in the worst case) of CCEP to CEP traffic can pass through the ICL from the backup to the master device. This fact should be considered when designing ICL capacity in the network.

ARP resolution

ARP resolution is initially done through the ICL if the S1 MAC address is not already known on the CCEP at A. When the MDUP message from the cluster peer device moves the S1 MAC from ICL to CCEP, the ARP is also moved.

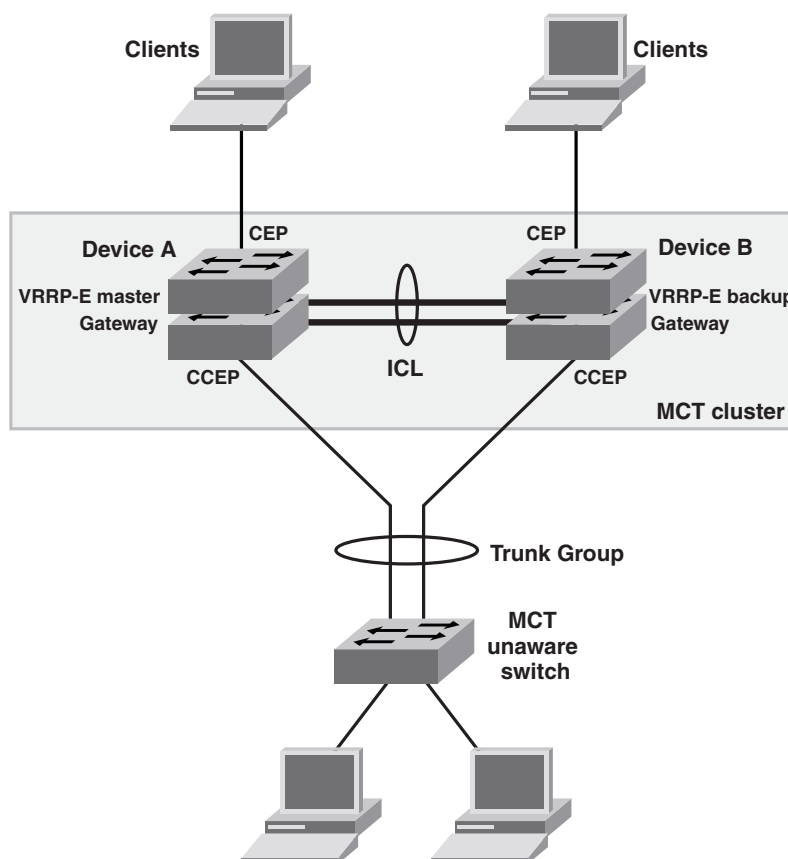
If S1 triggers an ARP request, it generally does so for the default gateway address (virtual IP address if VRRP is deployed). This ARP request can reach A either directly from S1, or through B.

- If the ARP request reaches A directly, it replies through the same port on which it learned S1's MAC address.
- If the request is by way of B, S1's ARP response will be learned on the ICL first, then it will move to the CCEP link when the MDUP message for S1's MAC address is received from B.

VRRP or VRRP-E configuration with MCT

In [Figure 103](#), MCT devices A and B both need to ensure packets sent to VRRP-E virtual IP address can be Layer 2-switched to the VRRP-E master router for routing. Both data traffic and VRRP-E control traffic travel through ICL, unless the short-path forwarding feature is enabled. The VRRP or VRRP-E virtual MAC address will be learned on ICL ports on backup routers through the ICL. The VRRP or VRRP-E master router will broadcast hello packets to all VLAN member ports, including ICL ports.

FIGURE 103 Example of MCT configuration with Layer 2 switching



Configuration considerations

- MCT devices must have complete routing information using static routes for Layer 3 forwarding.

- For MCT devices configured with VRRP or VRRP-E, track-port features can be enabled to track the link status to the core devices so the VRRP or VRRP-E failover can be triggered.
- It is not supported to configure several Layer 3 features on VE of the session VLAN. If already configured, such a VLAN cannot be made the session VLAN.
- It is not supported to configure UC/MC routing protocols and the IP follow feature on VEs of member VLANs. If already configured, such a VLAN cannot be made a member VLAN.
- IPv6 configurations are not supported on VEs of session and member VLANs,
- Route-only ports cannot be used as CCEP and ICL ports.
- Global route-only configuration is mutually exclusive with MCT cluster configuration.
- It is not supported to use MCT management interface IPs for tunnel source.
- It is not supported to configure static and policy based routes using VE on MCT session VLAN.
- Configurations to redistribute connected routes will not advertise IPs on MCT management interface.
- IP addresses on the MCT management interface should not be used for BGP peers on neighboring devices.
- IP addresses on the MCT management interface should not be used for static configurations on neighboring devices.
- The track port feature should be used for VRRP switchover and controlling the validity of SPF feature.
- Up to 64 VRRP or VRRP-E instances are supported on an MCT cluster; however, with Jumbo enabled, a maximum of 32 VRRP or VRRP-E instances is supported on an MCT cluster.

NOTE

Brocade recommends disabling ICMP redirect globally to avoid unintended CPU forwarding of traffic when VRRP or VRRP-E is configured.

Layer 3 traffic forwarding behaviors

When one MCT device acts as a VRRP or VRRP-E master router and the peer device is VRRP or VRRP-E backup, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be Layer 2-switched to the VRRP-E master device for routing.
- The VRRP-E MAC address will be learned by the other MCT device that acts as backup router.
- Both data traffic and VRRP-E control traffic received by the VRRP backup from MCT client will need to travel through ICL unless the short-path forwarding feature is enabled.

When both MCT devices act as the VRRP or VRRP-E backup routers, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be Layer 2-switched to the VRRP-E master router for routing.
- The VRRP-E MAC address will be learned by both MCT devices acting as backup routers.
- Both data traffic and VRRP-E control traffic will travel through the links connecting them to the VRRP master.

VRRP-E short-path forwarding and revertible option

Under the VRRP-E VRID configuration level, use the **short-path-forwarding** command. If the revertible option is not enabled, the default behavior will remain the same. Use the following command to enable short path forwarding.

The **track-port** command will monitor the status of the outgoing port on the backup. It will revert back to standard behavior (no short-path forwarding) temporarily even if short-path forwarding is configured.

```
Brocade(config-if-e1000-vrid-2)#short-path-forwarding revert-priority 60
```

Syntax: [no] **short-path-forwarding** [revert-priority <value>]

Use the supplied priority value as a threshold to determine if the **short-path-forwarding** behavior should be effective or not. If one or more ports tracked by the **track-port** command go down, the current priority of VRRP-E will be lowered by a specific amount configured in the **track-port** command for each port that goes down.

Once the current-priority is lower than the threshold, short path forwarding will be temporarily suspended and revert back to the regular VRRP-E forwarding behavior (non-short path forwarding behavior).

The reverting behavior is only temporary. If one or more of the already down ports tracked by the **track-port** command come back, it is possible that the current priority of VRRP-E will be higher than the threshold again and the **short-path-forwarding** behavior will be resumed.

Displaying MCT information

This section describes the commands available to display information about MCT configuration and operation.

Displaying peer and client states

Use the **show cluster config** command to display the peer device and client states.

```
Brocade#show cluster SXR122 config
cluster SXR122 100
rbridge-id 100
session-vlan 1
keep-alive-vlan 3
icl SXR122-MCT ethernet 1/1
peer 172.17.0.2 rbridge-id 101 icl SXR122-MCT
deploy
client KL134
rbridge-id 14
client-interface ethernet 1/23
deploy
client AGG131
rbridge-id 10
client-interface ethernet 12/2
deploy
client FOX135
rbridge-id 15
client-interface ethernet 1/25
deploy
```

Syntax: `show cluster <cluster-name/cluster-id> config`

Displaying state machine information

Use the **show cluster client** command to display additional state machine information including the reason for Local CCEP down. You can optionally specify an individual cluster and client.

```
Brocade#show cluster 1 client
Cluster 1 1
=====
Rbridge Id: 101, Session Vlan: 3999, Keep-Alive Vlan: 4001
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range: 100 to 105
Active Member Vlan Range: 100 to 105
MCT Peer's Reachability status using Keep-Alive Vlan: Peer Reachable

Client Info:
-----
Client: c1, rbridge-id: 300, Deployed
Client Port: 3/11
State: Up
Number of times Local CCEP down: 0
Number of times Remote CCEP down: 0
Number of times Remote Client undeployed: 0
Total CCRR packets sent: 4
Total CCRR packets received: 3
```

Syntax: `show cluster <cluster_name/cluster_id> client [<client_name/client_RbridgeID>]`

Table 147 shows the reasons for local CCEP down.

TABLE 147 Reasons for Local CCEP down

Reason for Local CCEP down	Meaning
client-interfaces shutdown	Command is configured.
client-isolation strict	Command is configured.
Deploy mismatch	Client is not deployed remotely.
Slave state	Client is in slave state when CCP is down.
cluster and client undeployed	Neither the cluster nor client is deployed.
cluster undeployed	Cluster is not deployed.
client undeployed	Client is not deployed.

Displaying cluster, peer, and client states

Use the **show cluster ccp peer** command to display cluster, peer device, and client states. You can optionally specify an individual cluster and request additional details.

```
Brocade#show cluster 1 ccp peer
...
PEER IP ADDRESS          STATE          UP TIME
-----
1.1.1.1                  OPERATIONAL    0 days: 2 hr:25 min:16 sec

Brocade (config-cluster-SX_1)# show cluster 1 ccp peer detail
*****Peer Session Details*****
IP address of the peer          1.1.1.1
Rbridge ID of the peer          100
Session state of the peer        OPERATIONAL
Next message ID to be send       287
Keep Alive interval in seconds   30
Hold Time Out in seconds         90
Fast Failover is enable for the session
UP Time                          0 days: 2 hr:22 min:58 sec
Number of tcp packet allocations failed 0
Message      Init      Keepalive  Notify      Application  Badmessages
Send         3         2421     2           53           0
Receive      3         2415     0           37           0
TCP connection is up
TCP connection is initiated by    1.1.1.2
TCP connection tcbHandle not pending
TCP connection packets not received
*****TCP Connection Details*****
TCP Connection state: ESTABLISHED      Maximum segment size: 1436
Local host: 1.1.1.2, Local Port: 12203
Remote host: 1.1.1.1, Remote Port: 4175
ISentSeq: 1867652277 SendNext: 1867660731 TotUnAck: 0
TotSent: 8454 ReTrans: 9 UnAckSeq: 1867660731
IRcvSeq: 3439073167 RcvNext: 3439078415 SendWnd: 16384
TotalRcv: 5248 DupliRcv: 16 RcvWnd: 16384
SendQue: 0 RcvQue: 0 CngstWnd: 1452
```

Syntax: **show cluster** [*<cluster_name/cluster-id>*] **ccp peer** [*detail*]

Displaying information about Ethernet interfaces

Use the **show interface ethernet** command to display information about Ethernet interfaces. The MCT-related information is shown in bold in the following example.

```
Brocade#show interface ethernet 7/1
...
GigabitEthernet7/1 is disabled, line protocol is down
Hardware is GigabitEthernet, address is 0024.3822.8260 (bia 0024.3822.8260)
Configured speed auto, actual unknown, configured duplex fdx, actual unknown
Configured mdi mode AUTO, actual unknown
Member of L2 VLAN ID 1, port is untagged, port state is DISABLED
BPDU guard is Disabled, ROOT protect is Disabled
Link Error Dampening is Disabled
STP configured to ON, priority is level0
Flow Control is config enabled, oper disabled, negotiation disabled
```

```

Mirror disabled, Monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 96 bits-time, IPG GMII 96 bits-time
MTU 1500 bytes, encapsulation Ethernet
ICL port for icl1 in cluster id 1

```

```

300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runs, 0 giants
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions
Relay Agent Information option: Disabled

```

```
show interface ethernet 7/3
```

```

GigabitEthernet7/3 is disabled, line protocol is down
Hardware is GigabitEthernet, address is 0024.3822.8262 (bia 0024.3822.8262)
Configured speed auto, actual unknown, configured duplex fdx, actual unknown
Configured mdi mode AUTO, actual unknown
Member of L2 VLAN ID 1, port is untagged, port state is DISABLED
BPDU guard is Disabled, ROOT protect is Disabled
Link Error Dampening is Disabled
STP configured to ON, priority is level0
Flow Control is config enabled, oper disabled, negotiation disabled
Mirror disabled, Monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 96 bits-time, IPG GMII 96 bits-time
MTU 1500 bytes, encapsulation Ethernet
CCEP for client c149_150 in cluster id 1

```

```

300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runs, 0 giants
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions
Relay Agent Information option: Disabled

```

Syntax: show interface ethernet <x/y>

Displaying STP information

Use the **show span** command to display STP information for an entire device. The MCT-related information is shown in bold in the following example.

```
Brocade#show span
...
STP instance owned by VLAN 10
Global STP (IEEE 802.1D) Parameters:
```

VLAN ID	Root ID	Root Cost	Root Port	Root Prio	Max He- rity	Ho- Age	Fwd llo	Last ld	Chg dly	Bridge cnt	Address
		Hex	sec	sec	sec	sec	sec				
10	00000012f2aacf84	00243822226e	4	7/15	8000	20	2	1	15	10	1

```
Port STP Parameters:
```

Port Num	Prio	Path rity	State Cost	Fwd	Design Trans	Designated Cost	Designated Root
7/15	80	4	FWD_BY_MCT	1	0	00000012f2aacf84	00000012f2aacf84
7/21	80	4	BLK_BY_ICL	1	4	00000012f2aacf84	800000243822226e
7/23	80	4	FORWARDING	7	4	00000012f2aacf84	800000243822226e

```
STP instance owned by VLAN 15
Global STP (IEEE 802.1D) Parameters:
```

VLAN ID	Root ID	Root Cost	Root Port	Root Prio	Max He- rity	Ho- Age	Fwd llo	Last ld	Chg dly	Bridge cnt	Address
		Hex	sec	sec	sec	sec	sec				
15	00000012f2aacf84	00243822226e	4	7/15	8000	20	2	1	15	10	1

```
Port STP Parameters:
```

Port Num	Prio	Path rity	State Cost	Fwd	Design Trans	Designated Cost	Designated Root
7/15	80	4	BLK_BY_MCT	1	0	00000012f2aacf84	00000012f2aacf84
7/21	80	4	BLK_BY_ICL	1	4	00000012f2aacf84	800000243822226e
7/23	80	4	FORWARDING	7	4	00000012f2aacf84	800000243822226e

Syntax: **show span** [vlan <vlan-id>] | [pvst-mode] | [<num>] | [detail [vlan <vlan-id> [Ethernet [<stack-unit><slotnum>]/<portnum>] | <num>]]

Displaying information for multicast snooping

Use the **show ip multicast pimsm-snooping** command to display port numbers for multicast snooping. Refer to [Figure 107](#) for port numbers. In the example, join is received from CCEP on mct-1 and CEP on mct-2.

(S,G) entry:

```
Brocade# show ip multicast pimsm-snooping
vlan 100, has 1 caches.
1 (11.0.0.2 224.10.10.10) has 2 pim join ports out of 2 OIF
  7/3 (age=10), 7/5 (age=10),
```

```
Brocade# show ip multicast pimsm-snooping
vlan 100, has 1 caches.
1 (11.0.0.2 224.10.10.10) has 3 pim join ports out of 3 OIF
  3/8 (age=0), 3/3 (age=50), 3/7 (age=50),
```

(*G) entry:

```
Brocade# show ip multicast pimsm-snooping
vlan 100, has 1 caches.
1 (* 224.10.10.10) has 2 pim join ports out of 2 OIF
  7/5 (age=10), 7/3 (age=10),
  7/5 has 1 src: 11.0.0.2(10)
  7/3 has 1 src: 11.0.0.2(10)
```

```
Brocade# show ip multicast pimsm-snooping
vlan 100, has 1 caches.
1 (* 224.10.10.10) has 3 pim join ports out of 3 OIF
  3/3 (age=20), 3/7 (age=20), 3/8 (age=20),
  3/3 has 1 src: 11.0.0.2(20)
  3/7 has 1 src: 11.0.0.2(20)
```

Syntax: show ip multicast pimsm-snooping

Use the **show ip multicast cluster** commands to display information about multicast snooping activity.

In the following command, YES indicates that reports/leaves were received by locally (processing native control packets).

```
Brocade(config)#show ip multicast cluster group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL100 : 1 groups, 1 group-port
group p-port ST QR life mode source local
1 225.1.1.1 e5/5 no no 200 EX 0 YES
2 225.1.1.1 e5/10 no no 200 EX 0 YES
```

In the following command, NO indicates that reports/leaves were received remotely. In this case, a join was received on the CCEP of the MCT peer device. Native control packets were processed by the peer device and then the entries were synched over MDUP to this cluster device.

```
Brocade(config)#show ip multicast cluster group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL100 : 1 groups, 1 group-port
group p-port ST QR life mode source local
1 225.1.1.1 e1/10 no no 200 EX 0 NO
2 225.1.1.1 e1/10 no no 200 EX 0 NO
```

The following command displays information about the IGMP multicast mcache. It is used to verify if FDB is programmed when a data packet arrives.

```
Brocade(config)#show ip multicast mcache
Example: (S G) cnt=: cnt is number of SW processed packets
OIF: e1/22 TR(e1/32), TR is trunk, e1/32 primary
vlan 100, 1 caches. use 1 VIDX
1 (* 230.1.1.10) cnt=673
```

22 Displaying MCT information

```
OIF: TR(e5/4) tag TR(e5/5)
age=1s up-time=2005s, change=4s vidx=8187 (ref-cnt=1)
```

The following command displays status about the IGMP router port.

```
Brocade(config)#show ip multicast cluster vlan 100
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260
VL100: cfg V3, vlan cfg passive, 1 grp, 2 (SG) cache, rtr ports,
router ports: e5/9(260) 100.100.100.1 (local:1, mct peer:0),
e5/4 has 1 groups,
This interface is non-Querier (passive)
default V3 trunk
(local:1, mct peer:0)
```

Syntax: `show ip multicast cluster {group | mcache | vlan <vlan-id>}`

Use the `show ip multicast cluster pimsm-snooping` command to display detailed information about multicast snooping activity.

```
Brocade(config)#show ip multicast cluster pimsm-snooping
Example: Port: 7/3 (age, port type, ref_count, owner flag, pruned flag)
source: 7/3 has 1 src: 11.0.0.5(age, ref_count, owner flag, pruned flag)
owner flag: 0x0: local, 0x1 remote cep, 0x2 remote ccep

vlan 100, has 1 caches.
1 (* 224.10.10.10) has 2 pim join ports out of 2 OIF
7/3 (20, ICL, 1, 0x0, 0), 7/5 (20, CCEP, 1, 0x0, 0),
7/3 has 4 src: 11.0.0.5(20, 1, 0x0, 0), 11.0.0.4(20, 1, 0x0, 0), 11.0.0.3(20, 1,
0x0, 0), ...cut
7/5 has 4 src: 11.0.0.5(20, 1, 0x0, 0), 11.0.0.4(20, 1, 0x0, 0), 11.0.0.3(20, 1,
0x0, 0), ...cut
```

Syntax: `show ip multicast cluster pimsm-snooping [group | vlan]`

The following `show ipv6 mld` commands have output that is similar to the corresponding `show ip multicast` commands.

MLD multicast group commands:

```
Brocade(config)#show ipv6 mld cluster group
```

MLD multicast mcache output:

```
Brocade(config)#show ipv6 mld mcache
```

MLD router port output:

```
Brocade(config)#show ipv6 mld cluster vlan 100
```

Syntax: `show ipv6 mld cluster {group | mcache | vlan <vlan-id>}`

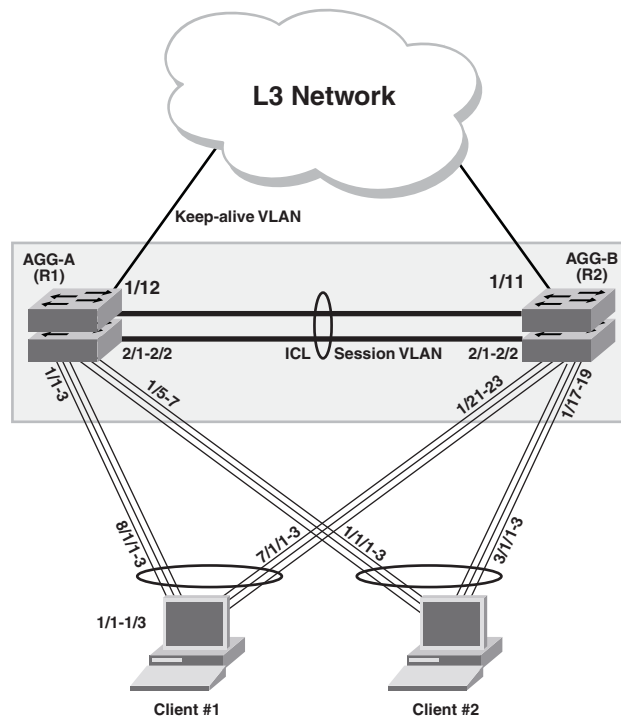
MCT configuration examples

The examples in this section show the topology and configuration for a single-level MCT deployment, two-level MCT deployment, VRRP/VRRP-E, and multicast snooping.

Single-level MCT example

[Table 104](#) shows an example single-level MCT configuration. The associated configuration follows.

FIGURE 104 Single level MCT configuration



Client 1 - Configuration

This section presents the configuration for client 1 in [Table 104](#).

```
!
vlan 1905 by port
  tagged ethe 5/1/47 to 5/1/48 ethe 7/1/1 to 7/1/3 ethe 8/1/1 to 8/1/3 ethe 8/1/45
  spanning-tree
!
!
interface ethernet 7/1/1
  link-aggregate configure timeout short
  link-aggregate configure key 10011
  link-aggregate active
!
interface ethernet 7/1/2
  link-aggregate configure key 10011
```

```

link-aggregate configure timeout short
link-aggregate active
!
interface ethernet 7/1/3
link-aggregate configure key 10011
link-aggregate configure timeout short
link-aggregate active
!
interface ethernet 8/1/1
link-aggregate configure key 10011
link-aggregate configure timeout short
link-aggregate active
!
interface ethernet 8/1/2
link-aggregate configure key 10011
link-aggregate configure timeout short
link-aggregate active
!
interface ethernet 8/1/3
link-aggregate configure key 10011
link-aggregate configure timeout short
link-aggregate active
!

```

Client 2- Configuration

This section presents the configuration for client 2 in [Table 104](#).

```

!
vlan 1905 name MAC-scaling-vlan by port
tagged ethe 1/1/1 to 1/1/3 ethe 1/1/45 ethe 2/1/47 to 2/1/48 ethe 3/1/1 to 3/1/3
spanning-tree
!
!
interface ethernet 1/1/1
link-aggregate configure timeout short
link-aggregate configure key 20011
link-aggregate active
!
interface ethernet 1/1/2
link-aggregate configure key 20011
link-aggregate configure timeout short
link-aggregate active
!
interface ethernet 1/1/3
link-aggregate configure key 20011
link-aggregate configure timeout short
link-aggregate active
!
interface ethernet 3/1/1
link-aggregate configure key 20011
link-aggregate configure timeout short
link-aggregate active
!
interface ethernet 3/1/2
link-aggregate configure key 20011
link-aggregate configure timeout short
link-aggregate active
!

```

```

interface ethernet 3/1/3
  link-aggregate configure key 20011
  link-aggregate configure timeout short
  link-aggregate active
!

```

AGG-A (R1) - Configuration

This section presents the configuration for the AGG-A(R1) cluster device in [Table 104](#).

```

trunk ethe 2/1 to 2/2
!
vlan 2 name session-vlan by port
  tagged ethe 2/1 to 2/2
  router-interface ve 2
!
vlan 3 name keep-alive-vlan by port
  tagged ethe 1/12
  router-interface ve 3
!
!
vlan 1905 name MAC-scaling-vlan by port
  tagged ethe 1/1 to 1/3 ethe 1/5 to 1/7 ethe 1/15 to 1/16 ethe 2/1 to 2/2 ethe 3/1
!
hostname R1
!
interface ve 2
  ip address 21.1.1.1 255.255.255.0
!
interface ve 3
  ip address 31.1.1.1 255.255.255.0
!
!
cluster MCT1 1
  rbridge-id 1
  session-vlan 2
  keep-alive-vlan 3
  icl BH1 ethernet 2/1
  peer 21.1.1.2 rbridge-id 2 icl BH1
  deploy
    client client-1
      rbridge-id 1901
      client-interface link-aggregation ethe 1/1 to 1/3
      client-interface link-aggregation passive
      client-interface link-aggregation timeout-short
    deploy
  client client-2
    rbridge-id 1902
    client-interface link-aggregation ethe 1/5 to 1/7
    client-interface link-aggregation passive
    client-interface link-aggregation timeout-short
  deploy
!

```

AGG-B(R2) - Configuration

This section presents the configuration for the AGG-B(R2) cluster device in [Table 104](#).

```

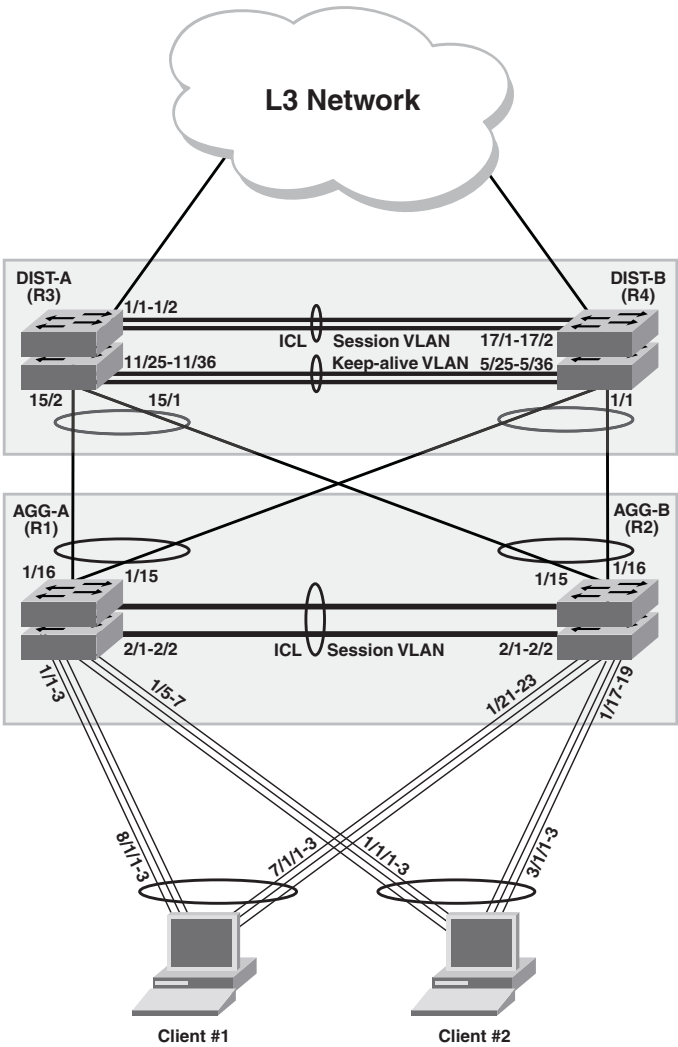
trunk ethe 2/1 to 2/2
!
vlan 2 name session-vlan by port
    tagged ethe 2/1 to 2/2
    router-interface ve 2
!
vlan 3 by port
    tagged ethe 1/11
    router-interface ve 3
!
!
vlan 1905 name MAC-scaling-vlan by port
    tagged ethe 1/15 to 1/19 ethe 1/21 to 1/23 ethe 2/1 to 2/2 ethe 2/5 ethe 3/1
!
hostname R2
!
interface ve 2
    ip address 21.1.1.2 255.255.255.0
!
interface ve 3
    ip address 31.1.1.2 255.255.255.0
!
interface ve 4
    ip address 101.1.1.2 255.255.255.0
!
cluster MCT1 1
    rbridge-id 2
    session-vlan 2
    keep-alive-vlan 3
    icl BH1 ethernet 2/1
    peer 21.1.1.1 rbridge-id 1 icl BH1
    deploy
        client client-1
            rbridge-id 1901
            client-interface link-aggregation ethe 1/21 to 1/23
            client-interface link-aggregation passive
            client-interface link-aggregation timeout-short
        deploy
    client client-2
        rbridge-id 1902
        client-interface link-aggregation ethe 1/17 to 1/19
        client-interface link-aggregation passive
        client-interface link-aggregation timeout-short
    deploy
!

```

Two-level MCT example

Table 105 shows an example two-level MCT configuration. The associated configuration follows.

FIGURE 105 Two-level MCT configuration



The client configuration is the same as in the single-level example (refer to “Single-level MCT example” on page 867).

AGG-A (R1) - Configuration

This example presents the configuration for the AGG-A(R1) cluster device in [Table 105](#).

```

!
trunk ethe 1/15 to 1/16
trunk ethe 2/1 to 2/2
!
vlan 2 name session-vlan by port
    tagged ethe 2/1 to 2/2
    router-interface ve 2
!
vlan 3 name keep-alive-vlan by port
    tagged ethe 1/12
    router-interface ve 3
!
!
vlan 1905 name MAC-scaling-vlan by port
    tagged ethe 1/1 to 1/3 ethe 1/5 to 1/7 ethe 1/15 to 1/16 ethe 2/1 to 2/2 ethe 3/1
!
hostname R1
!
interface ve 2
    ip address 21.1.1.1 255.255.255.0
!
interface ve 3
    ip address 31.1.1.1 255.255.255.0
!
!
cluster MCT1 1
    rbridge-id 1
    session-vlan 2
    keep-alive-vlan 3
    icl BH1 ethernet 2/1
    peer 21.1.1.2 rbridge-id 2 icl BH1
    deploy
client MCT2
    rbridge-id 5
    client-interface ethernet 1/15
    deploy
    client client-1
    rbridge-id 1901
    client-interface link-aggregation ethe 1/1 to 1/3
    client-interface link-aggregation passive
    client-interface link-aggregation timeout-short
    deploy
    client client-2
    rbridge-id 1902
    client-interface link-aggregation ethe 1/5 to 1/7
    client-interface link-aggregation passive
    client-interface link-aggregation timeout-short
    deploy
!

```

AGG-B (R2) - Configuration

This example presents the configuration for the AGG-B(R2) cluster device in [Table 105](#).

```

!
trunk ethe 1/15 to 1/16
trunk ethe 2/1 to 2/2
!
vlan 2 name session-vlan by port
    tagged ethe 2/1 to 2/2
    router-interface ve 2
!
vlan 3 by port
    tagged ethe 1/11
    router-interface ve 3
!
!
vlan 1905 name MAC-scaling-vlan by port
    tagged ethe 1/15 to 1/19 ethe 1/21 to 1/23 ethe 2/1 to 2/2 ethe 2/5 ethe 3/1
!
hostname R2
!
interface ve 2
    ip address 21.1.1.2 255.255.255.0
!
interface ve 3
    ip address 31.1.1.2 255.255.255.0
!
interface ve 4
    ip address 101.1.1.2 255.255.255.0
!
cluster MCT1 1
    rbridge-id 2
    session-vlan 2
    keep-alive-vlan 3
    icl BH1 ethernet 2/1
    peer 21.1.1.1 rbridge-id 1 icl BH1
    deploy
client MCT2
    rbridge-id 5
    client-interface ethernet 1/15
    deploy
    client client-1
    rbridge-id 1901
    client-interface link-aggregation ethe 1/21 to 1/23
    client-interface link-aggregation passive
    client-interface link-aggregation timeout-short
    deploy
    client client-2
    rbridge-id 1902
    client-interface link-aggregation ethe 1/17 to 1/19
    client-interface link-aggregation passive
    client-interface link-aggregation timeout-short
    deploy
!

```

DIST-A (R3) - Configuration

This example presents the configuration for the DIST-A(R3) cluster device in [Table 105](#).

```

!
trunk ethe 1/1 to 1/2
trunk ethe 15/1 to 15/2
!
vlan 5 name session-vlan by port
    tagged ethe 1/1 to 1/2
    router-interface ve 5
!
vlan 6 name keep-alive-vlan by port
    tagged ethe 11/25 to 11/36
    router-interface ve 6
    spanning-tree
!
!
vlan 1905 name MAC-scaling-vlan by port
    tagged ethe 1/1 to 1/2 ethe 3/24 ethe 15/1 to 15/2
!
hostname R3

hitless-failover enable
!

interface ethernet 11/25
    link-aggregate configure key 10001
    link-aggregate active
!
.....
!
interface ethernet 11/36
    link-aggregate configure key 10001
    link-aggregate active
!
interface ve 5
    ip address 51.1.1.1 255.255.255.252
!
interface ve 6
    ip address 61.1.1.1 255.255.255.248
!
cluster MCT2 2
    rbridge-id 3
    session-vlan 5
    keep-alive-vlan 6
    icl BH3 ethernet 1/1
    peer 51.1.1.2 rbridge-id 4 icl BH3
    deploy
    client MCT1
        rbridge-id 5
        client-interface ethernet 15/1
        deploy

```


DIST-B (R4) - Configuration

This example presents the configuration for the DIST-B(R4) cluster device in [Table 105](#).

```

trunk ethe 1/1 to 1/2
trunk ethe 17/1 to 17/2
!
vlan 5 name session-vlan by port
    tagged ethe 17/1 to 17/2
    router-interface ve 5
!
vlan 6 name keep-alive-vlan by port
    tagged ethe 5/25 to 5/36
    router-interface ve 6
    spanning-tree
!
vlan 1905 name MAC-scaling-vlan by port
    tagged ethe 1/1 to 1/2 ethe 17/1 to 17/2
!
hostname R4

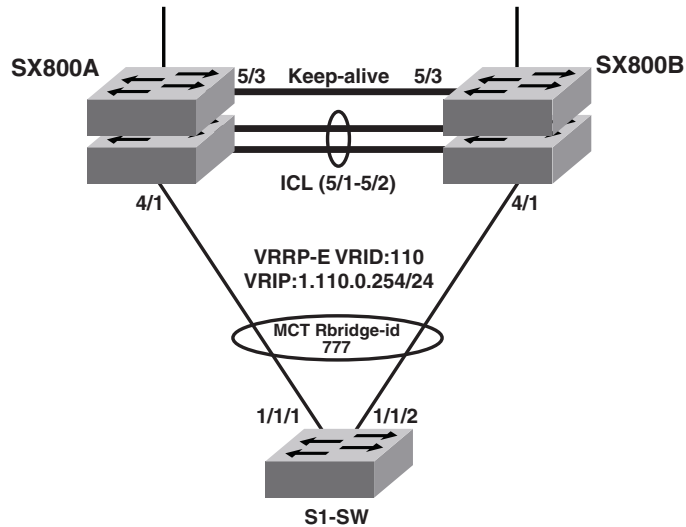
hitless-failover enable
!
interface ethernet 5/25
    link-aggregate configure key 10001
    link-aggregate active
!
.....
!
interface ethernet 5/36
    link-aggregate configure key 10001
    link-aggregate active
!
interface ve 5
    ip address 51.1.1.2 255.255.255.252
!
interface ve 6
    ip address 61.1.1.2 255.255.255.248
!
cluster MCT2 2
    rbridge-id 4
    session-vlan 5
    keep-alive-vlan 6
    icl BH3 ethernet 17/1
    peer 51.1.1.1 rbridge-id 3 icl BH3
    deploy
    client MCT1
        rbridge-id 5
        client-interface ethernet 1/1
        deploy

```

MCT configuration with VRRP-E example

Figure 106 shows a sample MCT configuration with VRRP-E. The associated configuration follows. The configuration for VRRP is similar.

FIGURE 106 Sample MCT configuration with VRRP-E



SX800A - MCT configuration

This example presents the MCT configuration for the SX800A cluster device in Table 105.

```
!
trunk ethe 5/1 to 5/2
  port-name "ICL-To_SX800B_eth5/1" ethernet 5/1
  port-name "ICL-To_SX800B_eth5/2" ethernet 5/2
!
!
vlan 110 name VRRP-E by port
  tagged ethe 4/1 ethe 5/1 to 5/2
  router-interface ve 110
!
vlan 1000 name ICL-Session-VLAN by port
  tagged ethe 5/1 to 5/2
  router-interface ve 1000
!
vlan 1001 name MCT-Keep-Alive by port
  tagged ethe 5/3
!
interface ve 1000
  ip address 1.0.0.254 255.255.255.252
!
cluster FI-MCT 1750
  rbridge-id 801
  session-vlan 1000
  keep-alive-vlan 1001
  icl FI_SX-MCT ethernet 5/1
```

```

peer 1.0.0.253 rbridge-id 800 icl FI_SX-MCT
deploy
client S1-SW
  rbridge-id 777
  client-interface ethe 4/1
deploy
!
```

SX800A - VRRP-E configuration

This example presents the VRRP-E configuration for the SX800A cluster device in [Table 105](#).

```

!
router vrrp-extended
!
interface ve 110
  port-name S1-SW
  ip address 1.110.0.253 255.255.255.0
  ip vrrp-extended vrid 110
  backup
  ip-address 1.110.0.254
  short-path-forwarding
  enable
!
```

SX800B- MCT configuration

This example presents the MCT configuration for the SX800B cluster device in [Table 105](#).

```

!
trunk ethe 5/1 to 5/2
  port-name "ICL-To_SX800A_eth5/1" ethernet 5/1
  port-name "ICL-To_SX800A_eth5/2" ethernet 5/2
!
!
vlan 110 name VRRP-E by port
  tagged ethe 4/1 ethe 5/1 to 5/2
  router-interface ve 110
!
vlan 1000 name ICL-Session-VLAN by port
  tagged ethe 5/1 to 5/2
  router-interface ve 1000
!
vlan 1001 name MCT-Keep-Alive by port
  tagged ethe 5/3
!
interface ve 1000
  ip address 1.0.0.253 255.255.255.252
!
cluster FI-MCT 1750
  rbridge-id 800
  session-vlan 1000
  keep-alive-vlan 1001
  icl FI_SX-MCT ethernet 5/1
  peer 1.0.0.254 rbridge-id 801 icl FI_SX-MCT
  deploy
  client S1-SW
```

```

rbridge-id 777
client-interface ethe 4/1
deploy
!

```

SX800B - VRRP-E configuration

This example presents the VRRP-E configuration for the SX800B cluster device in [Table 105](#).

```

!
router vrrp-extended
!
interface ve 110
port-name S1-SW
ip address 1.110.0.252 255.255.255.0
ip vrrp-extended vrid 110
backup
ip-address 1.110.0.254
short-path-forwarding
enable
!

```

S1-SW configuration

This example presents the configuration for the S1-SW device in [Table 105](#).

```

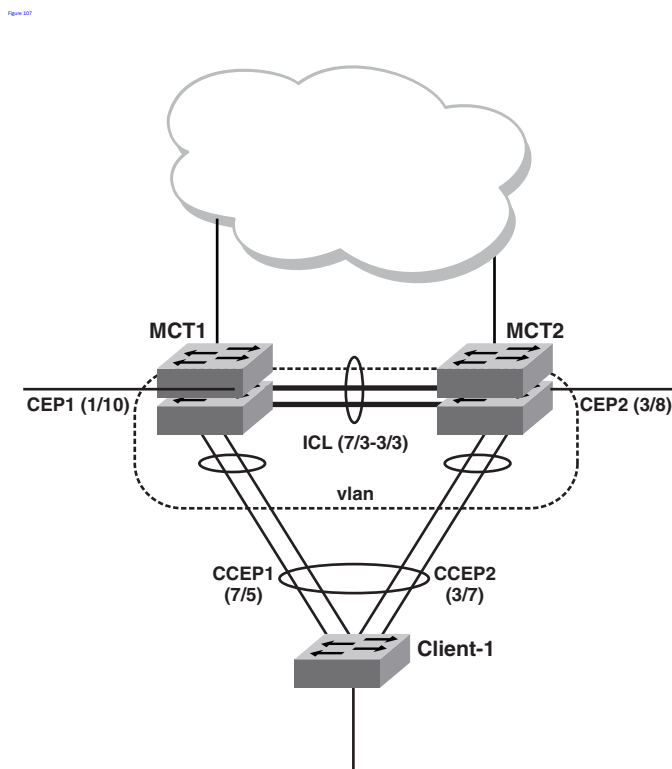
!
trunk ethe 1/1/1 to 1/1/2
!
Vlan 110 by port
tagged ethe 1/1/1 to 1/1/2
router-interface ve 110
!
interface ve 110
ip address 1.110.0.1 255.255.255.0
!

```

Multicast snooping configuration example

Figure 107 shows an example multicast snooping configuration. Sample configurations follow.

FIGURE 107 Multicast snooping over MCT



The following example shows the configuration for multicast snooping for the MCT1 cluster device in Figure 107.

```
vlan 100 by port
tagged ethe 7/3
untagged ethe 7/5 ethe 7/6
multicast passive
multicast pimsm-snooping
!
vlan 3000 name session by port
tagged ethe 7/3
router-interface ve 3000

vlan 3001 name keep-alive-vlan
tagged eth 7/4

interface ve 3000
ip address 1.1.1.2 255.255.255.0
!
cluster SX 3000
rbridge-id 2
session-vlan 3000
keep-alive-vlan 3001
```

22 MCT configuration examples

```
icl SX-MCT ethernet 7/3
peer 1.1.1.3 rbridge-id 3 icl SX-MCT
deploy
client client-1
  rbridge-id 100
  client-interface ethernet 7/5
  deploy
!
```

The following example shows the configuration for multicast snooping for the MCT2 cluster device in [Figure 107](#).

```
!
vlan 100 by port
  tagged ethe 3/3
  untagged ethe 3/7 ethe 3/8
  multicast passive
  multicast pimsm-snooping
!
vlan 3000 name session by port
  tagged ethe 3/3
  router-interface ve 3000

vlan 3001 name keep-alive-vlan
  tagged eth 3/4

interface ve 3000
  ip address 1.1.1.3 255.255.255.0
!
cluster SX 3000
  rbridge-id 3
  session-vlan 3000
  keep-alive-vlan 3001
  icl SX-MCT ethernet 3/3
  peer 1.1.1.2 rbridge-id 2 icl SX-MCT
  deploy
  client client-1
    rbridge-id 100
    client-interface ethernet 3/7 to 3/8
    deploy
  !
```

The following example shows the global configuration for multicast snooping for the MCT1 cluster device in [Figure 107](#).

```
vlan 100 by port
  tagged ethe 7/3
  tagged ethe 7/5 ethe 7/6
!

vlan 1000 by port
  tagged ethe 7/3
  untagged ethe 7/5 ethe 7/6
!

vlan 3000 name session by port
  tagged ethe 7/3
  router-interface ve 3000
```

```

vlan 3001 name keep-alive-vlan
tagged eth 7/4

ip multicast active

interface ve 3000
ip address 1.1.1.2 255.255.255.0
!
cluster SX 3000
rbridge-id 2
session-vlan 3000
keep-alive-vlan 3001
icl SX-MCT ethernet 7/3
peer 1.1.1.3 rbridge-id 3 icl SX-MCT
deploy
client client-1
rbridge-id 100
client-interface ethernet 7/5
deploy
!
```

The following example shows the global configuration for multicast snooping for the MCT2 cluster device in [Figure 107](#).

```

!
vlan 100 by port
tagged ethe 3/3
tagged ethe 3/7 ethe 3/8
!

vlan 1000 by port
tagged ethe 3/3
tagged ethe 3/7 ethe 3/8
!

vlan 3000 name session by port
tagged ethe 3/3
router-interface ve 3000

vlan 3001 name keep-alive-vlan
tagged eth 3/4

ip multicast passive

interface ve 3000
ip address 1.1.1.3 255.255.255.0
!
cluster SX 3000
rbridge-id 3
session-vlan 3000
keep-alive-vlan 3001
icl SX-MCT ethernet 3/3
peer 1.1.1.2 rbridge-id 2 icl SX-MCT
deploy
client client-1
rbridge-id 100
client-interface ethernet 3/7 to 3/8
deploy
```


GVRP

[Table 148](#) lists the individual Brocade FastIron switches and the GARP VLAN Registration Protocol (GVRP) features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 148 Supported GVRP features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
GVRP	Yes	Yes	Yes	Yes	Yes
Configurable GVRP base VLAN ID	Yes	Yes	Yes	Yes	Yes
Leaveall timer	Yes	Yes	Yes	Yes	Yes
Ability to disable VLAN advertising	Yes	Yes	Yes	Yes	Yes
Ability to disable VLAN learning	Yes	Yes	Yes	Yes	Yes
GVRP timers	Yes	Yes	Yes	Yes	Yes
Conversion of a GVRP VLAN to a statically-configured VLAN	Yes	Yes	Yes	Yes	Yes

GVRP overview

GARP VLAN Registration Protocol (GVRP) is a Generic Attribute Registration Protocol (GARP) application that provides VLAN registration service by means of dynamic configuration (registration) and distribution of VLAN membership information.

A Brocade device enabled for GVRP can do the following:

- Learn about VLANs from other Brocade devices and configure those VLANs on the ports that learn about the VLANs. The device listens for GVRP Protocol Data Units (PDUs) from other devices, and implements the VLAN configuration information in the PDUs.
- Advertise VLANs configured on the device to other Brocade devices. The device sends GVRP PDUs advertising its VLANs to other devices. GVRP advertises statically configured VLANs and VLANs learned from other devices through GVRP.

GVRP enables a Brocade device to dynamically create 802.1Q-compliant VLANs on links with other devices that are running GVRP. GVRP reduces the chances for errors in VLAN configuration by automatically providing VLAN ID consistency across the network. You can use GVRP to propagate VLANs to other GVRP-aware devices automatically, without the need to manually configure the VLANs on each device. In addition, if the VLAN configuration on a device changes, GVRP automatically changes the VLAN configurations of the affected devices.

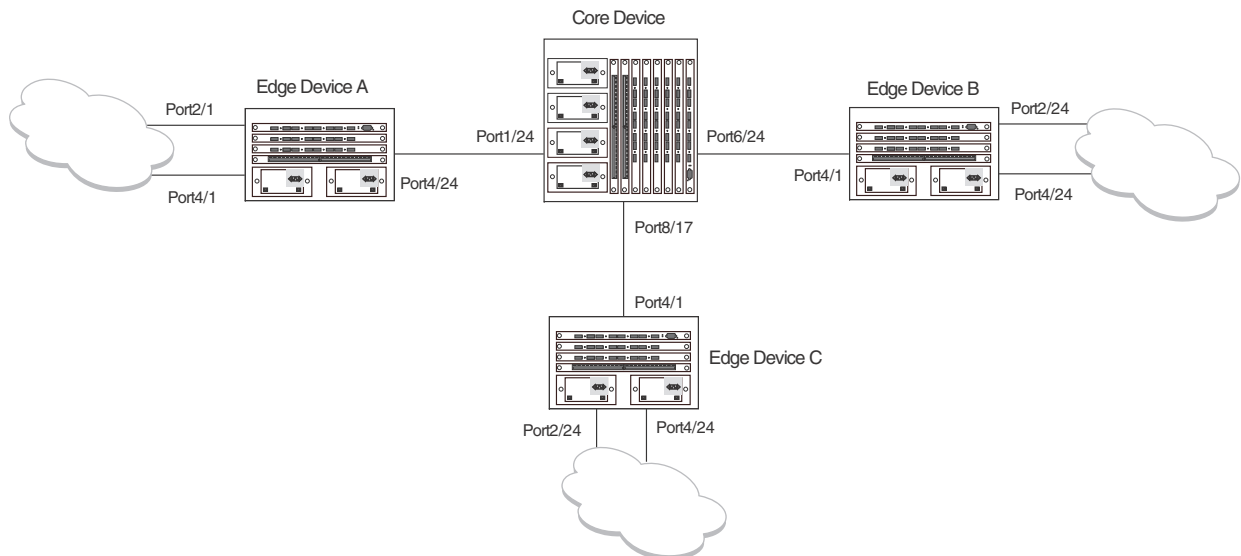
The Brocade implementation of GARP and GVRP is based on the following standards:

- ANSI/IEEE standard 802.1D, 1998 edition
- IEEE standard 802.1Q, 1998 edition; approved December 8, 1998
- IEEE draft P802.1w/D10, March 26, 2001
- IEEE draft P802.1u/D9, November 23, 2000
- IEEE draft P802.1t/D10, November 20, 2000

GVRP application examples

Figure 108 shows an example of a network that uses GVRP. This section describes various ways you can use GVRP in a network such as this one. “GVRP CLI examples” on page 902 lists the CLI commands to implement the applications of GVRP described in this section.

FIGURE 108 Example of GVRP



In this example, a core device is attached to three edge devices. Each of the edge devices is attached to other edge devices or host stations (represented by the clouds).

The effects of GVRP in this network depend on which devices the feature is enabled on, and whether both learning and advertising are enabled. In this type of network (a core device and edge devices), you can have the following four combinations:

- Dynamic core and fixed edge
- Dynamic core and dynamic edge
- Fixed core and dynamic edge
- Fixed core and fixed edge

Dynamic core and fixed edge

In this configuration, all ports on the core device are enabled to learn and advertise VLAN information. The edge devices are configured to advertise their VLAN configurations on the ports connected to the core device. GVRP learning is disabled on the edge devices.

Core device	Edge device A	Edge device B	Edge device C
<ul style="list-style-type: none"> GVRP is enabled on all ports. Both learning and advertising are enabled. <p>NOTE: Since learning is disabled on all the edge devices, advertising on the core device has no effect in this configuration.</p>	<ul style="list-style-type: none"> GVRP is enabled on port 4/24. Learning is disabled. VLAN 20 Port 2/1 (untagged) Port 4/24 (tagged) VLAN 40 Port 4/1 (untagged) Port 4/24 (tagged) 	<ul style="list-style-type: none"> GVRP is enabled on port 4/1. Learning is disabled. VLAN 20 Port 2/24 (untagged) Port 4/1 (tagged) VLAN 30 Port 4/24 (untagged) Port 4/1 (tagged) 	<ul style="list-style-type: none"> GVRP is enabled on port 4/1. Learning is disabled. VLAN 30 Port 2/24 (untagged) Port 4/1 (tagged) VLAN 40 Port 4/24 (untagged) Port 4/1 (tagged)

In this configuration, the edge devices are statically (manually) configured with VLAN information. The core device dynamically configures itself to be a member of each of the edge device VLANs. The operation of GVRP on the core device results in the following VLAN configuration on the device:

- VLAN 20
 - 1/24 (tagged)
 - 6/24 (tagged)
- VLAN 30
 - 6/24 (tagged)
 - 8/17 (tagged)
- VLAN 40
 - 1/24 (tagged)
 - 8/17 (tagged)

VLAN 20 traffic can now travel through the core between edge devices A and B. Likewise, VLAN 30 traffic can travel between B and C and VLAN 40 traffic can travel between A and C. If an edge device is moved to a different core port or the VLAN configuration of an edge device is changed, the core device automatically reconfigures itself to accommodate the change.

Notice that each of the ports in the dynamically created VLANs is tagged. All GVRP VLAN ports configured by GVRP are tagged, to ensure that the port can be configured for additional VLANs.

NOTE

This example assumes that the core device has no static VLANs configured. However, you can have static VLANs on a device that is running GVRP. GVRP can dynamically add other ports to the statically configured VLANs but cannot delete statically configured ports from the VLANs.

Dynamic core and dynamic edge

GVRP is enabled on the core device and on the edge devices. This type of configuration is useful if the devices in the edge clouds are running GVRP and advertise their VLANs to the edge devices. The edge devices learn the VLANs and also advertise them to the core. In this configuration, you do not need to statically configure the VLANs on the edge or core devices, although you can have statically configured VLANs on the devices. The devices learn the VLANs from the devices in the edge clouds.

Fixed core and dynamic edge

GVRP learning is enabled on the edge devices. The VLANs on the core device are statically configured, and the core device is enabled to advertise its VLANs but not to learn VLANs. The edge devices learn the VLANs from the core.

Fixed core and fixed edge

The VLANs are statically configured on the core and edge devices. On each edge device, VLAN advertising is enabled but learning is disabled. GVRP is not enabled on the core device. This configuration enables the devices in the edge clouds to learn the VLANs configured on the edge devices.

VLAN names created by GVRP

The **show vlans** command lists VLANs created by GVRP as “GVRP_VLAN_<vlan-id>”. VLAN names for statically configured VLANs are not affected. To distinguish between statically-configured VLANs that you add to the device and VLANs that you convert from GVRP-configured VLANs into statically-configured VLANs, the **show vlans** command displays a converted VLAN name as “STATIC_VLAN_<vlan-id>”.

Configuration notes for GVRP

- If you disable GVRP, all GVRP configuration information is lost if you save the configuration change (**write memory** command) and then reload the software. However, if you reload the software without first saving the configuration change, the GVRP configuration is restored following a software reload.
- The maximum number of VLANs supported on a device enabled for GVRP is the same as the maximum number on a device that is not enabled for GVRP.
 - To display the maximum number of VLANs allowed on your device, enter the **show default values** command. See the “vlan” row in the System Parameters section. Make sure you allow for the default VLAN (1), the GVRP base VLAN (4093), and the Single STP VLAN (4094). These VLANs are maintained as “Registration Forbidden” in the GVRP database. Registration Forbidden VLANs cannot be advertised or learned by GVRP.
 - To increase the maximum number of VLANs supported on the device, enter the **system-max vlan <num>** command at the global CONFIG level of the CLI, then save the configuration and reload the software. The maximum number you can specify is listed in the Maximum column of the **show default values** display.

- The default VLAN (VLAN 1) is not advertised by the Brocade implementation of GVRP. The default VLAN contains all ports that are not members of statically configured VLANs or VLANs enabled for GVRP.

NOTE

The default VLAN has ID 1 by default. You can change the VLAN ID of the default VLAN, but only before GVRP is enabled. You cannot change the ID of the default VLAN after GVRP is enabled.

- Single STP must be enabled on the device. Brocade implementation of GVRP requires Single STP. If you do not have any statically configured VLANs on the device, you can enable Single STP as follows.

```
Brocade(config)#vlan 1
Brocade(config-vlan-1)#exit
Brocade(config)#span
Brocade(config)#span single
```

These commands enable configuration of the default VLAN (VLAN 1), which contains all the device ports, and enable STP and Single STP.

- All VLANs that are learned dynamically through GVRP are added to the single spanning tree.
- All ports that are enabled for GVRP become tagged members of the GVRP base VLAN (4093). If you need to use this VLAN ID for another VLAN, you can change the GVRP VLAN ID. Refer to [“Changing the GVRP base VLAN ID”](#) on page 888. The software adds the GVRP base VLAN to the single spanning tree.
- All VLAN ports added by GVRP are tagged.
- GVRP is supported only for tagged ports or for untagged ports that are members of the default VLAN. GVRP is not supported for ports that are untagged and are members of a VLAN other than the default VLAN.
- To configure GVRP on a trunk group, enable the protocol on the primary port in the trunk group. The GVRP configuration of the primary port is automatically applied to the other ports in the trunk group.
- You can use GVRP on a device even if the device has statically configured VLANs. GVRP does not remove any ports from the statically configured VLANs, although GVRP can add ports to the VLANs. GVRP advertises the statically configured VLANs. Ports added by GVRP do not appear in the running-config and will not appear in the startup-config file when save the configuration. You can manually add a port to make the port a permanent member of the VLAN. After you manually add the port, the port will appear in the running-config and be saved to the startup-config file when you save the configuration.
- VLANs created by GVRP do not support virtual routing interfaces or protocol-based VLANs. virtual routing interfaces and protocol-based VLANs are still supported on statically configured VLANs even if GVRP adds ports to those VLANs.
- You cannot manually configure any parameters on a VLAN that is created by GVRP. For example, you cannot change STP parameters for the VLAN.
- The GVRP timers (Join, Leave, and Leaveall) must be set to the same values on all the devices that are exchanging information using GVRP.
- If the network has a large number of VLANs, the GVRP traffic can use a lot of CPU resources. If you notice high CPU utilization after enabling GVRP, set the GVRP timers to longer values. In particular, set the Leaveall timer to a longer value. Refer to [“Changing the GVRP timers”](#) on page 890.
- The feature is supported only on Ethernet ports.

NOTE

If you plan to change the GVRP base VLAN ID (4093) or the maximum configurable value for the Leaveall timer (300000 ms by default), you must do so before you enable GVRP.

GVRP configuration

To configure a device for GVRP, globally enable support for the feature, then enable the feature on specific ports. Optionally, you can disable VLAN learning or advertising on specific interfaces.

You can also change the protocol timers and the GVRP base VLAN ID.

Changing the GVRP base VLAN ID

By default, GVRP uses VLAN 4093 as a base VLAN for the protocol. All ports that are enabled for GVRP become tagged members of this VLAN. If you need to use VLAN ID 4093 for a statically configured VLAN, you can change the GVRP base VLAN ID.

NOTE

If you want to change the GVRP base VLAN ID, you must do so before enabling GVRP.

To change the GVRP base VLAN ID, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#gvrp-base-vlan-id 1001
```

This command changes the GVRP VLAN ID from 4093 to 1001.

Syntax: [no] **gvrp-base-vlan-id** <vlan-id>

The <vlan-id> parameter specifies the new VLAN ID. You can specify a VLAN ID from 2 – 4092 or 4095.

Increasing the maximum configurable value of the Leaveall timer

By default, the highest value you can specify for the Leaveall timer is 300000 ms. You can increase the maximum configurable value of the Leaveall timer to 1000000 ms.

NOTE

You must enter this command before enabling GVRP. Once GVRP is enabled, you cannot change the maximum Leaveall timer value.

NOTE

This command does not change the default value of the Leaveall timer itself. The command only changes the maximum value to which you can set the Leaveall timer.

To increase the maximum value you can specify for the Leaveall timer, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#gvrp-max-leaveall-timer 1000000
```

Syntax: [no] **gvrp-max-leaveall-timer** <ms>

The <ms> parameter specifies the maximum number of ms to which you can set the Leaveall timer. You can specify from 300000 – 1000000 (one million) ms. The value must be a multiple of 100 ms. The default is 300000 ms.

Enabling GVRP

To enable GVRP, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable all
```

The first command globally enables support for the feature and changes the CLI to the GVRP configuration level. The second command enables GVRP on all ports on the device.

The following command enables GVRP on ports 1/24, 2/24, and 4/17.

```
Brocade(config-gvrp)#enable ethernet 1/24 ethernet 2/24 ethernet 4/17
```

Syntax: [no] **gvrp-enable**

Syntax: [no] **enable all** | **ethernet** <port> [**ethernet** <port> | **to** <port>]

The **all** keyword enables GVRP on all ports.

ethernet <port> specifies a port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To specify a list of ports, enter each port as **ethernet** <port> followed by a space. For example, **ethernet 1/24 ethernet 6/24 ethernet 8/17**

To specify a range of ports, enter the first port in the range as **ethernet** <port> followed by the last port in the range. For example, **ethernet 1/1 to 1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 1/1 to 1/8 ethernet 1/24 ethernet 6/24 ethernet 8/17**.

Disabling VLAN advertising

To disable VLAN advertising on a port enabled for GVRP, enter a command such as the following at the GVRP configuration level.

```
Brocade(config-gvrp)#block-applicant ethernet 1/24 ethernet 6/24 ethernet 8/17
```

This command disables advertising of VLAN information on ports 1/24, 6/24, and 8/17.

Syntax: [no] **block-applicant all** | **ethernet** <port> [**ethernet** <port> | **to** <port>]

NOTE

Leaveall messages are still sent on the GVRP ports.

The **all** keyword disables VLAN advertising on all ports enabled for GVRP.

ethernet <port> specifies a port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To specify a list of ports, enter each port as **ethernet** <port> followed by a space. For example, **ethernet 1/24 ethernet 6/24 ethernet 8/17**

To specify a range of ports, enter the first port in the range as **ethernet** <port> followed by the last port in the range. For example, **ethernet 1/1 to 1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 1/1 to 1/8 ethernet 1/24 ethernet 6/24 ethernet 8/17**.

Disabling VLAN learning

To disable VLAN learning on a port enabled for GVRP, enter a command such as the following at the GVRP configuration level.

```
Brocade(config-gvrp)#block-learning ethernet 6/24
```

This command disables learning of VLAN information on port 6/24.

NOTE

The port still advertises VLAN information unless you also disable VLAN advertising.

Syntax: [no] **block-learning** all | **ethernet** <port> [**ethernet** <port> | **to** <port>]

The **all** keyword disables VLAN learning on all ports enabled for GVRP.

ethernet <port> specifies a port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To specify a list of ports, enter each port as **ethernet** <port> followed by a space. For example, **ethernet 1/24 ethernet 6/24 ethernet 8/17**

To specify a range of ports, enter the first port in the range as **ethernet** <port> followed by the last port in the range. For example, **ethernet 1/1 to 1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 1/1 to 1/8 ethernet 1/24 ethernet 6/24 ethernet 8/17**.

Changing the GVRP timers

GVRP uses the following timers:

- **Join** – The maximum number of milliseconds (ms) a device GVRP interfaces wait before sending VLAN advertisements on the interfaces. The actual interval between Join messages is randomly calculated to a value between 0 and the maximum number of milliseconds specified for Join messages. You can set the Join timer to a value from 200 – one third the value of the Leave timer. The default is 200 ms.
- **Leave** – The number of ms a GVRP interface waits after receiving a Leave message on the port to remove the port from the VLAN indicated in the Leave message. If the port receives a Join message before the Leave timer expires, GVRP keeps the port in the VLAN. Otherwise, the port is removed from the VLAN. When a port receives a Leave message, the port GVRP state is changed to Leaving. Once the Leave timer expires, the port GVRP state changes to Empty. You can set the Leave timer to a value from three times the Join timer – one fifth the value of the Leaveall timer. The default is 600 ms.

NOTE

When all ports in a dynamically created VLAN (one learned through GVRP) leave the VLAN, the VLAN is immediately deleted from the device's VLAN database. However, this empty VLAN is still maintained in the GVRP database for an amount of time equal to the following.

$(\text{number-of-GVRP-enabled-up-ports}) * (2 * \text{join-timer})$

While the empty VLAN is in the GVRP database, the VLAN does not appear in the **show vlans** display but does still appear in the **show gvrp vlan all** display.

- **Leaveall** – The minimum interval at which GVRP sends Leaveall messages on all GVRP interfaces. Leaveall messages ensure that the GVRP VLAN membership information is current by aging out stale VLAN information and adding information for new VLAN memberships, if the information is missing. A Leaveall message instructs the port to change the GVRP state for all its VLANs to Leaving, and remove them unless a Join message is received before the Leave timer expires. By default, you can set the Leaveall timer to a value from five times the Leave timer – maximum value allowed by software (configurable from 300000 – 1000000 ms). The default is 10000.

NOTE

The actual interval is a random value between the Leaveall interval and $1.5 * \text{the Leaveall time}$ or the maximum Leaveall time, whichever is lower.

NOTE

You can increase the maximum configurable value of the Leaveall timer from 300000 ms up to 1000000 ms using the **gvrp-max-leaveall-timer** command. (Refer to [“Increasing the maximum configurable value of the Leaveall timer”](#) on page 888.)

Timer configuration requirements

- All timer values must be in multiples of 100 ms.
- The Leave timer must be $\geq 3 * \text{the Join timer}$.
- The Leaveall timer must be $\geq 5 * \text{the Leave timer}$.
- The GVRP timers must be set to the same values on all the devices that are exchanging information using GVRP.

Changing the Join, Leave, and Leaveall timers

The same CLI command controls changes to the Join, Leave, and Leaveall timers. To change values to the timers, enter a command such as the following.

```
Brocade(config-gvrp)#join-timer 1000 leave-timer 3000 leaveall-timer 15000
```

This command changes the Join timer to 1000 ms, the Leave timer to 3000 ms, and the Leaveall timer to 15000.

Syntax: [no] join-timer <ms> leave-timer <ms> leaveall-timer <ms>

NOTE

When you enter this command, all the running GVRP timers are canceled and restarted using the new times specified by the command.

Resetting the timers to their defaults

To reset the Join, Leave, and Leaveall timers to their default values, enter the following command.

```
Brocade(config-gvrp)#default-timers
```

Syntax: default-timers

This command resets the timers to the following values:

- Join – 200 ms
- Leave – 600 ms
- Leaveall – 10000 ms

Converting a VLAN created by GVRP into a statically-configured VLAN

You cannot configure VLAN parameters on VLANs created by GVRP. Moreover, VLANs and VLAN ports added by GVRP do not appear in the running-config and cannot be saved in the startup-config file.

To be able to configure and save VLANs or ports added by GVRP, you must convert the VLAN ports to statically-configured ports.

To convert a VLAN added by GVRP into a statically-configured VLAN, add the ports using commands such as the following.

```
Brocade(config)#vlan 22
Brocade(config-vlan-222)#tagged ethernet 1/1 to 1/8
```

These commands convert GVRP-created VLAN 22 containing ports 1/1 through 1/8 into statically-configured VLAN 22.

Syntax: [no] vlan <vlan-id>

Syntax: [no] tagged ethernet <port> [to <port> | ethernet <port>]

Use the same commands to statically add ports that GVRP added to a VLAN.

NOTE

You cannot add the VLAN ports as untagged ports.

NOTE

After you convert the VLAN, the VLAN name changes from “GVRP_VLAN_<vlan-id>” to “STATIC_VLAN_<vlan-id>”.

ethernet <port> specifies a port. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To specify a list of ports, enter each port as **ethernet** <port> followed by a space. For example, **ethernet 1/24 ethernet 6/24 ethernet 8/17**

To specify a range of ports, enter the first port in the range as **ethernet** <port> followed by the last port in the range. For example, **ethernet 1/1 to 1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 1/1 to 1/8 ethernet 1/24 ethernet 6/24 ethernet 8/17**.

Displaying GVRP information

You can display the following GVRP information:

- GVRP configuration information
- GVRP VLAN information
- GVRP statistics
- CPU utilization statistics
- GVRP diagnostic information

Displaying GVRP configuration information

To display GVRP configuration information, enter a command such as the following.

```
Brocade#show gvrp
GVRP is enabled on the system

GVRP BASE VLAN ID          : 4093
GVRP MAX Leaveall Timer    : 300000 ms

GVRP Join Timer            : 200 ms
GVRP Leave Timer           : 600 ms
GVRP Leave-all Timer      : 10000 ms

=====
Configuration that is being used:

block-learning ethe 1/3
block-applicant ethe 2/7 ethe 2/11
enable ethe 1/1 to 1/7 ethe 2/1 ethe 2/7 ethe 2/11

=====

Spanning Tree: SINGLE SPANNING TREE
Dropped Packets Count: 0

=====

Number of VLANs in the GVRP Database: 15
Maximum Number of VLANs that can be present: 4095

=====
```

Syntax: `show gvrp [ethernet <port>]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

This display shows the following information.

TABLE 149 CLI display of summary GVRP information

Field	Description
Protocol state	The state of GVRP. The display shows one of the following: <ul style="list-style-type: none"> • GVRP is disabled on the system • GVRP is enabled on the system
GVRP BASE VLAN ID	The ID of the base VLAN used by GVRP.
GVRP MAX Leaveall Timer	The maximum number of ms to which you can set the Leaveall timer. <p>NOTE: To change the maximum value, refer to “Increasing the maximum configurable value of the Leaveall timer” on page 888.</p>

TABLE 149 CLI display of summary GVRP information (Continued)

Field	Description
GVRP Join Timer	The value of the Join timer. NOTE: For descriptions of the Join, Leave, and Leaveall timers or to change the timers, refer to “Changing the GVRP timers” on page 890.
GVRP Leave Timer	The value of the Leave timer.
GVRP Leave-all Timer	The value of the Leaveall timer.
Configuration that is being used	The configuration commands used to enable GVRP on individual ports. If GVRP learning or advertising is disabled on a port, this information also is displayed.
Spanning Tree	The type of STP enabled on the device. NOTE: The current release supports GVRP only with Single STP.
Dropped Packets Count	The number of GVRP packets that the device has dropped. A GVRP packet can be dropped for either of the following reasons: <ul style="list-style-type: none"> GVRP packets are received on a port on which GVRP is not enabled. NOTE: If GVRP support is not globally enabled, the device does not drop the GVRP packets but instead forwards them at Layer 2. <ul style="list-style-type: none"> GVRP packets are received with an invalid GARP Protocol ID. The protocol ID must always be 0x0001.
Number of VLANs in the GVRP Database	The number of VLANs in the GVRP database. NOTE: This number includes the default VLAN (1), the GVRP base VLAN (4093), and the single STP VLAN (4094). These VLANs are not advertised by GVRP but are maintained as “Registration Forbidden”.
Maximum Number of VLANs that can be present	The maximum number of VLANs that can be configured on the device. This number includes statically configured VLANs, VLANs learned through GVRP, and VLANs 1, 4093, and 4094. To change the maximum number of VLANs the device can have, use the system-max vlan <num> command. Refer to “Displaying and modifying system parameter default settings” on page 578.

To display detailed GVRP information for an individual port, enter a command such as the following.

Displaying GVRP information

```
Brocade#show gvrp ethernet 2/1
Port 2/1 -
  GVRP Enabled    : YES
  GVRP Learning   : ALLOWED
  GVRP Applicant  : ALLOWED
  Port State      : UP
  Forwarding      : YES

VLAN Membership:      [VLAN-ID]      [MODE]
                      1              FORBIDDEN
                      2              FIXED
                      1001           NORMAL
                      1003           NORMAL
                      1004           NORMAL
                      1007           NORMAL
                      1009           NORMAL
                      1501           NORMAL
                      2507           NORMAL
                      4001           NORMAL
                      4093           FORBIDDEN
                      4094           FORBIDDEN
```

This display shows the following information.

TABLE 150 CLI display of detailed GVRP information for a port

Field	Description
Port number	The port for which information is being displayed.
GVRP Enabled	Whether GVRP is enabled on the port.
GVRP Learning	Whether the port can learn VLAN information from GVRP.
GVRP Applicant	Whether the port can advertise VLAN information into GVRP.
Port State	The port link state, which can be UP or DOWN.
Forwarding	Whether the port is in the GVRP Forwarding state: <ul style="list-style-type: none">• NO – The port is in the Blocking state.• YES – The port is in the Forwarding state.
VLAN Membership	The VLANs of which the port is a member. For each VLAN, the following information is shown: <ul style="list-style-type: none">• VLAN ID – The VLAN ID.• Mode – The type of VLAN, which can be one of the following:• FIXED – The port will always be a member of this VLAN and the VLAN will always be advertised on this port by GVRP. A port becomes FIXED when you configure the port as a tagged member of a statically configured VLAN.• FORBIDDEN – The VLAN is one of the special VLANs that is not advertised or learned by GVRP. In the current release, the following VLANs are forbidden: the default VLAN (1), the GVRP base VLAN (4093), or the Single STP VLAN (4094).• NORMAL – The port became a member of this VLAN after learning about the VLAN through GVRP. The port membership in the VLAN depends on GVRP. If the VLAN is removed from the ports that send GVRP advertisements to this device, then the port will stop being a member of the VLAN.

Displaying GVRP VLAN information

To display information about all the VLANs on the device, enter the following command.

```
Brocade#show gvrp vlan brief
```

Number of VLANs in the GVRP Database: 7

Maximum Number of VLANs that can be present: 4095

[VLAN-ID]	[MODE]	[VLAN-INDEX]
1	STATIC-DEFAULT	0
7	STATIC	2
11	STATIC	4
1001	DYNAMIC	7
1003	DYNAMIC	8
4093	STATIC-GVRP-BASE-VLAN	6
4094	STATIC-SINGLE-SPAN-VLAN	5

=====

Syntax: show gvrp vlan all | brief | <vlan-id>

This display shows the following information.

TABLE 151 CLI display of summary VLAN information for GVRP

Field	Description
Number of VLANs in the GVRP Database	The number of VLANs in the GVRP database. NOTE: This number includes the default VLAN (1), the GVRP base VLAN (4093), and the single STP VLAN (4094). These VLANs are not advertised by GVRP but are included in the total count.
Maximum Number of VLANs that can be present	The maximum number of VLANs that can be configured on the device. This number includes statically configured VLANs, VLANs learned through GVRP, and VLANs 1, 4093, and 4094. To change the maximum number of VLANs the device can have, use the system-max vlan <num> command. Refer to “Displaying and modifying system parameter default settings” on page 578.
VLAN-ID	The VLAN ID.
MODE	The type of VLAN, which can be one of the following: <ul style="list-style-type: none"> • STATIC – The VLAN is statically configured and cannot be removed by GVRP. This includes VLANs you have configured as well as the default VLAN (1), base GVRP VLAN (4093), and Single STP VLAN (4094). • DYNAMIC – The VLAN was learned through GVRP.
VLAN-INDEX	A number used as an index into the internal database.

To display detailed information for a specific VLAN, enter a command such as the following.

```
Brocade#show gvrp vlan 1001
```

```
VLAN-ID: 1001, VLAN-INDEX: 7, STATIC: NO, DEFAULT: NO, BASE-VLAN: NO
Timer to Delete Entry Running: NO
Legend: [S=Slot]
```

Forbidden Members: None

Fixed Members: None

Normal(Dynamic) Members: (S2) 1

This display shows the following information.

TABLE 152 CLI display of summary VLAN information for GVRP

Field	Description
VLAN-ID	The VLAN ID.
VLAN-INDEX	A number used as an index into the internal database.
STATIC	Whether the VLAN is a statically configured VLAN.
DEFAULT	Whether this is the default VLAN.
BASE-VLAN	Whether this is the base VLAN for GVRP.
Timer to Delete Entry Running	Whether all ports have left the VLAN and the timer to delete the VLAN itself is running. The timer is described in the note for the Leave timer in “Changing the GVRP timers” on page 890.
Legend	The meanings of the letter codes used in other parts of the display.
Forbidden Members	The ports that cannot become members of a VLAN advertised or learned by GVRP.
Fixed Members	The ports that are statically configured members of the VLAN. GVRP cannot remove these ports.
Normal(Dynamic) Members	The ports that were added by GVRP. These ports also can be removed by GVRP.
MODE	The type of VLAN, which can be one of the following: <ul style="list-style-type: none"> • STATIC – The VLAN is statically configured and cannot be removed by GVRP. This includes VLANs you have configured as well as the default VLAN (1), base GVRP VLAN (4093), and Single STP VLAN (4094). • DYNAMIC – The VLAN was learned through GVRP.

To display detailed information for all VLANs, enter the **show gvrp vlan all** command.

Displaying GVRP statistics

To display GVRP statistics for a port, enter a command such as the following.

```
Brocade#show gvrp statistics ethernet 2/1
PORT 2/1 Statistics:
  Leave All Received           : 147
  Join Empty Received         : 4193
  Join In Received            : 599
  Leave Empty Received        : 0
  Leave In Received           : 0
  Empty Received              : 588
  Leave All Transmitted       : 157
  Join Empty Transmitted      : 1794
  Join In Transmitted         : 598
  Leave Empty Transmitted     : 0
  Leave In Transmitted        : 0
  Empty Transmitted           : 1248
  Invalid Messages/Attributes Skipped : 0
  Failed Registrations        : 0
```

Syntax: **show gvrp statistics all | ethernet <port>**

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

This display shows the following information for the port.

TABLE 153 CLI display of GVRP statistics

Field	Description
Leave All Received	The number of Leaveall messages received.
Join Empty Received	The number of Join Empty messages received.
Join In Received	The number of Join In messages received.
Leave Empty Received	The number of Leave Empty messages received.
Leave In Received	The number of Leave In messages received.
Empty Received	The number of Empty messages received.
Leave All Transmitted	The number of Leaveall messages sent.
Join Empty Transmitted	The number of Join Empty messages sent.
Join In Transmitted	The number of Join In messages sent.
Leave Empty Transmitted	The number of Leave Empty messages sent.
Leave In Transmitted	The number of Leave In messages sent.
Empty Transmitted	The number of Empty messages sent.
Invalid Messages/Attributes Skipped	<p>The number of invalid messages or attributes received or skipped. This can occur in the following cases:</p> <ul style="list-style-type: none"> • The incoming GVRP PDU has an incorrect length. • "End of PDU" was reached before the complete attribute could be parsed. • The Attribute Type of the attribute that was being parsed was not the GVRP VID Attribute Type (0x01). • The attribute that was being parsed had an invalid attribute length. • The attribute that was being parsed had an invalid GARP event. • The attribute that was being parsed had an invalid VLAN ID. The valid range is 1 – 4095.
Failed Registrations	<p>The number of failed registrations that have occurred. A failed registration can occur for the following reasons:</p> <ul style="list-style-type: none"> • Join requests were received on a port that was blocked from learning dynamic VLANs (GVRP Blocking state). • An entry for a new GVRP VLAN could not be created in the GVRP database.

To display GVRP statistics for all ports, enter the **show gvrp statistics all** command.

Displaying CPU utilization statistics

You can display CPU utilization statistics for GVRP.

To display CPU utilization statistics for GVRP for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI.

```
Brocade#show process cpu
```

Process Name	5Sec(%)	1Min(%)	5Min(%)	15Min(%)	Runtime(ms)
ARP	0.01	0.03	0.09	0.22	9
BGP	0.00	0.00	0.00	0.00	0
GVRP	0.00	0.03	0.04	0.07	4
ICMP	0.00	0.00	0.00	0.00	0
IP	0.00	0.00	0.00	0.00	0
OSPF	0.00	0.00	0.00	0.00	0
RIP	0.00	0.00	0.00	0.00	0
STP	0.00	0.00	0.00	0.00	0
VRRP	0.00	0.00	0.00	0.00	0

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. An example is given below.

```
Brocade#show process cpu
```

The system has only been up for 6 seconds.

Process Name	5Sec(%)	1Min(%)	5Min(%)	15Min(%)	Runtime(ms)
ARP	0.01	0.00	0.00	0.00	0
BGP	0.00	0.00	0.00	0.00	0
GVRP	0.00	0.00	0.00	0.00	0
ICMP	0.01	0.00	0.00	0.00	1
IP	0.00	0.00	0.00	0.00	0
OSPF	0.00	0.00	0.00	0.00	0
RIP	0.00	0.00	0.00	0.00	0
STP	0.00	0.00	0.00	0.00	0
VRRP	0.00	0.00	0.00	0.00	0

To display utilization statistics for a specific number of seconds, enter a command such as the following.

```
Brocade#show process cpu 2
```

Statistics for last 1 sec and 80 ms

Process Name	Sec(%)	Time(ms)
ARP	0.00	0
BGP	0.00	0
GVRP	0.01	1
ICMP	0.00	0
IP	0.00	0
OSPF	0.00	0
RIP	0.00	0
STP	0.01	1
VRRP	0.00	0

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: `show process cpu [<num>]`

The `<num>` parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Clearing GVRP statistics

To clear the GVRP statistics counters, enter the **clear gvrp statistics all** command.

```
Brocade#clear gvrp statistics all
```

This command clears the counters for all ports. To clear the counters for a specific port only, enter a command such as the following.

```
Brocade#clear gvrp statistics ethernet 2/1
```

Syntax: `clear gvrp statistics all | ethernet <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

GVRP CLI examples

The following sections show the CLI commands for implementing the applications of GVRP described in [“GVRP application examples”](#) on page 884.

NOTE

Although some of the devices in these configuration examples do not have statically configured VLANs, this is not a requirement. You always can have statically configured VLANs on a device that is running GVRP.

Dynamic core and fixed edge

In this configuration, the edge devices advertise their statically configured VLANs to the core device. The core device does not have any statically configured VLANs but learns the VLANs from the edge devices.

Enter the following commands on the core device.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable all
```

These commands globally enable GVRP support and enable the protocol on all ports.

Enter the following commands on edge device A.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#vlan 20
Brocade(config-vlan-20)#untag ethernet 2/1
Brocade(config-vlan-20)#tag ethernet 4/24
Brocade(config-vlan-20)#vlan 40
Brocade(config-vlan-40)#untag ethernet 2/1
Brocade(config-vlan-40)#tag ethernet 4/24
Brocade(config-vlan-40)#exit
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable ethernet 4/24
Brocade(config-gvrp)#block-learning ethernet 4/24
```

These commands statically configure two port-based VLANs, enable GVRP on port 4/24, and block GVRP learning on the port. The device will advertise the VLANs but will not learn VLANs from other devices.

Enter the following commands on edge device B.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#vlan 20
Brocade(config-vlan-20)#untag ethernet 2/24
Brocade(config-vlan-20)#tag ethernet 4/1
Brocade(config-vlan-20)#vlan 30
Brocade(config-vlan-30)#untag ethernet 4/24
Brocade(config-vlan-30)#tag ethernet 4/1
Brocade(config-vlan-30)#exit
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable ethernet 4/1
Brocade(config-gvrp)#block-learning ethernet 4/1
```

Enter the following commands on edge device C.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#vlan 30
Brocade(config-vlan-30)#untag ethernet 2/24
Brocade(config-vlan-30)#tag ethernet 4/1
Brocade(config-vlan-20)#vlan 40
Brocade(config-vlan-40)#untag ethernet 4/24
Brocade(config-vlan-40)#tag ethernet 4/1
Brocade(config-vlan-40)#exit
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable ethernet 4/1
Brocade(config-gvrp)#block-learning ethernet 4/1
```

Dynamic core and dynamic edge

In this configuration, the core and edge devices have no statically configured VLANs and are enabled to learn and advertise VLANs. The edge and core devices learn the VLANs configured on the devices in the edge clouds. To enable GVRP on all the ports, enter the following command on each edge device **and** on the core device.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable all
```

Fixed core and dynamic edge

In this configuration, GVRP learning is enabled on the edge devices. The VLANs on the core device are statically configured, and the core device is enabled to advertise its VLANs but not to learn VLANs. The edge devices learn the VLANs from the core.

Enter the following commands on the core device.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#vlan 20
Brocade(config-vlan-20)#tag ethernet 1/24
Brocade(config-vlan-20)#tag ethernet 6/24
Brocade(config-vlan-20)#vlan 30
Brocade(config-vlan-30)#tag ethernet 6/24
Brocade(config-vlan-30)#tag ethernet 8/17
Brocade(config-vlan-30)#vlan 40
Brocade(config-vlan-40)#tag ethernet 1/5
Brocade(config-vlan-40)#tag ethernet 8/17
Brocade(config-vlan-40)#vlan 50
Brocade(config-vlan-50)#untag ethernet 6/1
Brocade(config-vlan-50)#tag ethernet 1/11
Brocade(config-vlan-50)#exit
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable ethernet 1/24 ethernet 6/24 ethernet 8/17
Brocade(config-gvrp)#block-learning ethernet 1/24 ethernet 6/24 ethernet 8/17
```

These VLAN commands configure VLANs 20, 30, 40, and 50. The GVRP commands enable the protocol on the ports that are connected to the edge devices, and disable VLAN learning on those ports. All the VLANs are advertised by GVRP.

Enter the following commands on edge devices A, B, and C.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#gvrp-enable
Brocade(config-gvrp)#enable all
Brocade(config-gvrp)#block-applicant all
```

Fixed core and fixed edge

The VLANs are statically configured on the core and edge devices. On each edge device, VLAN advertising is enabled but learning is disabled. GVRP is not configured on the core device. This configuration enables the devices in the edge clouds to learn the VLANs configured on the edge devices.

This configuration does not use any GVRP configuration on the core device.

The configuration on the edge device is the same as in [“Dynamic core and fixed edge”](#) on page 902.

MAC-based VLANs

Table 154 lists the individual Brocade FastIron switches and the MAC-based VLAN features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 154 Supported MAC-based VLAN features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
MAC-Based VLANs:	No	Yes	Yes	Yes	Yes
<ul style="list-style-type: none"> Source MAC address authentication Policy-based classification and forwarding 					
MAC-based VLANs and 802.1X security on the same port	No	Yes	Yes	Yes	Yes
MAC-based VLAN aging	No	Yes	Yes	Yes	Yes
Dynamic MAC-Based VLANs	No	Yes	Yes	Yes	Yes

MAC-based VLAN overview

The MAC-based VLAN feature controls network access by authenticating a host source MAC address, and mapping the incoming packet source MAC to a VLAN. Mapping is based on the MAC address of the end station connected to the physical port. Users who relocate can remain on the same VLAN as long as they connect to any switch in the same domain, on a port which is permitted in the VLAN. The MAC-based VLAN feature may be enabled for two types of hosts: static and dynamic.

MAC-based VLAN activity is determined by authentication through a RADIUS server. Incoming traffic that originates from a specific MAC address is forwarded only if the source MAC address-to-VLAN mapping is successfully authenticated. While multi-device port authentication is in progress, all traffic from the new MAC address will be blocked or dropped until the authentication succeeds. Traffic is dropped if the authentication fails.

Static and dynamic hosts

Static hosts are devices on the network that do not speak until spoken to. Static hosts may not initiate a request for authentication on their own. Such static hosts can be managed through a **link up** or **link down** notification.

Dynamic hosts are “chatty” devices that generate packets whenever they are in the **link up** state. Dynamic hosts must be authenticated before they can switch or forward traffic.

MAC-based VLAN feature structure

The MAC-based VLAN feature operates in two stages:

- Source MAC Address Authentication
- Policy-Based Classification and Forwarding

Source MAC address authentication

Source MAC address authentication is performed by a central RADIUS server when it receives a PAP request with a username and password that match the MAC address being authenticated. When the MAC address is successfully authenticated, the server must return the VLAN identifier, which is carried in the Tunnel-Type, Tunnel-Medium-Type, and Tunnel-Private-Group-ID attributes of the RADIUS packets. If the Tunnel-Type is *tagged*, the MAC address will be blocked or restricted. If the identified VLAN does not exist, then the authentication is considered a failure, and action is taken based on the configured failure options. (The default failure action is to drop the traffic.) The RADIUS server may also optionally return the QoS attribute for the authenticated MAC address. Refer to [Table 157](#) on page 911 for more information about attributes.

Policy-based classification and forwarding

Once the authentication stage is complete, incoming traffic is classified based on the response from the RADIUS server. There are three possible actions:

- Incoming traffic from a specific source MAC is dropped because authentication failed
- Incoming traffic from a specific source MAC is classified as untagged into a specific VLAN
- Incoming traffic from a specific source MAC is classified as untagged into a restricted VLAN

Traffic classification is performed by programming incoming traffic and RADIUS-returned attributes in the hardware. Incoming traffic attributes include the source MAC address and the port on which the feature is enabled. The RADIUS-returned attributes are the VLAN into which the traffic is to be classified, and the QoS priority.

NOTE

This feature drops any incoming *tagged* traffic on the port, and classifies and forwards untagged traffic into the appropriate VLANs.

This feature supports up to a maximum of 32 MAC addresses per physical port, with a default of 2.

NOTE

Even though the feature supports up to a maximum of 32 MAC address per physical port, the configuration of the maximum number of MAC addresses per port is limited by the available hardware resources.

Once a client MAC address is successfully authenticated and registered, the MAC-to-VLAN association remains until the port connection is dropped, or the MAC entry expires.

MAC-based VLAN and port up or down events

When the state of a port is changed to *down*, all authorized and unauthorized MAC addresses are removed from the MAC-to-VLAN mapping table, any pending authentication requests are cancelled.

Dynamic MAC-based VLAN

When enabled, the dynamic MAC-based VLAN feature allows the dynamic addition of mac-vlan-permit ports to the VLAN table only after successful RADIUS authentication. Ports that fail RADIUS authentication are not added to the VLAN table.

When this feature is not enabled, the physical port is statically added to the hardware table, regardless of the outcome of the authentication process. This feature prevents the addition of unauthenticated ports to the VLAN table. For information about how to configure Dynamic MAC-based VLAN, refer to [“Configuring dynamic MAC-based VLAN”](#) on page 914.

Configuration notes and feature limitations for dynamic MAC-based VLAN

The following guidelines apply to MAC-based VLAN configurations:

- MAC-based VLAN is not currently supported for trunk ports and LACP.
- MAC-based VLAN is not supported for VLAN groups, topology groups and dual-mode configuration.
- MAC-based VLAN is not supported together with ACLs or MAC address filters.
- FastIron devices do not support UDLD link-keepalives on ports with MAC-based VLAN enabled.
- FastIron devices do not support STP BPDU packets on ports with MAC-based VLAN enabled.
- MAC-to-VLAN mapping must be associated with VLANs that exist on the switch. Create the VLANs before you configure the MAC-based VLAN feature.
- Ports participating in MAC-based VLANs must first be configured as **mac-vlan-permit** ports under the VLAN configuration.
- In the RADIUS server configuration file, a MAC address cannot be configured to associate with more than one VLAN.
- This feature does not currently support dynamic assignment of a port to a VLAN. Users must pre-configure VLANs and port membership before enabling the feature.
- Multi-device port authentication filters will not work with MAC-based VLANs on the same port.

Dynamic MAC-based VLAN CLI commands

The following table describes the CLI commands used to configure MAC-based VLANs.

TABLE 155 CLI commands for MAC-based VLANs

CLI command	Description	CLI level
mac-auth mac-vlan enable	Enables per-port MAC-based VLAN	Interface
mac-auth mac-vlan disable	Disables per-port MAC-based VLAN	interface
mac-auth mac-vlan-dyn-activation	Enables Dynamic MAC-based VLAN	global
no mac-auth mac-vlan-dyn-activation	Disables Dynamic MAC-based VLAN	global
no mac-auth mac-vlan	Removes the MAC-VLAN configuration from the port	interface

TABLE 155 CLI commands for MAC-based VLANs (Continued)

CLI command	Description	CLI level
mac-auth mac-vlan max-mac-entries <num of entries>	The maximum number of allowed and denied MAC addresses (static and dynamic) that can be learned on a port. The default is 2.	interface
mac-auth mac-vlan <mac-addr> vlan <vlan id> priority <0-7>	Adds a static MAC-VLAN mapping to the MAC-based VLAN table (for static hosts)	interface
clear table-mac-vlan	Clears the contents of the authenticated MAC address table	global
clear table-mac-vlan ethernet <port>	Clears all MAC-based VLAN mapping on a port	global
show table-mac-vlan	Displays information about allowed and denied MAC addresses on ports with MAC-based VLAN enabled.	global
show table-mac-vlan allowed-mac	Displays MAC addresses that have been successfully authenticated	global
show table-mac-vlan denied-mac	Displays MAC addresses for which authentication failed	global
show table-mac-vlan detailed	Displays detailed MAC-VLAN settings and classified MAC addresses for a port with the feature enabled	global
show table-mac-vlan <mac-address>	Displays status and details for a specific MAC address	global
show table-mac-vlan ethernet <port>	Displays all MAC addresses allowed or denied on a specific port	global

Dynamic MAC-based VLAN configuration example

The following example shows a MAC-based VLAN configuration.

```
Brocade#show run
Current configuration:
ver 04.0.00b122T7e1
fan-threshold mp speed-3 35 100
module 1 fls-24-port-copper-base-module
module 4 fls-xfp-1-port-10g-module
vlan 1 by port
  untagged ethe 0/1/10
  mac-vlan-permit ethe 0/1/1 to 0/1/3
  no spanning-tree
vlan 2 by port
  untagged ethe 0/1/24
  mac-vlan-permit ethe 0/1/1 to 0/1/3
  no spanning-tree
vlan 222 name RESTRICTED_MBV by port
  untagged ethe 0/1/4
  mac-vlan-permit ethe 0/1/1 to 0/1/3
vlan 666 name RESTRICTED_MAC_AUTH by port
  untagged ethe 0/1/20
  mac-vlan-permit ethe 0/1/1 to 0/1/3
  spanning-tree 802-1w
vlan 4000 name DEFAULT-VLAN by port
vlan 4004 by port
```

```

mac-vlan-permit ethe 0/1/1 ethe 0/1/3
default-vlan-id 4000
ip address 10.44.3.3 255.255.255.0
ip default-gateway 10.44.3.1
radius-server host 10.44.3.111
radius-server key 1 $-ndUno
mac-authentication enable
mac-authentication mac-vlan-dyn-activation
mac-authentication max-age 60
mac-authentication hw-deny-age 30
mac-authentication auth-passwd-format xxxx.xxxx.xxxx
mac-authentication auth-fail-vlan-id 666
interface ethernet 0/1/1
  mac-authentication mac-vlan max-mac-entries 5
  mac-authentication mac-vlan 0030.4888.b9fe vlan 1 priority 1
  mac-authentication mac-vlan enable
interface ethernet 0/1/2
  mac-authentication mac-vlan max-mac-entries 10
  mac-authentication mac-vlan enable
  mac-authentication auth-fail-action restrict-vlan 222
interface ethernet 0/1/3
  mac-authentication mac-vlan enable
  mac-authentication auth-fail-action restrict-vlan
!
end

```

MAC-based VLAN configuration

Configure MAC-based VLAN mapping on the switch statically for static hosts, or dynamically for non-static hosts, by directing the RADIUS server to authenticate the incoming packet.

To configure the a MAC-based VLAN, first perform the following tasks:

- In the VLANs, configure **mac-vlan-permit** for each port that will be participating in the MAC-based VLAN
- If a port has been MAC-based VLAN-enabled, but has **not** been added as **mac-vlan-permit** in any of the VLANs, any MAC addresses learned on this port will be blocked in the reserved VLAN. To prevent this, you must create all of the VLANs and add all ports as **mac-vlan-permit** **before** enabling MAC-based VLAN on any ports.
- Disable any multi-device port authentication on ports you will be using for MAC-to-VLAN mapping

NOTE

Do not configure MAC-based VLAN on ports that are tagged to any VLAN. Do not use ports on which MAC-based VLAN is configured as tagged ports.

NOTE

For FCX devices, MAC-based VLAN with 802.1X will not work on the same port if 802.1X has the RADIUS VLAN attribute defined as an untagged VLAN (for example U:1, U:2).

NOTE

MAC-based VLAN is not supported on trunk or LACP ports. Do not configure trunks on MAC-based VLAN-enabled ports.

Using MAC-based VLANs and 802.1X security on the same port

On Brocade devices, MAC-based VLANs and 802.1X security can be configured on the same port. When both of these features are enabled on the same port, MAC-based VLAN is performed prior to 802.1X authentication. If MAC-based VLAN is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

When both features are configured on a port, a device connected to the port is authenticated as follows.

1. MAC-based VLAN is performed on the device to authenticate the device MAC address.
2. If MAC-based VLAN is successful, the device then checks to see if the RADIUS server included the Foundry-802_1x-enable VSA (described in [Table 157](#)) in the Access-Accept message that authenticated the device.
3. If the Foundry-802_1x-enable VSA is not present in the Access-Accept message, or is present and set to 1, then 802.1X authentication is performed for the device.
4. If the Foundry-802_1x-enable VSA is present in the Access-Accept message, and is set to 0, then 802.1X authentication is skipped.

Configuring generic and Brocade vendor-specific attributes on the RADIUS server

If the RADIUS authentication process is successful, the RADIUS server sends an Access-Accept message to the Brocade device, authenticating the device. The Access-Accept message includes Vendor-Specific Attributes (VSAs) that specify additional information about the device.

Add Brocade vendor-specific attributes to your RADIUS server configuration, and configure the attributes in the individual or group profiles of the devices that will be authenticated. Brocade vendor-ID is 1991, vendor-type 1. [Table 156](#) lists generic RADIUS attributes. [Table 157](#) lists Brocade Vendor-Specific Attributes.

TABLE 156 Generic RADIUS attributes

Attribute name	Attribute ID	Data type	Optional or mandatory	Description
Tunnel-Type	64	13 decimal VLAN	Mandatory	RFC 2868.
Tunnel-Medium-Type	65	6 decimal 802	Mandatory	RFC 2868.
Tunnel-Private-Group-ID	81	decimal	Mandatory	RFC 2868. <vlan-id> or U:<vlan -id> – a MAC-based VLAN ID configured on the Brocade device.

TABLE 157 Brocade vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Optional or mandatory	Description
Foundry-MAC-based VLAN-QoS	8	decimal	Optional	The QoS attribute specifies the priority of the incoming traffic based on any value between 0 (lowest priority) and 7 (highest priority). Default is 0.
Foundry-802_1x-enabled	6	integer	Optional	Specifies whether 802.1X authentication is performed when MAC-based VLAN is successful for a device. This attribute can be set to one of the following: 0 - Do not perform 802.1X authentication on a device that passes MAC-based VLAN. Set the attribute to zero (0) for devices that do not support 802.1X authentication. 1 - Perform 802.1X authentication when a device passes MAC-based VLAN. Set the attribute to one (1) for devices that support 802.1X authentication.
Foundry-802_1x-val id	7	integer	Optional	Specifies whether the RADIUS record is valid only for MAC-based VLAN, or for both MAC-based VLAN and 802.1X authentication. This attribute can be set to one of the following: 0 - The RADIUS record is valid only for MAC-based VLAN. Set this attribute to zero (0) to prevent a user from using their MAC address as username and password for 802.1X authentication 1 - The RADIUS record is valid for both MAC-based VLAN and 802.1X authentication.

Aging for MAC-based VLAN

The aging process for MAC-based VLAN works as described below.

For permitted hosts

For permitted hosts, as long as the Brocade device is receiving traffic aging does not occur. The age column in the output of the **show table-mac-vlan** command displays Ena or S <num>. If the Brocade device stops receiving traffic, the entry first ages out from the MAC table (in the hardware) and then the aging cycle for MAC-based VLAN begins. Aging in the MAC-based VLAN continues for 2 minutes (the default is 120 seconds) after which the MAC-based VLAN session is flushed out.

For blocked hosts

For blocked hosts, as long as the Brocade device is receiving traffic, aging does not occur. In the output of the **show table-mac-vlan command**, the age column displays H0 to H70, S0, and H0 to H70, etc. Aging of the MAC-based VLAN MAC occurs in two phases: hardware aging and software aging. The hardware aging period can be configured using the **mac-authentication hw-deny-age** command in config mode. The default is 70 seconds. The software aging time for MAC-based VLAN MACs can be configured using the **mac-authentication max-age** command. When the Brocade device is no longer receiving traffic from a MAC-based VLAN MAC address, the hardware aging period begins and lasts for a fixed length of time (default or user-configured). When the hardware

aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (the default is 120 seconds). After the software aging period ends, the MAC-based VLAN session is flushed, and the MAC address can be authenticated or denied if the Brocade device again receives traffic from that MAC address.

For MAC-based dynamic activation

If all of the sessions age out on a port, the port is dynamically removed from the VLAN table. When any new session is established, the port is dynamically added back to the VLAN table.

NOTE

If the Brocade device receives a packet from an authenticated MAC address, and the MAC-based VLAN software aging is still in progress (hardware aging has already occurred), a RADIUS message is NOT sent to the RADIUS server. Instead the MAC address is reentered in the hardware along with the parameters previously returned from the RADIUS server. A RADIUS message is sent only when the MAC-based VLAN session ages out from the software.

To change the length of the software aging period

To change the length of the software aging period for blocked MAC addresses, enter a command such as the following.

```
Brocade(config)#mac-authentication max-age 180
```

Syntax: [no] **mac-authentication max-age** <seconds>

You can specify from 1 – 65535 seconds. The default is 120 seconds.

Disabling aging for MAC-based VLAN sessions

MAC addresses that have been authenticated or denied by a RADIUS server are aged out if no traffic is received from the MAC address for a certain period of time.

You can optionally disable aging for MAC-based VLAN session subject to authentication, either for all MAC addresses or for those learned on a specified interface.

Globally disabling aging

On most devices, you can disable aging on all interfaces where MAC-based VLAN has been enabled, by entering the following command.

```
Brocade(config)#mac-authentication disable-aging
```

Syntax: **mac-authentication disable-aging**

Enter the command at the global or interface configuration level.

The **denied-mac-only** parameter prevents denied sessions from being aged out, but ages out permitted sessions.

The **permitted-mac-only** parameter prevents permitted (authenticated and restricted) sessions from being aged out and ages denied sessions.

Disabling the aging on interfaces

To disable aging on a specific interface where MAC-based VLAN has been enabled, enter the command at the interface level.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication disable-aging
```

Syntax: [no] **mac-authentication disable-aging**

Configuring the maximum MAC addresses per port

To configure the maximum number of MAC addresses allowed per port, use the following commands:

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#mac-authentication mac-vlan max-mac-entries 24
```

NOTE

32 MAC addresses maximum are allowed per port. This total includes both static and dynamic hosts. The default number of allowed MACs is 2. Even though the feature supports up to a maximum of MAC address per physical port, the configuration of the maximum number of MAC addresses per port is limited by the available hardware resources.

NOTE

To change the maximum MAC addresses per port, you must first disable MAC-based VLAN on that port.

Configuring a MAC-based VLAN for a static host

Follow the steps given below to configure a MAC-based VLAN for a static host.

1. Enable multi-device port authentication globally using the following command.

```
Brocade(config)#mac-authentication enable
```

2. Add each port on which you want MAC-based VLAN enabled as **mac-vlan-permit** for a specific VLAN.

```
Brocade(config)#vlan 10 by port
Brocade(config-vlan-10)#mac-vlan-permit ethernet 0/1/1 to 0/1/6
added mac-vlan-permit ports ethe 0/1/1 to 0/1/6 to port-vlan 10.
```

3. Add the static MAC-based VLAN configuration on the port.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#mac-authentication mac-vlan 0000.0010.0011
vlan 10 priority 5
```

4. To enable MAC-based VLAN on the port.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#mac-authentication mac-vlan enable
```

5. To disable MAC-based VLAN on the port.

```
Brocade(config)#interface e 0/1/1
Brocade(interface-0/1/1)#mac-auth mac-vlan disable
```

6. To remove and disable the MAC-based VLAN configuration.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#no mac-auth mac-vlan
```

Configuring MAC-based VLAN for a dynamic host

Follow the steps given below to configure MAC-based VLAN for a dynamic host.

1. Enable multi-device port authentication globally using the following command.

```
Brocade(config)#mac-authentication enable
```

2. Add each port on which you want MAC-based VLAN enabled as **mac-vlan-permit** for a specific VLAN.

```
Brocade(config)#vlan 10 by port
Brocade(config-vlan-10)#mac-vlan-permit ethernet 0/1/1 to 0/1/6
```

3. Enable MAC-based VLAN on the port.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#mac-authentication mac-vlan enable
```

4. Disable MAC-based VLAN on the port.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#mac-auth mac-vlan disable
```

5. Remove and disable the MAC-based VLAN configuration.

```
Brocade(config)#interface e 0/1/1
Brocade(config-if-e1000-0/1/1)#no mac-auth mac-vlan
```

Configuring dynamic MAC-based VLAN

To globally enable MAC-based VLAN globally (for all MAC-based VLAN ports), enter the following commands.

```
Brocade(config)#mac-authentication enable
Brocade(config)#mac-authentication mac-vlan-dyn-activation
```

To configure Dynamic MAC-based VLAN to add a specific port to a specific VLAN, enter commands similar to the following.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#mac-vlan-permit e 0/1/35
```

Syntax: **mac-vlan-permit ethernet** <stack-unit/slotnum/portnum>

To disable Dynamic MAC-based VLAN, enter the following command.

```
Brocade(config)#no mac-authentication mac-vlan-dyn-activation
```

NOTE

If static Mac-Based VLAN is configured on a port, the port will be added only to the VLAN table for which the static MAC-based VLAN configuration exists.

NOTE

If the Dynamic MAC-based VLAN is enabled after any MAC-based VLAN sessions are established, all sessions are flushed and the mac-vlan-permit ports are removed from the VLAN. The ports are then added back to the VLAN dynamically after they successfully pass the RADIUS authentication process.

Configuring MAC-based VLANs using SNMP

Several MIB objects have been developed to allow the configuration of MAC-based VLANs using SNMP. For more information, refer to the *IronWare MIB Reference Guide*.

Displaying Information about MAC-based VLANs

This section describes the **show** commands that display information related to MAC-based VLANs.

Displaying the MAC-VLAN table

Enter the following command to display the MAC-VLAN table.

```
Brocade(config)#show table-mac-vlan
```

Port	Vlan	Accepted Macs	Rejected Macs	Attempted Macs	Static Macs	Static Conf	Max Macs
1/1/1	N/A	1	1	0	0	1	10

Syntax: **show table-mac-vlan**

The following table describes the information in this output.

Field	Description
Port	The port number where MAC-based VLAN is enabled.
Vlan	Not applicable for this feature, will always display n/a.
Accepted Macs	The number of MAC addresses that have been successfully authenticated (dynamic hosts) combined with the number of active static MAC addresses (static hosts).
Rejected Macs	The number of MAC addresses for which authentication has failed for dynamic hosts.
Attempted Macs	The number of attempts made to authenticate MAC addresses.
Static Macs	The number of currently connected active static hosts.
Static Conf	The number of static hosts that are configured on the physical port.
Max Macs	The maximum number of allowed MAC addresses.

Displaying the MAC-VLAN table for a specific MAC address

Enter the **show table-mac-vlan** command to display the MAC-VLAN table information for a specific MAC address.

```
Brocade(config)#show table-mac-vlan 0000.0010.1001
```

```
-----
MAC Address      Port          Vlan  Authenticated  Time      Age      dot1x
-----
0000.0010.1001  1/1/1        2     Yes           00d00h05m45s  Ena      Dis
```

Syntax: show table-mac-vlan <mac-address>

The following table describes the information in this output.

Field	Description
MAC Address	The MAC address for which this information is displayed.
Port	The port where MAC-based VLAN is enabled.
Vlan	The VLAN to which the MAC address has been assigned.
Authenticated	Yes indicates authentication is successful. No indicates authentication has failed. Inp indicates authentication in progress Rst indicates a restricted VLAN
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates if 802.1X authentication is enabled or disabled for the MAC address.

Displaying allowed MAC addresses

Enter the **show table-mac-vlan allowed-mac** command to display information about successfully authenticated MAC addresses.

```
Brocade#show table-mac-vlan allowed-mac
```

```
-----
MAC Address      Port          Vlan  Authenticated  Time      Age      dot1x
-----
0030.4874.3181  2/1/17        76     Yes           00d01h17m22s  Ena      Dis
```

Syntax: show table-mac-vlan allowed-mac

The following table describes the information in this output.

Field	Description
MAC Address	The allowed MAC addresses for which the information is displayed.
Port	The port where MAC-based VLAN is enabled.
Vlan	The VLAN to which the MAC address has been assigned.
Authenticated	Yes indicates authentication has been successful. Inp indicates authentication is in progress.

Field	Description
Time	The time at which each MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates whether 802.1X authentication is enabled or disabled for each MAC address.

Displaying denied MAC addresses

Enter the **show table-mac-vlan denied-mac** command to display information about denied (authentication failed) MAC addresses.

```
Brocade(config)#show table-mac-vlan denied-mac
```

```
-----
MAC Address      Port          Vlan  Authenticated Time      Age      dot1x
-----
0000.0030.1002   1/1/1         4092 No       00d00h11m57s H40      Dis
```

Syntax: show table-mac-vlan denied-mac

The following table describes the information in this output.

Field	Description
MAC Address	The denied MAC address for which the information is displayed.
Port	The port where MAC-based VLAN is enabled.
Vlan	This field displays VLAN 4092 for blocked hosts, or the restricted VLAN ID if it is configured on the port.
Authenticated	No indicates that authentication has failed. Inp indicates that authentication is in progress.
Time	The time at which authentication failed.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates whether 802.1X authentication is disabled (Dis) or enabled (Ena) for this MAC address.

Displaying detailed MAC-VLAN data

Enter the **show table-mac-vlan detailed** command to display a detailed version of MAC-VLAN information.

```
Brocade#show table-mac-vlan detailed e 0/1/2
```

```
Port                               : 0/1/2
Dynamic-Vlan Assignment            : Disabled
RADIUS failure action              : Block Traffic
    Failure restrict use dot1x     : No
Override-restrict-vlan            : Yes
Vlan                               : (MAC-PERMIT-VLAN )
Port Vlan State                    : DEFAULT
802.1X override Dynamic PVID      : NO
Original PVID                      : 1
DOS attack protection              : Disabled
Accepted Mac Addresses             : 32
Rejected Mac Addresses             : 0
Authentication in progress         : 0
Authentication attempts            : 54
RADIUS timeouts                    : 16817
Num of MAC entries in TCAM         : 32
Num of MAC entries in MAC          : 32
Aging of MAC-sessions              : Enabled
Port move-back vlan                : Port-configured-vlan
Max-Age of sw mac session          : 60 seconds
hw age for denied mac              : 30 seconds
MAC Filter applied                 : No
```

MAC Address	RADIUS	Authenticated	Time	Age	CAM Index	MAC Index	Dot1x	Type	Pri
0000.0200.0012	0.0.0.0	No	00d00h00m00s	S12	N/A	N/A	Dis	Dyn	0
0000.0200.0017	0.0.0.0	No	00d00h00m00s	S20	N/A	N/A	Dis	Dyn	0
0000.0200.0018	0.0.0.0	No	00d00h00m00s	S20	N/A	N/A	Dis	Dyn	0
0000.0100.000a	10.44.3.111	Yes	00d19h38m30s	Ena	000b	22d4	Dis	Dyn	5
0000.0200.0019	0.0.0.0	No	00d00h00m00s	S20	N/A	N/A	Dis	Dyn	0
0000.0200.001a	0.0.0.0	No	00d00h00m00s	S20	N/A	N/A	Dis	Dyn	0
0000.0200.001b	0.0.0.0	No	00d00h00m00s	S20	N/A	N/A	Dis	Dyn	0
0000.0200.001c	0.0.0.0	No	00d00h00m00s	S20	N/A	N/A	Dis	Dyn	0
0000.0200.001d	0.0.0.0	No	00d00h00m00s	S20	N/A	N/A	Dis	Dyn	0

MAC Address	RADIUS	Authenticated	Time	Age	CAM Index	MAC Index	Dot1x	Type	Pri
0000.feed.1111	0.0.0.0	No	07d17h00m43s	S0	0000	4000	Dis	Sta	1
0000.feed.1112	0.0.0.0	No	07d17h01m51s	S0	0001	4000	Dis	Sta	2
0000.feed.1113	0.0.0.0	No	07d17h03m00s	S0	0002	4000	Dis	Sta	3

Displaying MAC-VLAN information for a specific interface

Enter the **show table-mac-vlan e** command to display MAC-VLAN information for a specific interface.

```
Brocade#show table-mac-vlan e 0/1/1
```

MAC Address	Port	Vlan	Authenticated	Time	Age	CAM Index	MAC Index	Dot1x	Type	Pri
0000.0100.0001	0/1/1	1	Yes	00d19h38m29s	Ena	0008	0970	Dis	Dyn	0
0000.0100.0002	0/1/1	1	Yes	00d19h38m29s	Ena	0009	0a40	Dis	Dyn	1
0000.0100.0003	0/1/1	1	Yes	00d19h38m30s	Ena	000a	2b44	Dis	Dyn	2
0000.0100.0004	0/1/1	1	Yes	00d19h38m49s	S96	0013	4000	Dis	Dyn	3
0000.0100.0005	0/1/1	1	Yes	00d19h38m53s	Ena	0014	2d24	Dis	Dyn	4
0000.0100.0006	0/1/1	1	Yes	00d19h38m53s	Ena	0015	2e14	Dis	Dyn	5
0000.0100.0007	0/1/1	1	Yes	00d19h38m41s	S80	000f	4000	Dis	Dyn	6
0000.0100.0008	0/1/1	1	Yes	00d19h39m07s	Ena	001f	00e0	Dis	Dyn	7
0000.0100.000a	0/1/1	1	Yes	00d19h38m30s	Ena	000b	22d4	Dis	Dyn	0
0000.0100.0009	0/1/1	1	Yes	00d19h38m19s	Ena	0001	21e4	Dis	Dyn	0
0000.0100.000a	0/1/1	1	Yes	00d19h38m30s	Ena	000b	22d4	Dis	Dyn	0
0000.0100.000b	0/1/1	1	Yes	00d19h38m19s	Ena	0002	03d0	Dis	Dyn	0
0000.0100.000c	0/1/1	1	Yes	00d19h38m57s	Ena	001a	24b4	Dis	Dyn	0
0000.0100.000d	0/1/1	1	Yes	00d19h38m19s	Ena	0003	05b0	Dis	Dyn	0
0000.0100.000e	0/1/1	1	Yes	00d19h38m31s	S120	000c	4000	Dis	Dyn	0
0000.0100.000f	0/1/1	1	Yes	00d19h38m20s	Ena	0004	2784	Dis	Dyn	0
0000.0100.0010	0/1/1	1	Yes	00d19h39m04s	S32	001d	4000	Dis	Dyn	0
0000.0100.0011	0/1/1	1	Yes	00d19h38m43s	Ena	0010	3864	Dis	Dyn	0
0000.0100.0012	0/1/1	1	Yes	00d19h38m39s	Ena	000d	3b54	Dis	Dyn	0

The following table describes the information in this output.

Field	Description
MAC Address	The MAC addresses related to the specified interface.
Port	The interface for which this information is displayed.
Vlan	The VLAN to which the interface has been assigned.
Authenticated	Yes indicates authentication is successful. No indicates authentication has failed. Inp indicates authentication in progress Rst indicates a restricted VLAN
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	This field displays the index of the CAM entry. The index value will be between 0 and 31. A value of "ff" indicates that the index is not used.
MAC Index	The index of the entry in the hardware MAC table.
Dot1x	Indicates whether 802.1X authentication is enabled or disabled for this MAC address.

Field	Description
Type	Dyn Indicates a dynamic host. Sta indicates a static host.
Pri	This field indicates the value set for Foundry-MAC-based VLAN-QoS attribute in the RADIUS configuration for dynamic hosts, if configured. If the Foundry-MAC-based VLAN-QoS attribute is not configured, the value will be zero. For static hosts, the user-configured priority value for the MAC address is displayed.

Displaying MAC addresses in a MAC-based VLAN

Enter the **show mac-address** command to display a list of MAC addresses in a MAC-based VLAN.

```
Brocade#show mac-address
Total active entries from all ports = 1541
MAC-Address      Port          Type          Index      VLAN
0000.2000.0001   0/1/32        Dynamic (MBV)  1048       1
0000.2000.0002   0/1/32        Dynamic (MBV)  1832       1
0000.2000.0003   0/1/32        Dynamic (MBV)  9772       1
0000.2000.0004   0/1/32        Static (MBV)   328        1
0000.2000.0005   0/1/32        Dynamic (MBV)  8268       1
0000.2000.0006   0/1/32        Dynamic (MBV)  9084       1
0000.2000.0007   0/1/32        Dynamic (MBV)  632        1
0000.2000.0008   0/1/32        Dynamic (MBV)  3464       1
0000.2000.0009   0/1/32        Dynamic (MBV)  11404      1
0000.2000.000a   0/1/32        Dynamic (MBV)  12220      1
0000.2000.000b   0/1/32        Dynamic (MBV)  3768       1
```

NOTE

In this output, (MBV) indicates MAC-based VLAN is enabled.

The following table describes the output from this command.

Field	Description
Total active entries	The total number of active entries for all ports.
MAC Address	The MAC addresses assigned to this VLAN.
Port	The interface for which this information is displayed.
Type	Dynamic (MBV) Indicates a dynamic host. Static (MBV) indicates a static host.
Index	The index of the entry in the hardware MAC table.
VLAN	The VLAN to which these addresses are assigned.

Displaying MAC-based VLAN logging

Enter the **show logging** command to display MAC-based VLAN logging activity.

```
Brocade#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 15 overruns)
  Buffer logging: level ACDMEINW, 50 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
Static Log Buffer
0d00h00m12s:A:System: Power supply 1 is up
Dynamic Log Buffer (50 lines):
0d18h46m28s:I:running-config was changed from console
0d02h12m25s:A:MAC Based Vlan Mapping failed for [0000.1111.0108 ] on port 0/2/1
(Invalid User)
0d02h08m52s:A:MAC Based Vlan Mapping failed for [0000.1111.011b ] on port 0/2/1
(Invalid User)
0d02h05m01s:A:MAC Based Vlan Mapping failed for [0000.1111.00df ] on port 0/2/1
(Invalid User)
0d02h01m15s:A:MAC Based Vlan Mapping failed for [0000.1111.0108 ] on port 0/2/1
(Invalid User)
0d02h01m15s:A:MAC Based Vlan Mapping failed for [0000.1111.0107 ] on port 0/2/1
(Invalid User)
0d01h58m43s:N:MAC Based Vlan Enabled on port 0/2/1
0d01h58m32s:N:MAC Based Vlan Disabled on port 0/2/1
0d01h39m00s:I:running-config was changed from console
0d01h38m28s:I:System: Interface ethernet 0/1/47, state up
0d01h38m27s:I:System: Interface ethernet 0/1/46, state up
0d01h38m27s:I:System: Interface ethernet 0/1/34, state up
0d01h38m27s:I:System: Interface ethernet 0/1/25, state up
```

Clearing MAC-VLAN information

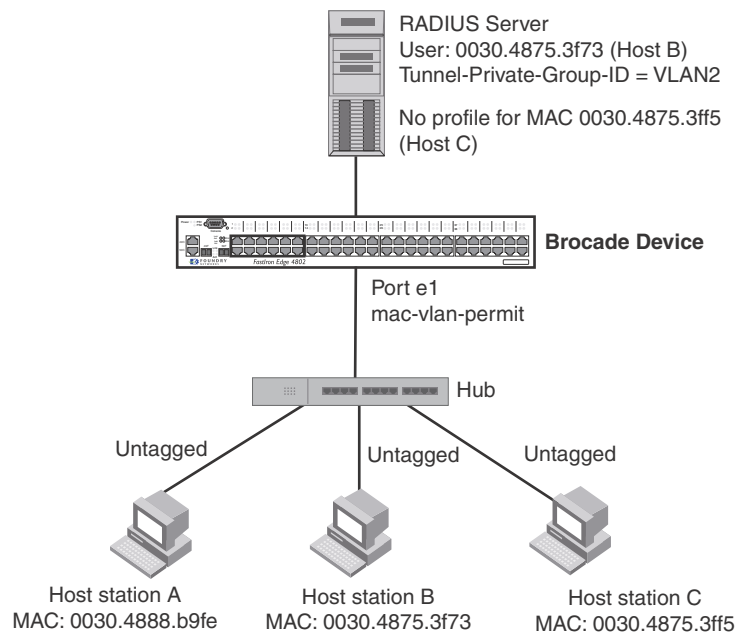
Enter the **clear table-mac-vlan <interface>** command to clear MAC-VLAN information. Add the interface id to clear information for a specific interface.

```
Brocade#clear table-mac-vlan <interface>
```

Sample MAC-based VLAN application

[Figure 109](#) illustrates a sample configuration that uses MAC-based VLAN on port e 0/1/1 on the Brocade device. In this configuration, three host PCs are connected to port e 0/1/1 through a hub.

Host A MAC address is statically configured on port e 0/1/1. The profile for Host B MAC address on the RADIUS server specifies that the PC should be assigned to VLAN 2. Host C profile does not exist in the RADIUS server, and will be put into a restricted VLAN.

FIGURE 109 Sample MAC-based VLAN configuration

Host A MAC address is statically mapped to VLAN 1 with priority 1 and is not subjected to RADIUS authentication. When Host B MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that Host B MAC address be placed into VLAN 2. Since Host C MAC address is not present in the RADIUS server, Host C will be rejected by the server and its MAC address will be placed into a restricted VLAN.

Below is the configuration for this example.

```
module 1 fgs-48-port-management-module
module 2 fgs-xfp-1-cx4-1-port-10g-module
vlan 1 by port
  untagged ethe 0/1/10
  mac-vlan-permit ethe 0/1/1 to 0/1/2
  no spanning-tree
vlan 2 by port
  untagged ethe 0/1/30
  mac-vlan-permit ethe 0/1/1 to 0/1/2
  no spanning-tree
vlan 666 name mac_restricted by port
  untagged ethe 0/1/20
  mac-vlan-permit ethe 0/1/1 to 0/1/2
  no spanning-tree
vlan 4000 name DEFAULT-VLAN by port
  no spanning-tree
vlan 4004 by port
  mac-vlan-permit ethe 0/1/1
default-vlan-id 4000
ip address 10.44.3.8 255.255.255.0
ip default-gateway 10.44.3.1
radius-server host 10.44.3.111
radius-server key 1 $-ndUno
mac-authentication enable
mac-authentication max-age 60
```



```

mac-authentication hw-deny-age 30
mac-authentication auth-passwd-format xxxx.xxxx.xxxx
interface ethernet 0/1/1
  mac-authentication mac-vlan max-mac-entries 5
  mac-authentication mac-vlan 0030.4888.b9fe vlan 1 priority 1
  mac-authentication mac-vlan enable
!
interface ethernet 0/1/2
mac-authentication mac-vlan max-mac-entries 5
mac-authentication mac-vlan enable
!
!
end

```

The **show table-mac-vlan** command returns the following results for all ports in this configuration.

```
Brocade#show table-mac-vlan
```

Port	Vlan	Accepted Mac	Rejected Mac	Attempted Mac	Static Mac	Static Conf	Max Mac
0/1/1	N/A	2	1	0	1	1	5
0/1/2	N/A	0	0	0	0	0	5

The **show table-mac-vlan e 0/1/1** command returns the following results for port 0/1/1 in this configuration.

```
Brocade#show table-mac-vlan e 0/1/1
```

MAC Address	Port	Vlan	Authenticated	Time	Age	CAM Index	MAC Index	Dot1x	Type	Pri
0030.4875.3f73	0/1/1	2	Yes	00d00h00m46s	S32	0001	3728	Dis	Dyn	4
0030.4888.b9fe	0/1/1	1	Yes	00d00h00m08s	Dis	0000	0970	Dis	Sta	1
0030.4875.3ff5	0/1/1	666	Rst	01d18h47m58s	S8	0002	1ee4	Dis	Dyn	0

Sample MAC-based VLAN application

Port mirroring and Monitoring

Table 158 lists the individual Brocade FastIron switches and the mirroring features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 158 Supported port mirroring and monitoring features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Port mirroring and monitoring (mirroring of both inbound and outbound traffic on individual ports)	Yes	Yes	Yes	Yes	Yes
ACL-based mirroring of denied traffic	Yes, on SX-FI48GPP, SX-FI-2XG, SX-FI8XG, SX-FI-24GPP, and SX-FI-24HF modules	Yes	Yes	Yes	Yes
ACL-based mirroring of permitted traffic	Yes	Yes	Yes	Yes	Yes
MAC address filter-based mirroring	No	Yes	Yes	Yes	Yes
VLAN-based mirroring	Supported on third generation SX modules only	Yes	Yes	Yes	Yes

NOTE

FWS, FCX, and ICX devices support only ingress VLAN-based mirroring. FSX devices support both ingress and egress VLAN-based mirroring.

The procedures in this chapter describe how to configure port mirroring on Brocade devices.

Port mirroring and monitoring overview

Port mirroring is a method of monitoring network traffic that forwards a copy of each incoming or outgoing packet from one port on a network switch to another port where the packet can be analyzed. Port mirroring can be used as a diagnostic tool or debugging feature, especially for preventing attacks. Port mirroring can be managed locally or remotely.

Port mirroring and monitoring overview

You can configure port mirroring, by assigning a port (known as the Monitor port), from which the packets are copied and sent to a destination port (known as the Mirror port). All packets received on the Monitor port or issued from it, are forwarded to the second port. You next attach a protocol analyzer on the mirror port to monitor each segment separately. The analyzer captures and evaluates the data without affecting the client on the original port.

The mirror port may be a port on the same switch with an attached RMON probe, a port on a different switch in the same hub, or the switch processor.

Port mirroring and monitoring configuration

To configure port monitoring, first specify the mirror port, then enable monitoring on the monitored port.

The *mirror port* is the port to which the monitored traffic is copied. Attach your protocol analyzer to the mirror port. The *monitored port* is the port with the traffic you want to monitor.

[Table 159](#) lists the number of mirror and monitor ports supported on the Brocade devices. For more information about port regions, refer to [“About port regions”](#) on page 556.

TABLE 159 Number of mirror and monitored ports supported

Port type	Maximum number supported		
	FastIron X Series	FCX	ICX
Ingress mirror ports	1 per port region	1 per port region	1 per port region
Egress mirror ports	1 per port region	1 per port region	1 per port region
Ingress monitored ports	No limit	No limit	No limit
Egress monitored ports	8	8	8

NOTE

FastIron X Series devices support multiple ingress and egress mirror ports. For 1 Gbps ports, ports in groups of 12 share one ingress mirror port and one egress mirror port. Therefore, ports 1 and 2 cannot have different mirror ports, but ports 1 and 13 can (port 25, for third-generation modules). Each 10 Gbps port (or each third-generation 10 GbE module) can have one ingress mirror port and one egress mirror port.

NOTE

For devices other than the FastIron X Series, it is possible to configure more than eight egress ports, although only the first eight are operational. This is also true for mirrored VLANs - more than eight can be configured, but only the first eight are operational.

Configuration notes for port mirroring and monitoring

Refer to the following guidelines when configuring port mirroring and monitoring:

- If you configure both ACL mirroring and ACL-based rate limiting on the same port, then all packets that match are mirrored, including the packets that exceed the rate limit.
- FWS and FCX Series devices support sFlow and port monitoring together on the same port.
- FastIron X Series devices support port monitoring and sFlow together on the same device. The caveat is that port monitoring and sFlow cannot be configured together within the same port region. Refer to [“About port regions”](#) on page 556 for a list of valid port ranges on these devices. This restriction only applies to first- and second-generation modules.
- You can configure a mirror port specifically as an ingress port, an egress port, or both.
- Mirror ports can run at any speed and are not related to the speed of the ingress or egress monitored ports.
- The same port cannot be both a monitored port and the mirror port.

- The same port can be monitored by one mirror port for ingress traffic and another mirror port for egress traffic.
- The mirror port cannot be a trunk port.
- The monitored port and its mirror port do not need to belong to the same port-based VLAN:
 - If the mirror port is in a *different* VLAN from the monitored port, the packets are tagged with the monitor port VLAN ID. This does not apply if the mirror port resides on the SX-FI48GPP module. In this case, mirrored packets are not tagged with a monitor port VLAN ID.
 - If the mirror port is in the *same* VLAN as the monitored port, the packets are tagged or untagged, depending on the mirror port configuration.
- More than one monitored port can be assigned to the same mirror port.
- If the primary interface of a trunk is enabled for monitoring, the entire trunk is monitored. You can also enable an individual trunk port for monitoring using the **config-trunk-ind** command.
- With FastIron X Series IPv4 hardware, the following port mirroring functions may not work across modules when only one switch fabric is present in the system:
 - Input mirroring
 - Output mirroring
 - Both input mirroring and output mirroring

Additional factors that can affect cross-module port mirroring include:

- Switch fabric slot configuration (SF1 or SF2)
- Interface modules configured for port mirroring
- For *stacked* devices, if the ingress and egress analyzer ports are always network ports on the local device, each device may configure the ingress and egress analyzer port independently. However, if you need to mirror to a remote port, then only one ingress and one egress analyzer port are supported for the entire system.
- For ingress ACL mirroring, the ingress rule for stacked devices also applies. The analyzer port setting command **acl-mirror-port** must be specified for each port, even though the hardware only supports one port per device. This applies whether the analyzer port is on the local device or on a remote device. For example, when port mirroring is set to a remote device, any mirroring-enabled ports (ACL, MAC address filter, or VLAN) enabled ports are set globally to a single analyzer port, as shown in the following example.

```
Brocade(config)# mirror ethernet 1/1/24
Brocade(config)# mirror ethernet 2/1/48
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e1000-1/1/1)# monitor ethernet 2/1/48 both
```

The analyzer port (2/1/48) is set to all devices in the system.

```
Brocade(config)# interface ethernet 1/1/2
Brocade(config-if-e1000-1/1/2)# ip access-group 101 in
Brocade(config-if-e1000-1/1/2)# interface ethernet 1/1/1
Brocade(config-if-e1000-1/1/1)# acl-mirror-port ethernet 2/1/48
```

The previous command is required even though the analyzer port is already set globally by the port mirroring command.

```
Brocade(config)# interface ethernet 1/1/3
Brocade(config-if-e1000-1/1/3)# ip access-group 101 in
Brocade(config-if-e1000-1/1/3)# acl-mirror-port ethernet 2/1/48
Brocade(config-if-e1000-1/1/3)# ip access-group 102 in
```

- Because of hardware limitations on the SX-FI48GPP interface module, if the monitored port is on the SX-FI48GPP module, mirrored packets vary slightly from original (monitored) packets, depending on the type of management or interface module on which the mirror port (analyzer) is configured:
 - When ingress or egress mirroring is enabled between a monitored port on an SX-FI48GPP interface module and a mirror port on an IPv4 interface module, mirrored packets contain CRC errors. All other data in the original and mirrored packets is the same.
 - When ingress or egress mirroring is enabled between a monitored port on an SX-FI48GPP interface module and a mirror port on an IPv4 8-port management module, IPv6 management module, IPv6 interface module, or SX-FI48GPP module, mirrored packets are larger than the original packet by 4 bytes, but do not contain CRC errors. All other data in the original and mirrored packets is the same.

Command syntax for port mirroring and monitoring

This section describes how to configure port mirroring and monitoring.

Monitoring a port

To configure port monitoring on an individual port on a Brocade device, enter commands similar to the following.

```
Brocade(config)#mirror-port ethernet 1/2/4
Brocade(config)#interface ethernet 1/2/11
Brocade(config-if-e1000-11)#monitor ethernet 1/2/4 both
```

Traffic on port e 1/2/11 will be monitored, and the monitored traffic will be copied to port e 1/2/4, the mirror port.

Syntax: [no] mirror-port ethernet <port> [input | output]

Syntax: [no] monitor ethernet <port> both | in | out

The <port> variable for **mirror-port ethernet** specifies the port to which the monitored traffic is copied. The <port> variable for **monitor ethernet** specifies the port on which traffic is monitored.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **input** and **output** parameters configure the mirror port exclusively for ingress or egress traffic. If you do not specify one, both types of traffic apply.

The **both**, **in**, and **out** parameters specify the traffic direction you want to monitor on the mirror port. There is no default.

To display the port monitoring configuration, enter the **show monitor** and **show mirror** commands.

Monitoring an individual trunk port

You can monitor the traffic on an individual port of a static trunk group, and on an individual port of an LACP trunk group.

By default, when you monitor the primary port in a trunk group, aggregated traffic for all the ports in the trunk group is copied to the mirror port. You can configure the device to monitor individual ports in a trunk group. You can monitor the primary port or a secondary port individually.

To configure port monitoring on an individual port in a trunk group, enter commands such as the following.

```
Brocade(config)#mirror-port ethernet 2/6
Brocade(config)#trunk e 2/2 to 2/5
Brocade(config-trunk-2/2-2/5)#config-trunk-ind
Brocade(config-trunk-2/2-2/5)#monitor ethe-port-monitored 2/4 ethernet 2/6 in
```

Traffic on trunk port e 2/4 is monitored, and the monitored traffic is copied to port e 2/6, the mirror port.

The **config-trunk-ind** command enables configuration of individual ports in the trunk group. You enter the **config-trunk-ind** command only once in a trunk group. After you enter the command, all applicable port configuration commands apply to individual ports only.

NOTE

If you enter **no config-trunk-ind**, all port configuration commands are removed from the individual ports and the configuration of the primary port is applied to all the ports. Also, once you enter the **no config-trunk-ind** command, the **enable**, **disable**, and **monitor** commands are valid only on the primary port and apply to the entire trunk group.

mirroring configuration on an IronStack

You can configure mirroring on a Brocade IronStack. An IronStack consists of up to eight FCX devices. The stack operates as a chassis. The following examples show how to configure mirroring for ports that are on different members of a stack, and for ports that are on the same stack member as the mirror port.

Configuration notes for Ironstack mirroring

The following mirroring configuration information applies to FCX devices connected in an IronStack topology:

- The input or output mirroring port can be on different ports.
- An IronStack can have one mirroring port that monitors multiple ports, but cannot have multiple mirror ports for one monitored port.
- If the mirror port and the monitored ports are on different stack units, only one active mirror port is allowed for the entire IronStack.
- If the mirror port and the monitored ports are on the same port region, multiple active mirror ports are allowed for the entire IronStack. Devices in an IronStack support 24 ports per port region.

- The maximum number of monitored VLANs on an IronStack is 8.

Configuring mirroring for ports on different members in an IronStack example

In this example, although two ports are configured as active ports, only one active mirror port (port 1/1/24) is allowed for the entire stack because the mirror ports and the monitored ports are on different stack members.

```
Brocade(config)#mirror-port ethernet 1/1/24
Brocade(config)#mirror-port ethernet 2/1/24
Brocade(config)#interface ethernet 1/1/1
Brocade(config-if-e1000-1/1/1)#monitor ethernet 1/1/24 both
Brocade(config)#interface ethernet 2/1/1
Brocade(config-if-e1000-2/1/1)#monitor ethernet 1/1/24 both
Brocade(config)#interface ethernet 4/1/1
Brocade(config-if-e1000-4/1/1)#monitor ethernet 1/1/24 both
```

Configuring mirroring for ports on the same stack member in an IronStack example

In this example, the mirror ports are assigned to different monitor ports.

```
Brocade(config)#mirror-port ethernet 1/1/24
Brocade(config)#mirror-port ethernet 2/1/24
Brocade(config)#mirror-port ethernet 3/1/24
Brocade(config)#mirror-port ethernet 4/1/24
Brocade(config)#interface ethernet 1/1/1
Brocade(config-if-e1000-1/1/1)#monitor ethernet 1/1/24 both
Brocade(config)#interface ethernet 2/1/1
Brocade(config-if-e1000-2/1/1)#monitor ethernet 2/1/24 both
Brocade(config)#interface ethernet 4/1/1
Brocade(config-if-e1000-4/1/1)#monitor ethernet 4/1/24 both
```

ACL-based inbound mirroring

This section describes ACL-based inbound mirroring for FastIron devices.

Creating an ACL-based inbound mirror clause for FWS and FCX devices

The following example shows how to configure an ACL-based inbound mirror clause.

1. Configure the mirror port.

```
Brocade(config)#mirror-port ethernet 1/1/2
```

2. Configure the ACL-based inbound mirror clause.

```
Brocade(config)#access-list 101 permit ip any any mirror
```

3. Apply the ACL-based inbound clause to the monitor port.

```
Brocade(config)#interface e 1/1/5
Brocade(config-if-e1000-1/1/5)#ip access-group 101 in
```

4. Create the ACL mirror port.

```
Brocade(config-if-e1000-1/1/5)#acl-mirror-port ethernet 1/1/2
```

To display ACL mirror settings, enter the **show access-list all** command.

```
Brocade#show access-list all
Extended IP access list 101
permit ip any any mirror
```

ACL-based inbound mirror clauses for FastIron X Series devices

The **mirror** parameter in an ACL clause causes the system to direct traffic that meets the clause to be sent to a mirror port. Consider the following example.

```
Brocade(config)#access-list 101 permit ip any any mirror
```

The **mirror** parameter directs selected traffic to the mirrored port. Traffic mirroring is only supported on Layer 3 ACLs for FastIron X Series devices.

You can select traffic to be mirrored using a **permit** or **deny** clause on ports on the following interface modules:

- SX-FI-24GPP
- SX-FI-24HF
- SX-FI-2XG
- SX-FI-8XG

On all other interface modules, you can select traffic to be mirrored using only a **permit** clause.

Destination mirror port

You can specify physical ports or a trunk to mirror traffic. If you complete the rest of the configuration but do not specify a destination mirror port, the port-mirroring ACL is non-operational. This can be useful if you want to be able to mirror traffic by a set criteria on demand. With this configuration, you configure a destination mirror port whenever you want the port-mirroring ACL to become operational.

The following sections describe how to specify a destination port for a port or a trunk, as well as the special considerations required when mirroring traffic from a virtual interface.

Specifying the destination mirror port for physical ports

When you want traffic that has been selected by ACL-based inbound mirroring to be mirrored, you must configure a destination mirror port. This configuration is performed at the interface configuration level of the port with the traffic you are mirroring. The destination port must be the same for all ports in a port region as described in [“Ports from a port region must be mirrored to the same destination mirror port”](#) on page 933.

In the following example, ACL mirroring traffic from port 1/1 is mirrored to port 1/3.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e1000-1/1)#ACL-mirror-port ethernet 1/3
```

Syntax: [no] **ACL-mirror-port ethernet <port>**

The **<port>** variable specifies the mirror port to which the monitored port traffic is copied.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Ports from a port region must be mirrored to the same destination mirror port

Port regions, as described in “[About port regions](#)” on page 556, are important when defining a destination mirror port. This is because all traffic mirrored from any single port in a port region is mirrored to the same destination mirror port as traffic mirrored from any other port in the same port region. For example, ports 1/1 to 1/12 are in the same port region. If you configure ports 1/1 and 1/2 to mirror their traffic, they should use the same destination mirror port as shown in the following configuration.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#ACL-mirror-port ethernet 2/3
Brocade(config)#interface ethernet 1/2
Brocade(config-if-e10000-1/2)#ACL-mirror-port ethernet 2/3
```

If ports within the same port region are mirrored to different destination ports, the configuration is disallowed, and an error message is generated, as shown in the following example.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#ACL-mirror-port ethernet 4/3
Brocade(config)#interface ethernet 1/2
Brocade(config-if-e10000-1/2)#ACL-mirror-port ethernet 4/7
Error - Inbound Mirror port 4/3 already configured for port region 1/1 - 1/12
```

When a destination port is configured for any port within a port region, traffic from any ACL with a mirroring clause assigned to any port in that port region is mirrored to that destination port. This will occur even if a destination port is not explicitly configured for the port with the ACL configured. In the following example, an ACL with a mirroring clause (101) is applied to a port (1/1). Another port in the same region (1/3) has a destination port set (4/3). In this example, traffic generated from operation of ACL 101 is mirrored to port 4/3 even though a destination port has not explicitly been defined for traffic from port 1/1.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#ip access-group 101 in
Brocade(config)#interface ethernet 1/3
Brocade(config-if-e10000-1/3)#ACL-mirror-port ethernet 4/3
```

NOTE

If a destination mirror port is not configured for any ports within the port region where the port-mirroring ACL is configured, the ACL does not mirror the traffic but the ACL is applied to traffic on the port.

Specifying the destination mirror port for trunk ports

You can mirror the traffic that has been selected by ACL-based inbound mirroring from a trunk by configuring a destination port for the primary port within the trunk configuration, as shown in the following example.

```
Brocade(config)#trunk ethernet 1/1 to 1/4
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#ACL-mirror-port ethernet 1/8
```

Using this configuration, all trunk traffic is mirrored to port 1/8.

Limitations when configuring ACL-based mirroring with trunks

The **config-trunk-ind** command, as described in [“Disabling or re-enabling a trunk port”](#) on page 712, cannot operate with ACL-based mirroring:

- If a trunk is configured with the **config-trunk-ind** command, ACL-based mirroring will not be allowed.
- If the **config-trunk-ind** command is added to a trunk, any ports that are configured for ACL-based mirroring will have monitoring removed and the following message is displayed.

Trunk port monitoring, if any, has been removed.

If an individual port is configured for ACL-based mirroring, you cannot add it to a trunk. If you try to add a port that is configured for ACL-based mirroring to a trunk, the following message appears.

Note - ACL-mirror-port configuration is removed from port 2 in new trunk.

NOTE

If you want to add a port configured for ACL-based mirroring to a trunk, you must first remove the **ACL-mirror-port** command from the port configuration. You can then add the port to a trunk that can then be configured for ACL-based trunk mirroring.

Behavior of ACL-based mirroring when deleting trunks

If you delete a trunk that has ACL-based mirroring configured, the ACL-based mirroring configuration is configured on the individual ports that made up the trunk.

For example, if a trunk is configured as shown in the following example and is then deleted from the configuration as shown, each of the ports that previously was contained in the trunk is configured for ACL-based mirroring.

```
Brocade(config)#trunk ethernet 4/1 to 4/2
Brocade(config)#trunk deploy
Brocade(config)#interface ethernet 4/1
Brocade(config-if-e10000)#ACL-mirror-port ethernet 5/3
```

To delete the trunk, enter the following command.

```
Brocade(config)#no trunk ethernet 4/1 to 4/2
```

The following configuration for ACL-based mirroring on ports 4/1 and 4/2 results from the trunk being deleted.

```
interface ethernet 4/1
  ACL-mirror-port ethernet 5/3
interface ethernet 4/2
  ACL-mirror-port ethernet 5/3
```

Configuring ACL-based mirroring for ACLs bound to virtual interfaces

For configurations that have an ACL configured for ACL-based mirroring bound to a virtual interface, you must use the **ACL-mirror-port** command on a physical port that is a member of the same VLAN as the virtual interface. Additionally, only traffic that arrives at ports that belong to the same port group as the physical port where the **ACL-mirror-port** command has been used is mirrored. This follows the same rules described in [“Ports from a port region must be mirrored to the same destination mirror port”](#) on page 933.

For example, in the following configuration, ports 4/1, 4/2, and 5/3 are in VLAN 10 with ve 10. Ports 4/1 and 4/2 belong to the same port group, while port 5/3 belongs to another port group.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#tagged ethernet 4/1 to 4/2
Brocade(config-vlan-10)#tagged ethernet 5/3
Brocade(config-vlan-10)#router-interface ve 10

Brocade(config)#interface ethernet 4/1
Brocade(config-if-e10000-4/1)#ACL-mirror-port ethernet 5/1
Brocade(config)#interface ve 10
Brocade(config-vif-10)#ip address 10.10.10.254/24
Brocade(config-vif-10)#ip access-group 102 in
Brocade(config)#access-list 102 permit ip any any mirror
```

In this configuration, the **ACL-mirror-port** command is applied to port 4/1, which is a member of ve 10. Because of this, ACL-based mirroring will only apply to VLAN 10 traffic that arrives on ports 4/1 and 4/2. It will not apply to VLAN 10 traffic that arrives on port 5/3 because that port belongs to a port group different from ports 4/1 and 4/2. This is because if you apply ACL-based mirroring on an entire VE, and enable mirroring in only one port region, traffic that is in the same VE but on a port in a different port region will not be mirrored.

To make the configuration apply ACL-based mirroring to VLAN 10 traffic arriving on port 5/3, you must add the following commands to the configuration.

```
Brocade(config)#interface ethernet 5/3
Brocade(config-if-e10000-5/3)#ACL-mirror-port ethernet 5/1
```

If a port is in both mirrored and non-mirrored VLANs, only traffic on the port from the mirrored VLAN is mirrored. For example, the following configuration adds VLAN 20 to the previous configuration. In this example, ports 4/1 and 4/2 are in both VLAN 10 and VLAN 20. ACL-based mirroring is only applied to VLAN 10. Consequently, traffic that is on ports 4/1 and 4/2 that belongs to VLAN 20 will not be mirrored.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#tagged ethernet 4/1 to 4/2
Brocade(config-vlan-10)#tagged ethernet 5/3
Brocade(config-vlan-10)#router-interface ve 10

Brocade(config)#vlan 20
Brocade(config-vlan-20)#tagged ethernet 4/1 to 4/2

Brocade(config)#interface ethernet 4/1
Brocade(config-if-e10000-4/1)#ACL-mirror-port ethernet 5/1
Brocade(config)#interface ve 10
Brocade(config-vif-10)#ip address 10.10.10.254/24
Brocade(config-vif-10)#ip access-group 102 in
Brocade(config)#access-list 102 permit ip any any mirror
```

MAC address filter-based mirroring

NOTE

The MAC address filter-based mirroring feature is not supported on FastIron X Series devices.

This feature allows traffic entering an ingress port to be monitored from a mirror port connected to a data analyzer, based on specific source and destination MAC addresses. This feature supports mirroring of inbound traffic only. Outbound mirroring is not supported.

MAC-filter-based mirroring allows a user to specify a particular stream of data for mirroring using a filter. This eliminates the need to analyze all incoming data to the monitored port. To configure MAC-filter-based mirroring, the user must perform three steps:

1. Define a mirror port
2. Create a MAC address filter with a mirroring clause
3. Apply the MAC address filter to an interface

Configuring MAC address filter-based mirroring

Complete the following steps to configure MAC address filter-based mirroring.

1. Defining a mirror port

To activate mirroring on a port, use the **mirror** command in global configuration mode.

```
Brocade(config)#mirror e 0/1/14
```

Configuration notes for defining a mirror port

- If there is no input mirror port configured, MAC-filter based mirroring does not take effect. It remains in the configuration, but is not activated.
- FWS devices support one ingress mirror and one egress mirror per system. These ports are shared by all mirroring features: port-based mirroring, VLAN-based mirroring, ACL-based mirroring and MAC-based mirroring.
- MAC-filter-based mirroring can be enabled on a port at the same time as either port-based mirroring or VLAN-based mirroring. When port-based mirroring and MAC-filter-based mirroring are enabled on a port at the same time, the preference order is port-based mirroring followed by MAC-based filtering. When VLAN-based mirroring and MAC-filter-based mirroring are enabled on a port at the same time, the preference order is VLAN-based mirroring and MAC-filter-based mirroring.

NOTE

Port-based mirroring and VLAN-based mirroring can not be enabled on a port at the same time.

2. Creating a MAC address filter with a mirroring clause

The **mirror** keyword is added to MAC address filter clauses to direct desired traffic to the mirror port. In the following example, the MAC address filter directs traffic to a mirror port.

```
Brocade(config)#mac filter 1 permit 0000.1111.2222 ffff.ffff.ffff 0000.2222.3333
fff.fff.fff mirror
```

In this example, any flow matching the source address (SA) 0000.1111.2222 and the destination address (DA) 0000.2222.3333 is mirrored. Other flows are not mirrored.

3. Applying the MAC address filter to an interface

Apply the MAC address filter to an interface using the **mac-filter-group** command.

```
Brocade(config)#interface ethernet 0/1/1
Brocade(config-if-e10000-0/1/1)#mac filter-group 1
```

4. Configuring the monitor port to use the mirror port

```
Brocade(config)#interface ethernet 0/1/5
Brocade(config-if-e10000-0/1/5)#acl-mirror-port ethernet 0/1/14
```

VLAN-based mirroring

NOTE

VLAN-based mirroring is supported on FastIron X Series devices equipped with third generation or later modules. Refer to [“VLAN-based mirroring on FastIron X Series devices”](#) on page 939.

The VLAN-based mirroring feature allows users to monitor all incoming traffic in one or more VLANs by sending a mirror image of that traffic to a configured mirror port. This feature meets the requirements of CALEA (Communications Assistance for Law Enforcement Act of 1994).

Configuring VLAN-based mirroring

Configure VLAN-based mirroring using the **monitor ethernet** command in VLAN configuration mode. For example, to enable mirroring on VLANs 10 and 20, to mirror port e 1/1/21, enter the following commands.

```
Brocade(config)#mirror-port ethernet 1/1/21 input
Brocade(config)#vlan 10
Brocade(config-VLAN-10)#monitor ethernet 1/1/21
Brocade(config-VLAN-10)#exit
Brocade(config)#vlan 20
Brocade(config-VLAN-20)#monitor ethernet 1/1/21
Brocade(config-VLAN-20)#end
```

Syntax: [no] monitor ethernet <port>

NOTE

For FCX devices, because it is possible to have multiple mirror ports, monitor ports must specify which mirror port they are monitoring.

To disable mirroring on VLAN 20, enter the following commands.

```
Brocade(config)#vlan 20
Brocade(config-VLAN-20)#no monitor ethernet 1/1/21
Brocade(config-VLAN-20)#end
```

Displaying VLAN-based mirroring status

The **show vlan** command displays the VLAN-based mirroring status.

VLAN-based mirroring

```
Brocade#show vlan
Total PORT-VLAN entries: 4
Maximum PORT-VLAN entries: 4060

Legend: [Stk=Stack-Unit, S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree On
Untagged Ports: (Stk0/S1)  3  4  5  6  7  8  9 10 11 12 13 14
Untagged Ports: (Stk0/S1) 15 16 17 18 19 20 21 22 23 24 25 26
Untagged Ports: (Stk0/S1) 27 28 29 30 31 32 33 34 35 36 37 38
Untagged Ports: (Stk0/S1) 39 40 41 42 43 44 45 46 47 48
Untagged Ports: (Stk0/S2)  1  2
Tagged Ports: None
Uplink Ports: None
DualMode Ports: None
Mac-Vlan Ports: None
Monitoring: Disabled
PORT-VLAN 10, Name [None], Priority level0, Spanning tree On
Untagged Ports: (Stk0/S1)  1
Tagged Ports: None
Uplink Ports: None
DualMode Ports: None
Mac-Vlan Ports: None
Monitoring: Enabled
PORT-VLAN 20, Name [None], Priority level0, Spanning tree On
Untagged Ports: (Stk0/S1)  2
Tagged Ports: None
Uplink Ports: None
DualMode Ports: None
Mac-Vlan Ports: None
Monitoring: Disabled
```

Configuration notes for VLAN-based mirroring

The following guidelines apply to VLAN-based mirroring configurations:

- A VLAN must have at least one port member configured before monitoring can be configured.
- Multiple VLANs can have monitoring enabled at the same time, and the maximum number of monitor-configured VLANs is 8.
- FWS devices support one ingress and one egress mirror per system. These mirror ports are shared by all mirroring features; port-based mirroring, VLAN-based mirroring, ACL-based mirroring, and MAC-filter-based mirroring.
- The mirror port is subject to the same scheduling and bandwidth management as the other ports in the system. If the amount of traffic being sent to the mirror port exceeds the available bandwidth, some of that traffic may be dropped.
- All incoming traffic (tagged and untagged) in the VLAN is mirrored. mirroring is “as-is”, and is not affected by the configuration of the mirror port itself. Incoming tagged traffic is sent out tagged and incoming untagged traffic is sent out untagged, regardless of which VLANs the mirror port belongs to, and whether the mirror port is tagged or untagged.
- VLAN-based mirroring is supported on Layer 2 and Layer 3 images.

VLAN-based mirroring on FastIron X Series devices

With the new FastIron X Series of modules, the sFlow processing has been separated from the packet mirroring functionality. This allows for support of VLAN-based mirroring on the FastIron X Series devices. The packet processor on the FastIron X Series of modules also allows for egress VLAN-based mirroring. In order for VLAN-based monitoring to function, the FastIron X Series must have only the following SX modules installed.

The following interface modules are new to the FastIron X Series:

- SX-FI-24GPP
- SX-FI-24HF
- SX-FI-2XG
- SX-FI-8XG
- SX-FI48GPP

NOTE

Egress VLAN-based mirroring is not currently supported on the stacking platforms.

The FastIron X Series of modules are capable of monitoring 4096 VLANs. In a chassis environment, this introduces restrictions to the number of ports that can be configured as mirror ports. Because a single VLAN can contain 384 untagged ports (24 per slot) if that VLAN is configured for monitoring, every device must have an identical number of corresponding analyzer ports. However, the egress mirror-port and ingress mirror-port do not have to be the same. You can use two separate ports.

This introduces restrictions on port-based mirroring coexisting with VLAN-based mirroring. Port-based mirroring allows for multiple ports to be configured as mirror-ports. However, once a particular port belonging to a particular device is configured for monitoring to a specific mirror-port, no other mirror-port can be used to monitor any other port on that device. This restriction has been extended to VLAN-based mirroring, with one caveat: only one mirror-port in either direction at a time can exist within the system. Refer to [“About port regions”](#) on page 556 for a list of valid port ranges on these devices.

Restrictions and capabilities of VLAN-based mirroring

The following is a list of restrictions and capabilities:

- Only the modules that support VLAN-based mirroring should be installed.
- There can be only one input or output mirror-port configured in the system at a time.
- The amount of traffic mirrored is limited by the bandwidth of the mirror-port.
- The maximum amount of egress traffic that can be mirrored is further limited by the bandwidth of the loopback port, which is 10 Gbps.
- The monitored VLAN must be created in hardware.
- An ingress or egress mirror-port must be configured when monitoring the ingress or egress VLAN traffic.
- A maximum of 4096 VLANs can be monitored at a time.
- A VLAN can be monitored for ingress and egress traffic concurrently.
- Port mirroring can be configured concurrently with VLAN-based mirroring, but only one mirror-port can be used for both.

- sFlow can be enabled concurrently with VLAN-based mirroring and port mirroring.
- VLAN-based mirroring is supported on the default VLAN. If the default VLAN is changed dynamically, the configuration is not lost.
- VLAN-based mirroring on VLAN groups is not supported, but it is supported on topology groups.
- In the case of enabling VLAN-based monitoring on the interface modules in an MCT-enabled chassis, the VLAN configuration is not synced across the cluster. Each chassis in the cluster is configured independently for VLAN configuration.

One of the concerns about VLAN-based mirroring is the effects of ingress and egress ACLs, as well as rate shaping and rate limiting, on mirrored packets:

- Ingress VLAN-based mirroring: Any packets that are coming in from the network on the VLAN should be mirrored out. Any ingress ACL actions or rate limiting actions do not take precedence in this case.
- Egress VLAN-based mirroring: Any packets that are sent out onto the network are not affected by egress ACLs or rate shaping.

Refer to Table 160 for a summary of the effects of ACLs and rate limiting.

TABLE 160 ACL and rate limiting effects

ACL profile	Ingress result	Egress result
Ingress ACL on port	Packets ingress mirrored at expected (sent) rate	Packets egress mirrored at expected (sent) rate
Egress ACL on port	Packets ingress mirrored at expected (sent) rate	Packets egress mirrored at expected (sent) rate
Ingress rate limiting on port	Packets ingress mirrored at expected (sent) rate	Packets egress mirrored at the limited rate
Egress rate shaping on port	Packets ingress mirrored at expected (sent) rate	Packets egress mirrored at expected (sent) rate

Tagged versus untagged ports in VLANs

Table 161 describes how VLAN-based mirroring coexists with port mirroring, assuming a VLAN consisting of one ingress and one egress port.

TABLE 161 VLAN-based mirroring and port mirroring effects

Packets sent	VLAN-based mirroring direction	Which port in VLAN is monitored	Ingress mirror traffic count expectation	Egress mirror traffic count expectation
10000	Ingress	Ingress	~10000	~0
10000	Egress	Egress	~0	~20000
10000	Ingress	Egress	~10000	~10000
10000	Egress	Ingress	~10000	~10000

Table 161 can be summarized into the following two rules, assuming that VLAN-based mirroring and port mirroring are operating concurrently.

- If the VLAN is ingress monitored, and ports belonging to the VLAN are also ingress monitored, the ingress traffic is only mirrored once and there are no duplicated mirrored packets.

- If the VLAN is egress monitored and ports belonging to the VLAN are also egress monitored, the egress traffic is mirrored for each egress port, as well as the VLAN, resulting in several duplicated mirrored packets. The count of duplicate packets is computed as (1 + Number of egress mirrored ports in the VLAN) * Number of egress packets.

In addition to the previously mentioned rules, the behavior of VLAN-based mirroring is affected by the tag type of the ports belonging to that VLAN, as shown in [Table 162](#).

TABLE 162 VLAN-based mirroring behavior: Tagged versus untagged ports

Ingress tag type	Egress tag type	VLAN-based mirroring direction	Mirrored traffic tag type
Untagged	Untagged	Ingress	Untagged
Untagged	Untagged	Egress	Tagged
Untagged	Tagged	Ingress	Untagged
Untagged	Tagged	Egress	Tagged
Tagged	Untagged	Ingress	Tagged
Tagged	Untagged	Egress	Tagged
Tagged	Tagged	Ingress	Tagged
Tagged	Tagged	Egress	Tagged

As illustrated in [Table 162](#), regardless of the egress port tag type, if a VLAN is egress mirrored, the mirrored traffic is always tagged. This functionality is built into the hardware and is the expected behavior, as the tag acts as an identifier for monitored traffic on the mirror-port (in case the mirror-port belongs to some other VLAN).

Configuring VLAN-based mirroring on FastIron X Series devices

Configure VLAN-based mirroring using the **monitor ethernet** command in VLAN configuration mode. For example, to enable incoming mirroring on VLANs 10 and 20, to mirror port e 6/24, enter the following commands.

```
Brocade(config)#mirror-port ethernet 6/24
Brocade(config)#vlan 10
Brocade(config-VLAN-10)#monitor ethernet 6/24 input
Brocade(config-VLAN-10)#exit
Brocade(config)#vlan 20
Brocade(config-VLAN-20)#monitor ethernet 6/24 input
Brocade(config-VLAN-20)#end
```

For example, to enable outgoing mirroring on VLANs 10 and 20, to mirror port e 6/24, enter the following commands.

```
Brocade(config)#mirror-port ethernet 6/24
Brocade(config)#vlan 10
Brocade(config-VLAN-10)#monitor ethernet 6/24 output
Brocade(config-VLAN-10)#exit
Brocade(config)#vlan 20
Brocade(config-VLAN-20)#monitor ethernet 6/24 output
Brocade(config-VLAN-20)#end
```

For example, to enable bidirectional mirroring on VLANs 10 and 20, to mirror port e 6/24, enter the following commands.

```
Brocade(config)#mirror-port ethernet 6/24
```

VLAN-based mirroring

```
Brocade(config)#vlan 10
Brocade(config-VLAN-10)#monitor ethernet 6/24 both
Brocade(config-VLAN-10)#exit
Brocade(config)#vlan 20
Brocade(config-VLAN-20)#monitor ethernet 6/24 both
Brocade(config-VLAN-20)#end
```

To disable mirroring on VLAN 20, enter the following commands.

```
Brocade(config)#vlan 20
Brocade(config-VLAN-20)#no monitor ethernet 6/24
Brocade(config-VLAN-20)#end
```

Displaying VLAN-based mirroring status

The **show vlan** command displays the VLAN-based mirroring status.

```
Brocade(config-if-e1000-5/2)#show vlan 10
```

```
Total PORT-VLAN entries: 2
```

```
Maximum PORT-VLAN entries: 64
```

```
Legend: [Stk=Stack-Id, S=Slot]
```

```
PORT-VLAN 10, Name To-SXR1600, Priority level0, Spanning tree Off
```

```
Untagged Ports: (S5) 2 4
```

```
Tagged Ports: None
```

```
Uplink Ports: None
```

```
DualMode Ports: None
```

```
RX Monitoring: Disabled
```

```
TX Monitoring: Enabled
```

Syntax: [no] monitor ethernet <port> [input | output | both]

IP Configuration

Table 163 lists the individual Brocade FastIron switches and the IP features they support. These features are supported with the base Layer 3, edge Layer 3 and full Layer 3 software image, except where explicitly noted.

TABLE 163 Supported IP features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
BootP/DHCP relay	Yes	Yes	Yes	Yes	ICX 6450 only
Specifying which IP address will be included in a DHCP/BootP reply packet	Yes	No	Yes	Yes	Yes
DHCP Server	Yes	Yes	Yes	Yes	Yes
DHCP Client-Based Auto-Configuration	Yes	Yes	Yes	Yes	Yes
DHCP Client-Based Flash image Auto-update	Yes	Yes	Yes	Yes	Yes
DHCP assist	Yes	Yes	Yes	Yes	Yes
Equal Cost Multi Path (ECMP) load sharing	Yes	Yes	Yes	Yes	ICX 6450 only
IP helper	Yes	Yes	Yes	Yes	ICX 6450 only
Single source address for the following packet types: <ul style="list-style-type: none"> • Telnet • TFTP • Syslog • SNTP • TACACS/TACACS+ • RADIUS • SSH • SNMP 	Yes	No	No	No	No
IPv4 point-to-point GRE IP tunnels	Yes (IPv6 devices only)	No	Yes	Yes	No
Routes in hardware maximum: <ul style="list-style-type: none"> • FESX6 – Up to 256K routes • FSX – Up to 512K routes • FCX – Up to 16K routes • FWS – Up to 1020 routes • ICX 6450 - Up to 12,000 routes 	Yes	Yes	Yes	Yes	ICX 6450 only

TABLE 163 Supported IP features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Routing for directly connected IP subnets	Yes	Yes	Yes	Yes	ICX 6450 only
Virtual Interfaces: • Up to 512 virtual interfaces	Yes	Yes	Yes	Yes	ICX 6450 only, up to 255
31-bit subnet mask on point-to-point networks	Yes on devices running full Layer 3 image	No	Yes on devices running full Layer 3 image	Yes on devices running full Layer 3 image	No
Address Resolution Protocol (ARP)	Yes	Yes	Yes	Yes	ICX 6450 only
Reverse Address Resolution Protocol (RARP)	Yes	Yes	Yes	Yes	ICX 6450 only
IP follow	Yes	Yes	Yes	Yes	ICX 6450 only
Proxy ARP	Yes	Yes	Yes	Yes	ICX 6450 only
Local proxy ARP	Yes	Yes	Yes	Yes	ICX 6450 only
Jumbo frames • Up to 10,240 bytes, or • Up to 10,232 bytes in an IronStack	Yes	Yes	Yes	Yes	Yes
IP MTU (individual port setting)	Yes	No	Yes	Yes	ICX 6450 only
Path MTU discovery	Yes	No	Yes	Yes	No
ICMP Router Discovery Protocol (IRDP)	Yes	Yes	Yes	Yes	ICX 6450 only
Domain Name Server (DNS) resolver	Yes	Yes	Yes	Yes	Yes
IP checksum check disable	Yes	No	No	No	No

NOTE

References to chassis-based Layer 3 Switches apply to the FSX 800 and FSX 1600.

NOTE

The terms Layer 3 Switch and router are used interchangeably in this chapter and mean the same.

Basic IP configuration

IP is enabled by default. Basic configuration consists of adding IP addresses for Layer 3 Switches, enabling a route exchange protocol, such as the Routing Information Protocol (RIP).

If you are configuring a Layer 3 Switch, refer to [“Configuring IP addresses”](#) on page 964 to add IP addresses, then enable and configure the route exchange protocols, as described in other chapters of this guide.

If you are configuring a Layer 2 Switch, refer to [“Configuring the management IP address and specifying the default gateway”](#) on page 1037 to add an IP address for management access through the network and to specify the default gateway.

The rest of this chapter describes IP and how to configure it in more detail. Use the information in this chapter if you need to change some of the IP parameters from their default values or you want to view configuration information or statistics.

IP configuration overview

Brocade Layer 2 Switches and Layer 3 Switches support Internet Protocol version 4 (IPv4) and IPv6. IP support on Brocade Layer 2 Switches consists of basic services to support management access and access to a default gateway.

Edge Layer 3 support

NOTE

Edge Layer 3 images are supported on FWS-EPREM and FWSG-EPREM models.

IP support on Brocade edge Layer 3 Switches includes support for all of the base Layer 3 features, in addition to the following:

- OSPF V2 (IPv4)
- Full RIP V1 and V2
- Route-only support (Global configuration level only)
- Route redistribution
- 1020 routes in hardware maximum
- VRRP-E

Full Layer 3 support

NOTE

Full Layer 3 images are supported on FastIron X Series -PREM and Brocade FCX Series devices only.

IP support on Brocade full Layer 3 Switches includes all of the following, in addition to a highly configurable implementation of basic IP services including Address Resolution Protocol (ARP), ICMP Router Discovery Protocol (IRDP), and Reverse ARP (RARP):

- Route exchange protocols:
 - Routing Information Protocol (RIP)
 - Open Shortest Path First (OSPF)
 - Border Gateway Protocol version 4 (BGP4)

- Multicast protocols:
 - Internet Group Membership Protocol (IGMP)
 - Protocol Independent Multicast Dense (PIM-DM)
 - Protocol Independent Multicast Sparse (PIM-SM)
 - Distance Vector Multicast Routing Protocol (DVMRP)
- Router redundancy protocols:
 - Virtual Router Redundancy Protocol Extended (VRRP-E)
 - Virtual Router Redundancy Protocol (VRRP)

IP interfaces

NOTE

This section describes IPv4 addresses. For information about IPv6 addresses on FastIron X Series devices, refer to [“IPv6 addressing overview”](#) on page 355. For information about IPv6 addresses on all other FastIron devices, refer to [“IPv6 addressing”](#) on page 412.

Brocade Layer 3 Switches and Layer 2 Switches allow you to configure IP addresses. On Layer 3 Switches, IP addresses are associated with individual interfaces. On Layer 2 Switches, a single IP address serves as the management access address for the entire device.

All Brocade Layer 3 Switches and Layer 2 Switches support configuration and display of IP addresses in classical subnet format (for example: 192.168.1.1 255.255.255.0) and Classless Interdomain Routing (CIDR) format (for example: 192.168.1.1/24). You can use either format when configuring IP address information. IP addresses are displayed in classical subnet format by default but you can change the display format to CIDR. Refer to [“Changing the network mask display to prefix format”](#) on page 1065.

Layer 3 Switches

Brocade Layer 3 Switches allow you to configure IP addresses on the following types of interfaces:

- Ethernet ports
- Virtual routing interfaces (used by VLANs to route among one another)
- Loopback interfaces

Each IP address on a Layer 3 Switch must be in a different subnet. You can have only one interface that is in a given subnet. For example, you can configure IP addresses 192.168.1.1/24 and 192.168.2.1/24 on the same Layer 3 Switch, but you cannot configure 192.168.1.1/24 and 192.168.1.2/24 on the same Layer 3 Switch.

You can configure multiple IP addresses on the same interface.

The number of IP addresses you can configure on an individual interface depends on the Layer 3 Switch model. To display the maximum number of IP addresses and other system parameters you can configure on a Layer 3 Switch, refer to [“Displaying and modifying system parameter default settings”](#) on page 578.

You can use any of the IP addresses you configure on the Layer 3 Switch for Telnet, Web management, or SNMP access.

Layer 2 Switches

You can configure an IP address on a Brocade Layer 2 Switch for management access to the Layer 2 Switch. An IP address is required for Telnet access, Web management access, and SNMP access.

You also can specify the default gateway for forwarding traffic to other subnets.

IP packet flow through a Layer 3 Switch

Figure 110 shows how an IP packet moves through a Brocade Layer 3 Switch.

FIGURE 110 IP Packet flow through a Brocade Layer 3 Switch

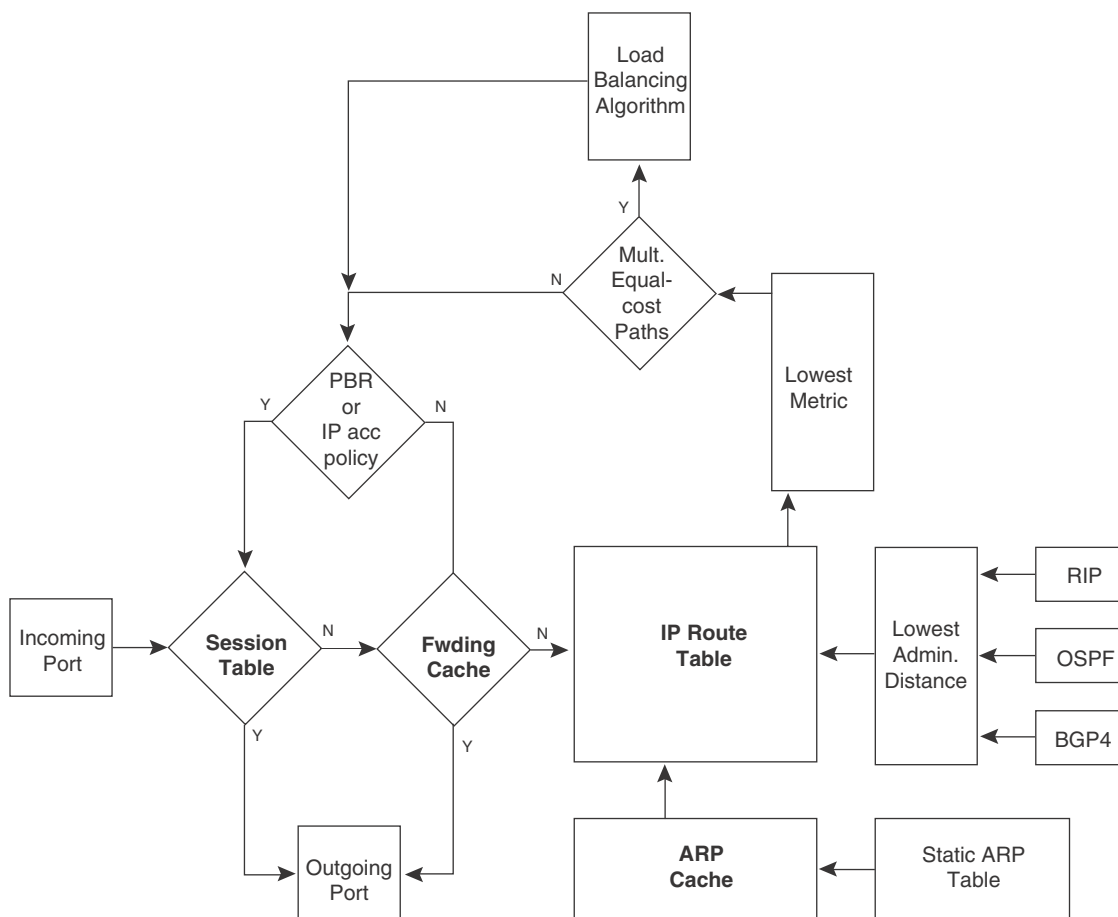


Figure 110 shows the following packet flow:

1. When the Layer 3 Switch receives an IP packet, the Layer 3 Switch checks for filters on the receiving interface.¹ If a deny filter on the interface denies the packet, the Layer 3 Switch discards the packet and performs no further processing, except generating a Syslog entry and SNMP message, if logging is enabled for the filter.
2. If the packet is not denied at the incoming interface, the Layer 3 Switch looks in the session table for an entry that has the same source IP address and TCP or UDP port as the packet. If the session table contains a matching entry, the Layer 3 Switch immediately forwards the packet, by addressing it to the destination IP address and TCP or UDP port listed in the session table entry and sending the packet to a queue on the outgoing ports listed in the session table. The Layer 3 Switch selects the queue based on the Quality of Service (QoS) level associated with the session table entry.
3. If the session table does not contain an entry that matches the packet source address and TCP or UDP port, the Layer 3 Switch looks in the IP forwarding cache for an entry that matches the packet destination IP address. If the forwarding cache contains a matching entry, the Layer 3 Switch forwards the packet to the IP address in the entry. The Layer 3 Switch sends the packet to a queue on the outgoing ports listed in the forwarding cache. The Layer 3 Switch selects the queue based on the Quality of Service (QoS) level associated with the forwarding cache entry.
4. If the IP forwarding cache does not have an entry for the packet, the Layer 3 Switch checks the IP route table for a route to the packet destination. If the IP route table has a route, the Layer 3 Switch makes an entry in the session table or the forwarding cache, and sends the route to a queue on the outgoing ports:
 - If the running-config contains an IP access policy for the packet, the software makes an entry in the session table. The Layer 3 Switch uses the new session table entry to forward subsequent packets from the same source to the same destination.
 - If the running-config does not contain an IP access policy for the packet, the software creates a new entry in the forwarding cache. The Layer 3 Switch uses the new cache entry to forward subsequent packets to the same destination.

The following sections describe the IP tables and caches:

- ARP cache and static ARP table
- IP route table
- IP forwarding cache
- Layer 4 session table

The software enables you to display these tables. You also can change the capacity of the tables on an individual basis if needed by changing the memory allocation for the table.

ARP cache and static ARP table

The ARP cache contains entries that map IP addresses to MAC addresses. Generally, the entries are for devices that are directly attached to the Layer 3 Switch.

An exception is an ARP entry for an interface-based static IP route that goes to a destination that is one or more router hops away. For this type of entry, the MAC address is either the destination device MAC address or the MAC address of the router interface that answered an ARP request on behalf of the device, using proxy ARP.

1. The filter can be an Access Control List (ACL) or an IP access policy.

ARP cache

The ARP cache can contain dynamic (learned) entries and static (user-configured) entries. The software places a dynamic entry in the ARP cache when the Layer 3 Switch learns a device MAC address from an ARP request or ARP reply from the device.

The software can learn an entry when the Layer 2 Switch or Layer 3 Switch receives an ARP request from another IP forwarding device or an ARP reply. Here is an example of a dynamic entry:

	IP Address	MAC Address	Type	Age	Port
1	207.95.6.102	0800.5afc.ea21	Dynamic	0	6

Each entry contains the destination device IP address and MAC address.

Static ARP table

In addition to the ARP cache, Layer 3 Switches have a static ARP table. Entries in the static ARP table are user-configured. You can add entries to the static ARP table regardless of whether or not the device the entry is for is connected to the Layer 3 Switch.

NOTE

Layer 3 Switches have a static ARP table. Layer 2 Switches do not.

The software places an entry from the static ARP table into the ARP cache when the entry interface comes up.

Here is an example of a static ARP entry.

Index	IP Address	MAC Address	Port
1	207.95.6.111	0800.093b.d210	1/1

Each entry lists the information you specified when you created the entry.

Displaying ARP entries

To display ARP entries, refer to the following sections:

- [“Displaying the ARP cache”](#) on page 1070 – Layer 3 Switch
- [“Displaying the static ARP table”](#) on page 1072 – Layer 3 Switch only
- [“Displaying ARP entries”](#) on page 1081 – Layer 2 Switch

To configure other ARP parameters, refer to the following sections:

- [“ARP parameter configuration”](#) on page 981 – Layer 3 Switch only

To increase the size of the ARP cache and static ARP table, refer to the following:

- For dynamic entries, refer to the section [“Displaying and modifying system parameter default settings”](#) on page 578. The `<ip-arp>` parameter controls the ARP cache size.
- Static entries, [“Changing the maximum number of entries the static ARP table can hold”](#) on page 986 (Layer 3 Switches only). The `<ip-static-arp>` parameter controls the static ARP table size.

IP route table

The IP route table contains paths to IP destinations.

NOTE

Layer 2 Switches do not have an IP route table. A Layer 2 Switch sends all packets addressed to another subnet to the default gateway, which you specify when you configure the basic IP information on the Layer 2 Switch.

The IP route table can receive the paths from the following sources:

- A directly-connected destination, which means there are no router hops to the destination
- A static IP route, which is a user-configured route
- A route learned through RIP
- A route learned through OSPF
- A route learned through BGP4

The IP route table contains the best path to a destination:

- When the software receives paths from more than one of the sources listed above, the software compares the administrative distance of each path and selects the path with the lowest administrative distance. The administrative distance is a protocol-independent value from 1 through 255.
- When the software receives two or more best paths from the same source and the paths have the same metric (cost), the software can load share traffic among the paths based on destination host or network address (based on the configuration and the Layer 3 Switch model).

Here is an example of an entry in the IP route table.

Destination	NetMask	Gateway	Port	Cost	Type
1.1.0.0	255.255.0.0	99.1.1.2	1/1	2	R

Each IP route table entry contains the destination IP address and subnet mask and the IP address of the next-hop router interface to the destination. Each entry also indicates the port attached to the destination or the next-hop to the destination, the route IP metric (cost), and the type. The type indicates how the IP route table received the route:

- To display the IP route table, refer to [“Displaying the IP route table”](#) on page 1074 (Layer 3 Switch only).
- To configure a static IP route, refer to [“Static routes configuration”](#) on page 991 (Layer 3 Switch only).
- To clear a route from the IP route table, refer to [“Clearing IP routes”](#) on page 1077 (Layer 3 Switch only).
- To increase the size of the IP route table for learned and static routes, refer to the section [“Displaying and modifying system parameter default settings”](#) on page 578:
 - For learned routes, modify the `<ip-route>` parameter.
 - For static routes, modify the `<ip-static-route>` parameter.

IP forwarding cache

The IP forwarding cache provides a fast-path mechanism for forwarding IP packets. The cache contains entries for IP destinations. When a Brocade Layer 3 Switch has completed processing and addressing for a packet and is ready to forward the packet, the device checks the IP forwarding cache for an entry to the packet destination:

- If the cache contains an entry with the destination IP address, the device uses the information in the entry to forward the packet out the ports listed in the entry. The destination IP address is the address of the packet final destination. The port numbers are the ports through which the destination can be reached.
- If the cache does not contain an entry and the traffic does not qualify for an entry in the session table instead, the software can create an entry in the forwarding cache.

Each entry in the IP forwarding cache has an age timer. If the entry remains unused for ten minutes, the software removes the entry. The age timer is not configurable.

Here is an example of an entry in the IP forwarding cache.

	IP Address	Next Hop	MAC	Type	Port	Vlan	Pri
1	192.168.1.11	DIRECT	0000.0000.0000	PU	n/a		0

Each IP forwarding cache entry contains the IP address of the destination, and the IP address and MAC address of the next-hop router interface to the destination. If the destination is actually an interface configured on the Layer 3 Switch itself, as shown here, then next-hop information indicates this. The port through which the destination is reached is also listed, as well as the VLAN and Layer 4 QoS priority associated with the destination if applicable.

To display the IP forwarding cache, refer to [“Displaying the forwarding cache”](#) on page 1073.

NOTE

You cannot add static entries to the IP forwarding cache, although you can increase the number of entries the cache can contain. Refer to the section [“Displaying and modifying system parameter default settings”](#) on page 578.

Layer 4 session table

The Layer 4 session provides a fast path for forwarding packets. A **session** is an entry that contains complete Layer 3 and Layer 4 information for a flow of traffic. Layer 3 information includes the source and destination IP addresses. Layer 4 information includes the source and destination TCP and UDP ports. For comparison, the IP forwarding cache contains the Layer 3 destination address but does not contain the other source and destination address information of a Layer 4 session table entry.

The Layer 2 Switch or Layer 3 Switch selects the session table instead of the IP forwarding table for fast-path forwarding for the following features:

- Layer 4 Quality-of-Service (QoS) policies
- IP access policies

To increase the size of the session table, refer to the section [“Displaying and modifying system parameter default settings”](#) on page 578. The ip-qos-session parameter controls the size of the session table.

IP route exchange protocols

Brocade Layer 3 Switches support the following IP route exchange protocols:

- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF)
- Border Gateway Protocol version 4 (BGP4)

All these protocols provide routes to the IP route table. You can use one or more of these protocols, in any combination. The protocols are disabled by default. For configuration information, refer to the following:

- [Chapter 29, “RIP \(IPv4\)”](#)
- [Chapter 31, “OSPF version 2 \(IPv4\)”](#)
- [Chapter 33, “BGP \(IPv4\)”](#)

IP multicast protocols

Brocade Layer 3 Switches also support the following Internet Group Membership Protocol (IGMP) based IP multicast protocols:

- Protocol Independent Multicast – Dense mode (PIM-DM)
- Protocol Independent Multicast – Sparse mode (PIM-SM)
- Distance Vector Multicast Routing Protocol (DVMRP)

For configuration information, refer to [Chapter 36, “IP Multicast Protocols”](#).

NOTE

Brocade Layer 2 Switches support IGMP and can forward IP multicast packets. Refer to [Chapter 34, “IP Multicast Traffic Reduction on Brocade FastIron X Series switches”](#) and [Chapter 35, “IP Multicast Traffic Reduction for FastIron WS and Brocade FCX and ICX Switches”](#).

IP interface redundancy protocols

You can configure a Brocade Layer 3 Switch to back up an IP interface configured on another Brocade Layer 3 Switch. If the link for the backed up interface becomes unavailable, the other Layer 3 Switch can continue service for the interface. This feature is especially useful for providing a backup to a network default gateway.

Brocade Layer 3 Switches support the following IP interface redundancy protocols:

- Virtual Router Redundancy Protocol (VRRP) – A standard router redundancy protocol based on RFC 2338. You can use VRRP to configure Brocade Layer 3 Switches and third-party routers to back up IP interfaces on other Brocade Layer 3 Switches or third-party routers.
- Virtual Router Redundancy Protocol Extended (VRRP-E) – A Brocade extension to standard VRRP that adds additional features and overcomes limitations in standard VRRP. You can use VRRP-E only on Brocade Layer 3 Switches.

For configuration information, refer to the [Chapter 39, “VRRP and VRRP-E”](#).

ACLs and IP access policies

Brocade Layer 3 Switches provide two mechanisms for filtering IP traffic:

- Access Control Lists (ACLs)
- IP access policies

Both methods allow you to filter packets based on Layer 3 and Layer 4 source and destination information.

ACLs also provide great flexibility by providing the input to various other filtering mechanisms such as route maps, which are used by BGP4.

IP access policies allow you to configure QoS based on sessions (Layer 4 traffic flows).

Only one of these filtering mechanisms can be enabled on a Brocade device at a time. Brocade devices can store forwarding information for both methods of filtering in the session table.

For configuration information, [Chapter 40, “Rule-Based IP ACLs”](#)

Basic IP parameters and defaults – Layer 3 Switches

IP is enabled by default. The following IP-based protocols are all disabled by default:

- Routing protocols:
 - Routing Information Protocol (RIP) – refer to [Chapter 29, “RIP \(IPv4\)”](#)
 - Open Shortest Path First (OSPF) – refer to [Chapter 31, “OSPF version 2 \(IPv4\)”](#)
 - Border Gateway Protocol version 4 (BGP4) – refer to [Chapter 33, “BGP \(IPv4\)”](#)
- Multicast protocols:
 - Internet Group Membership Protocol (IGMP) – refer to [“Global IP multicast parameters”](#) on page 1517
 - Protocol Independent Multicast Dense (PIM-DM) – refer to [“PIM Dense”](#) on page 1522
 - Protocol Independent Multicast Sparse (PIM-SM) – refer to [“PIM Sparse”](#) on page 1531
- Router redundancy protocols:
 - Virtual Router Redundancy Protocol Extended (VRRP-E) – refer to [Chapter 39, “VRRP and VRRP-E”](#)
 - Virtual Router Redundancy Protocol (VRRP) – refer to [Chapter 39, “VRRP and VRRP-E”](#)

The following tables list the Layer 3 Switch IP parameters, their default values, and where to find configuration information.

NOTE

For information about parameters in other protocols based on IP, such as RIP, OSPF, and so on, refer to the configuration chapters for those protocols.

When parameter changes take effect

Most IP parameters described in this chapter are dynamic. They take effect immediately, as soon as you enter the CLI command or select the Web Management Interface option. You can verify that a dynamic change has taken effect by displaying the running-config. To display the running-config, enter the **show running-config** or **write terminal** command at any CLI prompt. (You cannot display the running-config from the Web Management Interface.)

To save a configuration change permanently so that the change remains in effect following a system reset or software reload, save the change to the startup-config file:

- To save configuration changes to the startup-config file, enter the **write memory** command from the Privileged EXEC level of any configuration level of the CLI.
- To save the configuration changes using the Web Management Interface, select the Save link at the bottom of the dialog. Select Yes when prompted to save the configuration change to the startup-config file on the device flash memory. You also can access the dialog for saving configuration changes by clicking on Command in the tree view, then clicking on *Save to Flash*.

Changes to memory allocation require you to reload the software after you save the changes to the startup-config file. When reloading the software is required to complete a configuration change described in this chapter, the procedure that describes the configuration change includes a step for reloading the software.

IP global parameters – Layer 3 Switches

[Table 164](#) lists the IP global parameters for Layer 3 Switches.

TABLE 164 IP global parameters – Layer 3 Switches

Parameter	Description	Default	For more information
IP state	The Internet Protocol, version 4	Enabled NOTE: You cannot disable IP.	n/a
IP address and mask notation	Format for displaying an IP address and its network mask information. You can enable one of the following: <ul style="list-style-type: none"> • Class-based format; example: 192.168.1.1 255.255.255.0 • Classless Interdomain Routing (CIDR) format; example: 192.168.1.1/24 	Class-based NOTE: Changing this parameter affects the display of IP addresses, but you can enter addresses in either format regardless of the display setting.	page 1065
Router ID	The value that routers use to identify themselves to other routers when exchanging route information. OSPF and BGP4 use router IDs to identify routers. RIP does not use the router ID.	The IP address configured on the lowest-numbered loopback interface. If no loopback interface is configured, then the lowest-numbered IP address configured on the device.	page 976

TABLE 164 IP global parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	For more information
Maximum Transmission Unit (MTU)	The maximum length an Ethernet packet can be without being fragmented.	1500 bytes for Ethernet II encapsulation 1492 bytes for SNAP encapsulation	page 974
Address Resolution Protocol (ARP)	A standard IP mechanism that routers use to learn the Media Access Control (MAC) address of a device on the network. The router sends the IP address of a device in the ARP request and receives the device MAC address in an ARP reply.	Enabled	page 981
ARP rate limiting	Lets you specify a maximum number of ARP packets the device will accept each second. If the device receives more ARP packets than you specify, the device drops additional ARP packets for the remainder of the one-second interval.	Disabled	page 982
ARP age	The amount of time the device keeps a MAC address learned through ARP in the device ARP cache. The device resets the timer to zero each time the ARP entry is refreshed and removes the entry if the timer reaches the ARP age. NOTE: You also can change the ARP age on an individual interface basis. Refer to Table 165 on page 959.	Ten minutes	page 983
Proxy ARP	An IP mechanism a router can use to answer an ARP request on behalf of a host, by replying with the router own MAC address instead of the host.	Disabled	page 983
Static ARP entries	An ARP entry you place in the static ARP table. Static entries do not age out.	No entries	page 985
Time to Live (TTL)	The maximum number of routers (hops) through which a packet can pass before being discarded. Each router decreases a packet TTL by 1 before forwarding the packet. If decreasing the TTL causes the TTL to be 0, the router drops the packet instead of forwarding it.	64 hops	page 987
Directed broadcast forwarding	A directed broadcast is a packet containing all ones (or in some cases, all zeros) in the host portion of the destination IP address. When a router forwards such a broadcast, it sends a copy of the packet out each of its enabled IP interfaces. NOTE: You also can enable or disable this parameter on an individual interface basis. Refer to Table 165 on page 959.	Disabled	page 987
Directed broadcast mode	The packet format the router treats as a directed broadcast. The following formats can be directed broadcast: <ul style="list-style-type: none"> All ones in the host portion of the packet destination address. All zeroes in the host portion of the packet destination address. 	All ones NOTE: If you enable all-zeroes directed broadcasts, all-ones directed broadcasts remain enabled.	page 988

TABLE 164 IP global parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	For more information
Source-routed packet forwarding	A source-routed packet contains a list of IP addresses through which the packet must pass to reach its destination.	Enabled	page 988
Internet Control Message Protocol (ICMP) messages	The Brocade Layer 3 Switch can send the following types of ICMP messages: <ul style="list-style-type: none"> • Echo messages (ping messages) • Destination Unreachable messages 	Enabled	page 989
ICMP Router Discovery Protocol (IRDP)	An IP protocol a router can use to advertise the IP addresses of its router interfaces to directly attached hosts. You can enable or disable the protocol, and change the following protocol parameters: <ul style="list-style-type: none"> • Forwarding method (broadcast or multicast) • Hold time • Maximum advertisement interval • Minimum advertisement interval • Router preference level NOTE: You also can enable or disable IRDP and configure the parameters on an individual interface basis. Refer to Table 165 on page 959.	Disabled	page 1004
Reverse ARP (RARP)	An IP mechanism a host can use to request an IP address from a directly attached router when the host boots.	Enabled	page 1007
Static RARP entries	An IP address you place in the RARP table for RARP requests from hosts. NOTE: You must enter the RARP entries manually. The Layer 3 Switch does not have a mechanism for learning or dynamically generating RARP entries.	No entries	page 1008
Maximum BootP relay hops	The maximum number of hops away a BootP server can be located from a router and still be used by the router clients for network booting.	Four	page 1013
Domain name for Domain Name Server (DNS) resolver	A domain name (example: brocade.router.com) you can use in place of an IP address for certain operations such as IP pings, trace routes, and Telnet management connections to the router.	None configured	page 970
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	page 970
IP load sharing	A Brocade feature that enables the router to balance traffic to a specific destination across multiple equal-cost paths. <p>IP load sharing uses a hashing algorithm based on the source IP address, destination IP address, protocol field in the IP header, TCP, and UDP information.</p> NOTE: Load sharing is sometimes called Equal Cost Multi Path (ECMP).	Enabled	page 1001

TABLE 164 IP global parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	For more information
Maximum IP load sharing paths	The maximum number of equal-cost paths across which the Layer 3 Switch is allowed to distribute traffic.	Four	page 1004
Origination of default routes	You can enable a router to originate default routes for the following route exchange protocols, on an individual protocol basis: <ul style="list-style-type: none"> • RIP • OSPF • BGP4 	Disabled	page 1202 page 1236 page 1352
Default network route	The router uses the default network route if the IP route table does not contain a route to the destination and also does not contain an explicit default route (0.0.0.0 0.0.0.0 or 0.0.0.0/0).	None configured	page 1000
Static route	An IP route you place in the IP route table.	No entries	page 991
Source interface	The IP address the router uses as the source address for Telnet, RADIUS, or TACACS/TACACS+ packets originated by the router. The router can select the source address based on either of the following: <ul style="list-style-type: none"> • The lowest-numbered IP address on the interface the packet is sent on. • The lowest-numbered IP address on a specific interface. The address is used as the source for all packets of the specified type regardless of interface the packet is sent on. 	The lowest-numbered IP address on the interface the packet is sent on.	page 977

IP interface parameters – Layer 3 Switches

[Table 165](#) lists the interface-level IP parameters for Layer 3 Switches.

TABLE 165 IP interface parameters – Layer 3 Switches

Parameter	Description	Default	For more information
IP state	The Internet Protocol, version 4	Enabled NOTE: You cannot disable IP.	n/a
IP address	A Layer 3 network interface address NOTE: Layer 2 Switches have a single IP address used for management access to the entire device. Layer 3 Switches have separate IP addresses on individual interfaces.	None configured ¹	page 964
Encapsulation type	The format of the packets in which the router encapsulates IP datagrams. The encapsulation format can be one of the following: <ul style="list-style-type: none"> • Ethernet II • SNAP 	Ethernet II	page 973

TABLE 165 IP interface parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	For more information
Maximum Transmission Unit (MTU)	The maximum length (number of bytes) of an encapsulated IP datagram the router can forward.	1500 for Ethernet II encapsulated packets 1492 for SNAP encapsulated packets	page 975
ARP age	Locally overrides the global setting. Refer to Table 164 on page 956.	Ten minutes	page 983
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	page 1203
Directed broadcast forwarding	Locally overrides the global setting. Refer to Table 164 on page 956.	Disabled	page 987
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. Refer to Table 164 on page 956.	Disabled	page 1006
DHCP gateway stamp	The router can assist DHCP/BootP Discovery packets from one subnet to reach DHCP/BootP servers on a different subnet by placing the IP address of the router interface that receives the request in the request packet Gateway field. You can override the default and specify the IP address to use for the Gateway field in the packets. NOTE: UDP broadcast forwarding for client DHCP/BootP requests (bootps) must be enabled (this is enabled by default) and you must configure an IP helper address (the server IP address or a directed broadcast to the server subnet) on the port connected to the client.	The lowest-numbered IP address on the interface that receives the request	page 1012
DHCP Client-Based Auto-Configuration	Allows the switch to obtain IP addresses from a DHCP host automatically, for either a specified (leased) or infinite period of time.	Enabled	page 1028
DHCP Server	All FastIron devices can be configured to function as DHCP servers.	Disabled	page 1013

TABLE 165 IP interface parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	For more information
UDP broadcast forwarding	<p>The router can forward UDP broadcast packets for UDP applications such as BootP. By forwarding the UDP broadcasts, the router enables clients on one subnet to find servers attached to other subnets.</p> <p>NOTE: To completely enable a client UDP application request to find a server on another subnet, you must configure an IP helper address consisting of the server IP address or the directed broadcast address for the subnet that contains the server. See the next row.</p>	<p>The router helps forward broadcasts for the following UDP application protocols:</p> <ul style="list-style-type: none"> • bootps • dns • netbios-dgm • netbios-ns • tacacs • tftp • time 	page 1010
IP helper address	The IP address of a UDP application server (such as a BootP or DHCP server) or a directed broadcast address. IP helper addresses allow the router to forward requests for certain UDP applications from a client on one subnet to a server on another subnet.	None configured	page 1011

1. Some devices have a factory default, such as 209.157.22.154, used for troubleshooting during installation. For Layer 3 Switches, the address is on module 1 port 1 (or 1/1).

Basic IP parameters and defaults – Layer 2 Switches

IP is enabled by default. The following tables list the Layer 2 Switch IP parameters, their default values, and where to find configuration information.

NOTE

Brocade Layer 2 Switches also provide IP multicast forwarding, which is enabled by default. For information about this feature, refer to [Chapter 34, “IP Multicast Traffic Reduction on Brocade FastIron X Series switches”](#).

IP global parameters – Layer 2 Switches

[Table 166](#) lists the IP global parameters for Layer 2 Switches.

TABLE 166 IP global parameters – Layer 2 Switches

Parameter	Description	Default	For more information
IP address and mask notation	Format for displaying an IP address and its network mask information. You can enable one of the following: <ul style="list-style-type: none"> Class-based format; example: 192.168.1.1 255.255.255.0 Classless Interdomain Routing (CIDR) format; example: 192.168.1.1/24 	Class-based NOTE: Changing this parameter affects the display of IP addresses, but you can enter addresses in either format regardless of the display setting.	page 1065
IP address	A Layer 3 network interface address NOTE: Layer 2 Switches have a single IP address used for management access to the entire device. Layer 3 Switches have separate IP addresses on individual interfaces.	None configured ¹	page 1037
Default gateway	The IP address of a locally attached router (or a router attached to the Layer 2 Switch by bridges or other Layer 2 Switches). The Layer 2 Switch and clients attached to it use the default gateway to communicate with devices on other subnets.	None configured	page 1037
Address Resolution Protocol (ARP)	A standard IP mechanism that networking devices use to learn the Media Access Control (MAC) address of another device on the network. The Layer 2 Switch sends the IP address of a device in the ARP request and receives the device MAC address in an ARP reply.	Enabled NOTE: You cannot disable ARP.	n/a
ARP age	The amount of time the device keeps a MAC address learned through ARP in the device ARP cache. The device resets the timer to zero each time the ARP entry is refreshed and removes the entry if the timer reaches the ARP age.	Ten minutes NOTE: You cannot change the ARP age on Layer 2 Switches.	n/a

TABLE 166 IP global parameters – Layer 2 Switches (Continued)

Parameter	Description	Default	For more information
Time to Live (TTL)	The maximum number of routers (hops) through which a packet can pass before being discarded. Each router decreases a packet TTL by 1 before forwarding the packet. If decreasing the TTL causes the TTL to be 0, the router drops the packet instead of forwarding it.	64 hops	page 1039
Domain name for Domain Name Server (DNS) resolver	A domain name (example: brocade.router.com) you can use in place of an IP address for certain operations such as IP pings, trace routes, and Telnet management connections to the router.	None configured	page 1038
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	page 1038
Source interface	The IP address the Layer 2 Switch uses as the source address for Telnet, RADIUS, or TACACS/TACACS+ packets originated by the router. The Layer 2 Switch uses its management IP address as the source address for these packets.	The management IP address of the Layer 2 Switch. NOTE: This parameter is not configurable on Layer 2 Switches.	n/a
DHCP gateway stamp	The device can assist DHCP/BootP Discovery packets from one subnet to reach DHCP/BootP servers on a different subnet by placing the IP address of the router interface that forwards the packet in the packet Gateway field. You can specify up to 32 gateway lists. A gateway list contains up to eight gateway IP addresses. You activate DHCP assistance by associating a gateway list with a port. When you configure multiple IP addresses in a gateway list, the Layer 2 Switch inserts the addresses into the DHCP Discovery packets in a round robin fashion.	None configured	page 1043
DHCP Client-Based Auto-Configuration	Allows the switch to obtain IP addresses from a DHCP host automatically, for either a specified (leased) or infinite period of time.	Enabled	page 1028

1. Some devices have a factory default, such as 209.157.22.154, used for troubleshooting during installation. For Layer 3 Switches, the address is on port 1 (or 1/1).

Interface IP parameters – Layer 2 Switches

[Table 167](#) lists the interface-level IP parameters for Layer 2 Switches.

TABLE 167 Interface IP parameters – Layer 2 Switches

Parameter	Description	Default	For more information
DHCP gateway stamp	You can configure a list of DHCP stamp addresses for a port. When the port receives a DHCP/BootP Discovery packet from a client, the port places the IP addresses in the gateway list into the packet Gateway field.	None configured	page 1043

Configuring IP parameters – Layer 3 Switches

The following sections describe how to configure IP parameters. Some parameters can be configured globally while others can be configured on individual interfaces. Some parameters can be configured globally and overridden for individual interfaces.

NOTE

This section describes how to configure IP parameters for Layer 3 Switches. For IP configuration information for Layer 2 Switches, refer to [“Configuring IP parameters – Layer 2 Switches”](#) on page 1037.

Configuring IP addresses

You can configure an IP address on the following types of Layer 3 Switch interfaces:

- Ethernet port
- Virtual routing interface (also called a Virtual Ethernet or “VE”)
- Loopback interface

By default, you can configure up to 24 IP addresses on each interface.

You can increase this amount to up to 128 IP subnet addresses per port by increasing the size of the ip-subnet-port table.

On FWS devices, you can increase this to 128 IP addresses on each interface.

Refer to the section [“Displaying system parameter default values”](#) on page 578.

NOTE

Once you configure a virtual routing interface on a VLAN, you cannot configure Layer 3 interface parameters on individual ports. Instead, you must configure the parameters on the virtual routing interface itself.

Brocade devices support both classical IP network masks (Class A, B, and C subnet masks, and so on) and Classless Interdomain Routing (CIDR) network prefix masks:

- To enter a classical network mask, enter the mask in IP address format. For example, enter “209.157.22.99 255.255.255.0” for an IP address with a Class-C subnet mask.
- To enter a prefix network mask, enter a forward slash (/) and the number of bits in the mask immediately after the IP address. For example, enter “209.157.22.99/24” for an IP address that has a network mask with 24 significant bits (ones).

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the display to prefix format. Refer to [“Changing the network mask display to prefix format”](#) on page 1065.

Assigning an IP address to an Ethernet port

To assign an IP address to port 1/1, enter the following commands.

```
Brocade(config)# interface ethernet 1/1
Brocade(config-if-1/1)# ip address 192.45.6.1 255.255.255.0
```

You also can enter the IP address and mask in CIDR format, as follows.

```
Brocade(config-if-1/1)# ip address 192.45.6.1/24
```

Syntax: [no] ip address <ip-addr> <ip-mask> [ospf-ignore | ospf-passive | secondary]

or

Syntax: [no] ip address <ip-addr>/<mask-bits> [ospf-ignore | ospf-passive | secondary]

The **ospf-ignore** | **ospf-passive** parameters modify the Layer 3 Switch defaults for adjacency formation and interface advertisement. Use one of these parameters if you are configuring multiple IP subnet addresses on the interface but you want to prevent OSPF from running on some of the subnets:

- **ospf-passive** – This option disables adjacency formation with OSPF neighbors. By default, when OSPF is enabled on an interface, the software forms OSPF router adjacencies between each primary IP address on the interface and the OSPF neighbor attached to the interface.
- **ospf-ignore** – This option disables OSPF adjacency formation and also disables advertisement of the interface into OSPF. The subnet is completely ignored by OSPF.

NOTE

The **ospf-passive** option disables adjacency formation but does not disable advertisement of the interface into OSPF. To disable advertisement in addition to disabling adjacency formation, you must use the **ospf-ignore** option.

Use the **secondary** parameter if you have already configured an IP address within the same subnet on the interface.

NOTE

When you configure more than one address in the same subnet, all but the first address are secondary addresses and do not form OSPF adjacencies.

NOTE

All physical IP interfaces on Brocade FastIron Layer 3 devices share the same MAC address. For this reason, if more than one connection is made between two devices, one of which is a Brocade FastIron Layer 3 device, Brocade recommends the use of virtual interfaces. It is not recommended to connect two or more physical IP interfaces between two routers.

Assigning an IP address to a loopback interface

Loopback interfaces are always up, regardless of the states of physical interfaces. They can add stability to the network because they are not subject to route flap problems that can occur due to unstable links between a Layer 3 Switch and other devices. You can configure up to eight loopback interfaces on a Chassis Layer 3 Switch and FWS devices. You can configure up to four loopback interfaces on a Compact Layer 3 Switch.

You can add up to 24 IP addresses to each loopback interface.

NOTE

If you configure the Brocade Layer 3 Switch to use a loopback interface to communicate with a BGP4 neighbor, you also must configure a loopback interface on the neighbor and configure the neighbor to use that loopback interface to communicate with the Brocade Layer 3 Switch. Refer to [“Adding a loopback interface”](#) on page 1353.

To add a loopback interface, enter commands such as those shown in the following example.

```
Brocade(config-bgp-router)# exit
Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 10.0.0.1/24
```

Syntax: interface loopback <num>

The <num> parameter specifies the virtual interface number. You can specify from 1 to the maximum number of virtual interfaces supported on the device. To display the maximum number of virtual interfaces supported on the device, enter the **show default values** command. The maximum is listed in the System Parameters section, in the Current column of the virtual-interface row.

Refer to the syntax description in [“Assigning an IP address to an Ethernet port”](#) on page 965.

Assigning an IP address to a virtual interface

A virtual interface is a logical port associated with a Layer 3 Virtual LAN (VLAN) configured on a Layer 3 Switch. You can configure routing parameters on the virtual interface to enable the Layer 3 Switch to route protocol traffic from one Layer 3 VLAN to the other, without using an external router.¹

You can configure IP routing interface parameters on a virtual interface. This section describes how to configure an IP address on a virtual interface. Other sections in this chapter that describe how to configure interface parameters also apply to virtual interfaces.

NOTE

The Layer 3 Switch uses the lowest MAC address on the device (the MAC address of port 1 or 1/1) as the MAC address for all ports within all virtual interfaces you configure on the device.

1. The Brocade feature that allows routing between VLANs within the same device, without the need for external routers, is called Integrated Switch Routing (ISR).

To add a virtual interface to a VLAN and configure an IP address on the interface, enter commands such as the following.

```
Brocade(config)# vlan 2 name IP-Subnet_1.1.2.0/24
Brocade(config-vlan-2)# untag ethernet 1 to 4
Brocade(config-vlan-2)# router-interface ve1
Brocade(config-vlan-2)# interface ve1
Brocade(config-vif-1)# ip address 1.1.2.1/24
```

The first two commands in this example create a Layer 3 protocol-based VLAN name “IP-Subnet_1.1.2.0/24” and add a range of untagged ports to the VLAN. The **router-interface** command creates virtual interface 1 as the routing interface for the VLAN.

Syntax: **router-interface ve** <num>

The <num> variable specifies the virtual interface number. You can enter a number from 1 through 4095.

When configuring virtual routing interfaces on a device, you can specify a number from 1 through 4095. However, the total number of virtual routing interfaces that are configured must not exceed the system-max limit of 512. For more information on the number of virtual routing interfaces supported, refer to [“Allocating memory for more VLANs or virtual routing interfaces”](#) on page 795.

The last two commands change to the interface configuration level for the virtual interface and assign an IP address to the interface.

Syntax: **interface ve** <num>

Refer to the syntax description in [“Assigning an IP address to an Ethernet port”](#) on page 965.

Configuring IP follow on a virtual routing interface

IP Follow allows multiple virtual routing interfaces to share the same IP address. With this feature, one virtual routing interface is configured with an IP address, while the other virtual routing interfaces are configured to use that IP address, thus, they “follow” the virtual routing interface that has the IP address. This feature is helpful in conserving IP address space.

Configuration limitations and feature limitations for IP Follow on a virtual routing interface

- When configuring IP Follow, the primary virtual routing interface should not have ACL or DoS Protection configured. It is recommended that you create a dummy virtual routing interface as the primary and use the IP-follow virtual routing interface for the network.
- Global Policy Based Routing is not supported when IP Follow is configured.
- IPv6 is not supported with **ip-follow**.
- FastIron devices support **ip-follow** with OSPF and VRRP protocols only.

Configuration syntax for IP Follow on a virtual routing interface

Configure IP Follow by entering commands such as the following.

```
Brocade(config)# vlan 2 name IP-Subnet_1.1.2.0/24
Brocade(config-vlan-2)# untag ethernet 1 to 4
Brocade(config-vlan-2)# router-interface ve1
Brocade(config-vlan-2)# interface ve 1
Brocade(config-vif-1)# ip address 10.10.2.1/24
```

```
Brocade(config-vif-1)# interface ve 2
Brocade(config-vif-2)# ip follow ve 1
Brocade(config-vif-2)# interface ve 3
Brocade(config-vif-3)# ip follow ve 1
```

Syntax: [no] ip follow ve <number>

For <number>, enter the ID of the virtual routing interface.

Use the **no** form of the command to disable the configuration.

Virtual routing interface 2 and 3 do not have their own IP subnet addresses, but are sharing the IP address of virtual routing interface 1.

Deleting an IP address

To delete an IP address, enter the **no ip address** command.

```
Brocade(config-if-e1000-1)# no ip address 1.1.2.1
```

This command deletes IP address 1.1.2.1. You do not need to enter the subnet mask.

To delete all IP addresses from an interface, enter the **no ip address *** command.

```
Brocade(config-if-e1000-1)# no ip address *
```

Syntax: no ip address <ip-addr> | *

Configuring 31-bit subnet masks on point-to-point networks

NOTE

31-bit subnet masks are supported on FSX, FCX, and ICX 6610 devices running the full Layer 3 image.

To conserve IPv4 address space, a 31-bit subnet mask can be assigned to point-to-point networks. Support for an IPv4 address with a 31-bit subnet mask is described in RFC 3021.

With IPv4, four IP addresses with a 30-bit subnet mask are allocated on point-to-point networks. In contrast, a 31-bit subnet mask uses only two IP addresses: all zero bits and all one bits in the host portion of the IP address. The two IP addresses are interpreted as host addresses, and do not require broadcast support because any packet that is transmitted by one host is always received by the other host at the receiving end. Therefore, directed broadcast on a point-to-point interface is eliminated.

IP-directed broadcast CLI configuration at the global level, or the per interface level, is not applicable on interfaces configured with a 31-bit subnet mask IP address.

When the 31-bit subnet mask address is configured on a point-to-point link, using network addresses for broadcast purposes is not allowed. For example, in an IPV4 broadcast scheme, the following subnets can be configured:

- 10.10.10.1 - Subnet for directed broadcast: {<Network-number>, -1}
- 10.10.10.0 - Subnet for network address: {<Network-number>, 0}

In a point-to-point link with a 31-bit subnet mask, the previous two addresses are interpreted as host addresses and packets are not rebroadcast.

Configuring an IPv4 address with a 31-bit subnet mask

To configure an IPv4 address with a 31-bit subnet mask, enter the following commands.

You can configure an IPv4 address with a 31-bit subnet mask on any interface (for example, Ethernet, loopback, VE, or tunnel interfaces).

```
Brocade(config)# interface ethernet 1/1/5
Brocade(config-if-e1000-1/5)# ip address 9.9.9.9 255.255.255.254
```

You can also enter the IP address and mask in the Classless Inter-domain Routing (CIDR) format, as follows.

```
Brocade(config-if-e1000-1/1/5)# ip address 9.9.9.9/31
```

Syntax: [no] ip address <ip-address> <ip-mask>

Syntax: [no] ip address <ip-address>/<subnet mask-bits>

The <ip-address> variable specifies the host address. The <ip-mask> variable specifies the IP network mask. The <subnet mask-bits> variable specifies the network prefix mask.

To disable configuration for an IPv4 address with a 31-bit subnet mask on any interface, use the **no** form of the command.

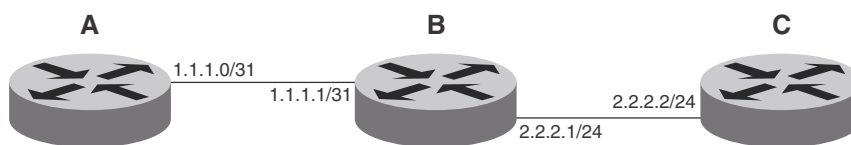
You cannot configure a secondary IPv4 address with a 31-bit subnet mask on any interface. The following error message is displayed when a secondary IPv4 address with a 31-bit subnet mask is configured.

```
Error: Cannot assign /31 subnet address as secondary
```

Configuration example

Figure 111 shows the usage of 31- and 24-bit subnet masks in configuring IP addresses.

FIGURE 111 Configured 31-bit and 24-bit subnet masks



Router A is connected to Router B as a point-to-point link with 1.1.1.0/31 subnet. There are only two available addresses in this subnet, 1.1.1.0 on Router A and 1.1.1.1 on Router B,

Routers B and C are connected by a regular 24-bit subnet. Router C can either be a switch with many hosts belonging to the 2.2.2.2/24 subnet connected to it, or it can be a router.

Router A

```
RouterA(config)# interface ethernet 1/1/1
RouterA(config-if-e1000-1/1/1)# ip address 1.1.1.0/31
```

Router B

```
RouterB(config)# interface ethernet 1/1/1
RouterB(config-if-e1000-1/1/1)# ip address 1.1.1.1/31
RouterB(config-if-e1000-1/1/1)# exit
```

```
RouterB(config# interface ethernet 1/3/1
RouterB(config-if-e1000-1/3/1)# ip address 2.2.2.1/24
```

Router C

```
RouterC(config# interface ethernet 1/3/1
RouterC(config-if-e1000-1/3/1)# ip address 2.2.2.2/24
```

Displaying information for a 31-bit subnet mask

Use the following commands to display information for the 31-bit subnet mask:

- **show run interface**
- **show ip route**
- **show ip cache**

Configuring DNS resolver

The Domain Name System (DNS) resolver is a feature in a Layer 2 or Layer 3 switch that sends and receives queries to and from the DNS server on behalf of a client.

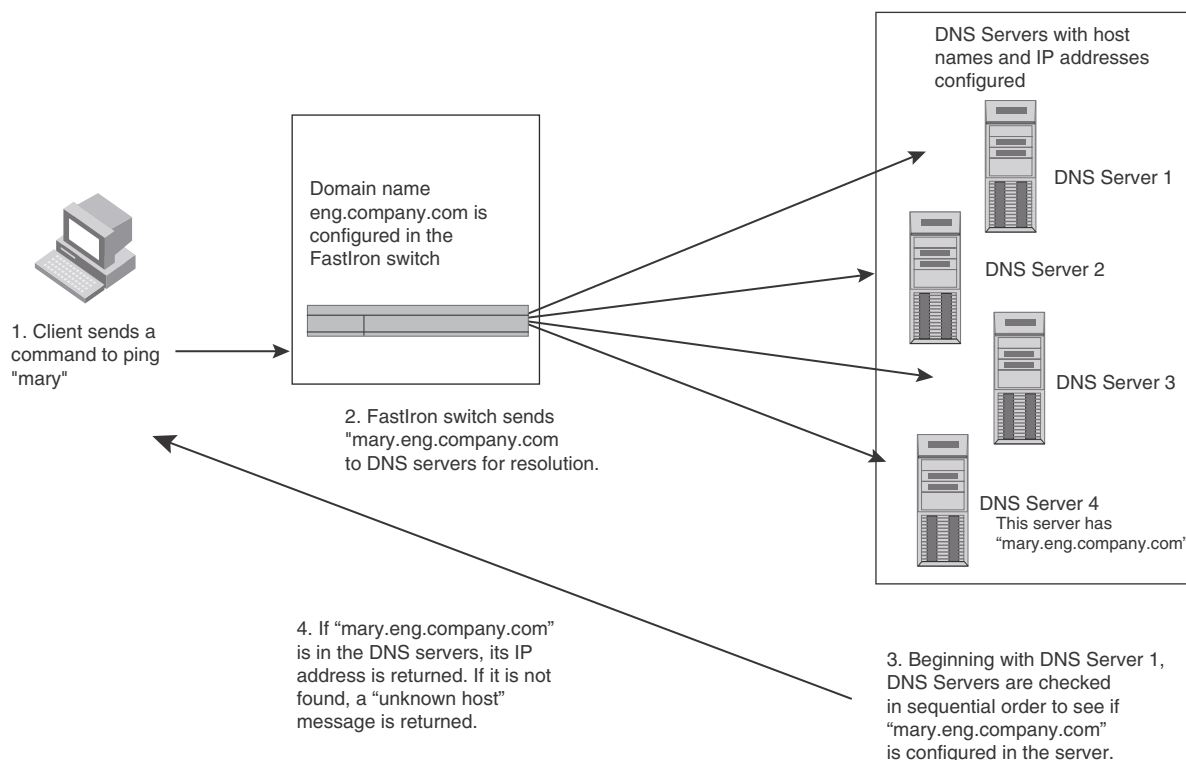
You can create a list of domain names that can be used to resolve host names. This list can have more than one domain name. When a client performs a DNS query, all hosts within the domains in the list can be recognized and queries can be sent to any domain on the list.

After you define a domain name, the Brocade device automatically appends the appropriate domain to a host and forwards it to the DNS servers for resolution.

For example, if the domain “ds.company.com” is defined on a Layer 2 or Layer 3 switch and you want to initiate a ping to “mary”, you must reference only the host name instead of the host name and its domain name. For example, you could enter the following command to initiate the ping.

```
U:> ping mary
```

The Layer 2 or Layer 3 switch qualifies the host name by appending a domain name (for example, mary.ds1.company.com). This qualified name is sent to the DNS server for resolution. If there are four DNS servers configured, it is sent to the first DNS server. If the host name is not resolved, it is sent to the second DNS server. If a match is found, a response is sent back to the client with the host IP address. If no match is found, an “unknown host” message is returned. (Refer to [Figure 112.](#))

FIGURE 112 DNS resolution with one domain name

Defining a domain name

To define a domain to resolve host names, enter the **ip dns domain-name** command.

```
Brocade(config)# ip dns domain-name ds.company.com
```

Syntax: [no] **ip dns domain-name** <domain-name>

Enter the domain name for <domain-name>.

Defining DNS server addresses

You can configure the Brocade device to recognize up to four DNS servers. The first entry serves as the primary default address. If a query to the primary address fails to be resolved after three attempts, the next DNS address is queried (also up to three times). This process continues for each defined DNS address until the query is resolved. The order in which the default DNS addresses are polled is the same as the order in which you enter them.

To define DNS servers, enter the **ip dns server-address** command.

```
Brocade(config)# ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25  
201.98.7.15
```

Syntax: [no] **ip dns server-address** <ip-addr> [<ip-addr>] [<ip-addr>] [<ip-addr>]

In this example, the first IP address entered becomes the primary DNS address and all others are secondary addresses. Because IP address 201.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

Defining a domain list

If you want to use more than one domain name to resolve host names, you can create a list of domain names. For example, enter the commands such as the following.

```
Brocade(config)# ip dns domain-list company.com
Brocade(config)# ip dns domain-list ds.company.com
Brocade(config)# ip dns domain-list hw_company.com
Brocade(config)# ip dns domain-list qa_company.com
Brocade(config)#
```

The domain names are tried in the order you enter them

Syntax: [no] ip dns domain-list <domain-name>

Using a DNS name to initiate a trace route

Suppose you want to trace the route from a Brocade Layer 3 Switch to a remote server identified as NYC02 on domain newyork.com. Because the NYC02@ds1.newyork.com domain is already defined on the Layer 3 Switch, you need to enter only the host name, NYC02, as noted in the following example.

```
Brocade# traceroute nyc02
```

Syntax: traceroute <host-ip-addr> [maxttl <value>] [minttl <value>] [numeric] [timeout <value>]
[source-ip <ip addr>]

The only required parameter is the IP address of the host at the other end of the route.

After you enter the command, a message indicating that the DNS query is in process and the current gateway address (IP address of the domain name server) being queried appear on the screen.

```
Type Control-c to abort
Sending DNS Query to 209.157.22.199
Tracing Route to IP node 209.157.22.80
To ABORT Trace Route, Please use stop-traceroute command.
Traced route to target IP node 209.157.22.80:
  IP Address      Round Trip Time1    Round Trip Time2
  207.95.6.30     93 msec            121 msec
```

NOTE

In the previous example, 209.157.22.199 is the IP address of the domain name server (default DNS gateway address), and 209.157.22.80 represents the IP address of the NYC02 host.

Configuring packet parameters

You can configure the following packet parameters on Layer 3 Switches. These parameters control how the Layer 3 Switch sends IP packets to other devices on an Ethernet network. The Layer 3 Switch always places IP packets into Ethernet packets to forward them on an Ethernet port.

- **Encapsulation type** – The format for the Layer 2 packets within which the Layer 3 Switch sends IP packets.
- **Maximum Transmission Unit (MTU)** – The maximum length of IP packet that a Layer 2 packet can contain. IP packets that are longer than the MTU are fragmented and sent in multiple Layer 2 packets. You can change the MTU globally or on individual ports:
 - **Global MTU** – The default MTU value depends on the encapsulation type on a port and is 1500 bytes for Ethernet II encapsulation and 1492 bytes for SNAP encapsulation.
 - **Port MTU** – A port default MTU depends on the encapsulation type enabled on the port.

Changing the encapsulation type

The Layer 3 Switch encapsulates IP packets into Layer 2 packets, to send the IP packets on the network. (A Layer 2 packet is also called a MAC layer packet or an Ethernet frame.) The source address of a Layer 2 packet is the MAC address of the Layer 3 Switch interface sending the packet. The destination address can be one of the following:

- The MAC address of the IP packet destination. In this case, the destination device is directly connected to the Layer 3 Switch.
- The MAC address of the next-hop gateway toward the packet destination.
- An Ethernet broadcast address.

The entire IP packet, including the source and destination address and other control information and the data, is placed in the data portion of the Layer 2 packet. Typically, an Ethernet network uses one of two different formats of Layer 2 packet:

- Ethernet II
- Ethernet SNAP (also called IEEE 802.3)

The control portions of these packets differ slightly. All IP devices on an Ethernet network must use the same format. Brocade Layer 3 Switches use Ethernet II by default. You can change the IP encapsulation to Ethernet SNAP on individual ports if needed.

NOTE

All devices connected to the Layer 3 Switch port must use the same encapsulation type.

To change the IP encapsulation type on interface 5 to Ethernet SNAP, enter the following commands.

```
Brocade(config)# interface ethernet 5
Brocade(config-if-e1000-5)# ip encapsulation snap
```

Syntax: `ip encapsulation snap | ethernet_ii`

Changing the MTU

The Maximum Transmission Unit (MTU) is the maximum length of IP packet that a Layer 2 packet can contain. IP packets that are longer than the MTU are fragmented and sent in multiple Layer 2 packets. You can change the MTU globally or on individual ports.

The default MTU is 1500 bytes for Ethernet II packets and 1492 for Ethernet SNAP packets.

MTU enhancements

Brocade devices contain the following enhancements to jumbo packet support:

- Hardware forwarding of Layer 3 jumbo packets – Layer 3 IP unicast jumbo packets received on a port that supports the frame MTU size and forwarded to another port that also supports the frame MTU size are forwarded in hardware. Previous releases support hardware forwarding of Layer 2 jumbo frames only.
- ICMP unreachable message if a frame is too large to be forwarded – If a jumbo packet has the Do not Fragment (DF) bit set, and the outbound interface does not support the packet MTU size, the Brocade device sends an ICMP unreachable message to the device that sent the packet.

NOTE

These enhancements apply only to transit traffic forwarded through the Brocade device.

Configuration considerations for increasing the MTU

- The MTU command is applicable to VEs and physical IP interfaces. It applies to traffic routed between networks.
- You cannot use this command to set Layer 2 maximum frame sizes per interface. The global **jumbo** command causes all interfaces to accept Layer 2 frames.
- When you increase the MTU size of a port, the increase uses system resources. Increase the MTU size only on the ports that need it. For example, if you have one port connected to a server that uses jumbo frames and two other ports connected to clients that can support the jumbo frames, increase the MTU only on those three ports. Leave the MTU size on the other ports at the default value (1500 bytes). Globally increase the MTU size only if needed.

Forwarding traffic to a port with a smaller MTU size

NOTE

This feature is not supported on FastIron X Series devices.

In order to forward traffic from a port with 1500 MTU configured to a port that has a smaller MTU (for example, 750) size, you must apply the **mtu-exceed forward** global command. To remove this setting, enter the **mtu-exceed hard-drop** command. **MTU-exceed hard-drop** is the default state of the router.

Syntax:mtu-exceed [forward | hard-drop]

- **forward** - forwards a packet from a port with a larger MTU to a port with a smaller MTU
- **hard-drop** - resets to default, removes the forward function.

Globally changing the Maximum Transmission Unit

The Maximum Transmission Unit (MTU) is the maximum size an IP packet can be when encapsulated in a Layer 2 packet. If an IP packet is larger than the MTU allowed by the Layer 2 packet, the Layer 3 Switch fragments the IP packet into multiple parts that will fit into the Layer 2 packets, and sends the parts of the fragmented IP packet separately, in different Layer 2 packets. The device that receives the multiple fragments of the IP packet reassembles the fragments into the original packet.

You can increase the MTU size to accommodate jumbo packet sizes up to up to 10,232 bytes in an IronStack. Devices that are not part of an IronStack support up to 10,240 bytes.

To globally enable jumbo support on all ports of a FastIron device, enter commands such as the following.

```
Brocade(config)# jumbo
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload
```

Syntax: [no] jumbo

NOTE

You must save the configuration change and then reload the software to enable jumbo support.

Changing the MTU on an individual port

By default, the maximum Ethernet MTU sizes are as follows:

- 1500 bytes – The maximum for Ethernet II encapsulation
- 1492 bytes – The maximum for SNAP encapsulation

When jumbo mode is enabled, the maximum Ethernet MTU sizes are as follows:

- 10,240 bytes– The maximum for Ethernet II encapsulation
- 10,240 bytes – The maximum for SNAP encapsulation

NOTE

If you set the MTU of a port to a value lower than the global MTU and from 576 through 1499, the port fragments the packets. However, if the port MTU is exactly 1500 and this is larger than the global MTU, the port drops the packets.

NOTE

The **ip mtu** command is not supported on FWS devices.

NOTE

You must save the configuration change and then reload the software to enable jumbo support.

To change the MTU for interface 1/5 to 1000, enter the following commands.

```
Brocade(config)# interface ethernet 1/5
Brocade(config-if-1/5)# ip mtu 1000
Brocade(config-if-1/5)# write memory
Brocade(config-if-1/5)# end
Brocade# reload
```

Syntax: [no] ip mtu <num>

The `<num>` parameter specifies the MTU. Ethernet II packets can hold IP packets from 576 through 1500 bytes long. If jumbo mode is enabled, Ethernet II packets can hold IP packets up to 10,240 bytes long. Ethernet SNAP packets can hold IP packets from 576 through 1492 bytes long. If jumbo mode is enabled, SNAP packets can hold IP packets up to 10,240 bytes long. The default MTU for Ethernet II packets is 1500. The default MTU for SNAP packets is 1492.

Path MTU discovery (RFC 1191) support

FastIron X Series devices support the path MTU discovery method described in RFC 1191. When the Brocade device receives an IP packet that has its Do not Fragment (DF) bit set, and the packet size is greater than the MTU value of the outbound interface, then the Brocade device returns an ICMP Destination Unreachable message to the source of the packet, with the Code indicating "fragmentation needed and DF set". The ICMP Destination Unreachable message includes the MTU of the outbound interface. The source host can use this information to help determine the maximum MTU of a path to a destination.

RFC 1191 is supported on all interfaces.

Changing the router ID

In most configurations, a Layer 3 Switch has multiple IP addresses, usually configured on different interfaces. As a result, a Layer 3 Switch identity to other devices varies depending on the interface to which the other device is attached. Some routing protocols, including Open Shortest Path First (OSPF) and Border Gateway Protocol version 4 (BGP4), identify a Layer 3 Switch by just one of the IP addresses configured on the Layer 3 Switch, regardless of the interfaces that connect the Layer 3 Switches. This IP address is the router ID.

NOTE

Routing Information Protocol (RIP) does not use the router ID.

NOTE

If you change the router ID, all current BGP4 sessions are cleared.

By default, the router ID on a Brocade Layer 3 Switch is one of the following:

- If the router has loopback interfaces, the default router ID is the IP address configured on the lowest numbered loopback interface configured on the Layer 3 Switch. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default router ID is 9.9.9.9/24:
 - Loopback interface 1, 9.9.9.9/24
 - Loopback interface 2, 4.4.4.4/24
 - Loopback interface 3, 1.1.1.1/24
- If the device does not have any loopback interfaces, the default router ID is the lowest numbered IP interface configured on the device.

If you prefer, you can explicitly set the router ID to any valid IP address. The IP address cannot be in use on another device in the network.

NOTE

Brocade Layer 3 Switches use the same router ID for both OSPF and BGP4. If the router is already configured for OSPF, you may want to use the router ID that is already in use on the router rather than set a new one. To display the router ID, enter the **show ip** command at any CLI level or select the IP->General links from the Configure tree in the Web Management Interface.

To change the router ID, enter a command such as the following.

```
Brocade(config)# ip router-id 209.157.22.26
```

Syntax: `ip router-id <ip-addr>`

The `<ip-addr>` can be any valid, unique IP address.

NOTE

You can specify an IP address used for an interface on the Brocade Layer 3 Switch, but do not specify an IP address in use by another device.

Specifying a single source interface for specified packet types

NOTE

This feature is supported on FWS, Brocade FCX Series switches, FastIron X Series Layer 3 switches, ICX 6430 and ICX 6450 switches.

When the Layer 3 Switch originates a packet of one of the following types, the source address of the packet is the lowest-numbered IP address on the interface that sends the packet:

- Telnet
- TACACS/TACACS+
- TFTP
- RADIUS
- Syslog
- SNMP
- SSH
- SNMP traps

You can configure the Layer 3 Switch to always use the lowest-numbered IP address on a specific Ethernet, loopback, or virtual interface as the source addresses for these packets. When configured, the Layer 3 Switch uses the same IP address as the source for all packets of the specified type, regardless of the ports that actually sends the packets.

Identifying a single source IP address for specified packets provides the following benefits:

- If your server is configured to accept packets only from specific IP addresses, you can use this feature to simplify configuration of the server by configuring the Brocade device to always send the packets from the same link or source address.
- If you specify a loopback interface as the single source for specified packets, servers can receive the packets regardless of the states of individual links. Thus, if a link to the server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for specific packets. You can configure a source interface for one or more of these types of packets separately.

The following sections show the syntax for specifying a single source IP address for specific packet types.

Telnet packets

To specify the lowest-numbered IP address configured on a virtual interface as the device source for all Telnet packets, enter commands such as the following.

```
Brocade(config)# interface loopback 2
Brocade(config-lbif-2)# ip address 10.0.0.2/24
Brocade(config-lbif-2)# exit
Brocade(config)# ip telnet source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all Telnet packets from the Layer 3 Switch.

The following commands configure an IP interface on an Ethernet port and designate the address port as the source for all Telnet packets from the Layer 3 Switch.

```
Brocade(config)# interface ethernet 1/4
Brocade(config-if-1/4)# ip address 209.157.22.110/24
Brocade(config-if-1/4)# exit
Brocade(config)# ip telnet source-interface ethernet 1/4
```

Syntax: [no] ip telnet source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <slotnum> variable is required on chassis devices.

The <portnum> variable is a valid port number.

The <num> variable is a loopback interface or virtual interface number.

TACACS/TACACS+ packets

To specify the lowest-numbered IP address configured on a virtual interface as the device source for all TACACS/TACACS+ packets, enter commands such as the following.

```
Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip address 10.0.0.3/24
Brocade(config-vif-1)# exit
Brocade(config)# ip tacacs source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all TACACS/TACACS+ packets from the Layer 3 Switch.

Syntax: [no] ip tacacs source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <slotnum> variable is required on chassis devices.

The <portnum> variable is a valid port number.

The <num> variable is a loopback interface or virtual interface number.

RADIUS packets

To specify the lowest-numbered IP address configured on a virtual interface as the device source for all RADIUS packets, enter commands such as the following.


```

Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip address 10.0.0.3/24
Brocade(config-vif-1)# exit
Brocade(config)# ip radius source-interface ve 1

```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all RADIUS packets from the Layer 3 Switch.

Syntax: [no] **ip radius source-interface ethernet** [<slotnum>/<portnum> | **loopback** <num> | **ve** <num>]

The <slotnum> variable is required on chassis devices.

The <portnum> variable is a valid port number.

The <num> variable is a loopback interface or virtual interface number.

TFTP packets

To specify the lowest-numbered IP address configured on a virtual interface as the device source for all TFTP packets, enter commands such as the following.

```

Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip address 10.0.0.3/24
Brocade(config-vif-1)# exit
Brocade(config)# ip tftp source-interface ve 1

```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface's address as the source address for all TFTP packets.

Syntax: [no] **ip tftp source-interface ethernet** [<slotnum>/<portnum> | **loopback** <num> | **ve** <num>]

The <slotnum> variable is required on chassis devices.

The <portnum> variable is a valid port number.

The <num> variable is a loopback interface or virtual interface number.

The default is the lowest-numbered IP address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

Syslog packets

To specify the lowest-numbered IP address configured on a virtual interface as the device source for all Syslog packets, enter commands such as the following.

```

Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip address 10.0.0.4/24
Brocade(config-vif-1)# exit
Brocade(config)# ip syslog source-interface ve 1

```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.4/24 to the interface, then designate the interface's address as the source address for all Syslog packets.

Syntax: [no] **ip syslog source-interface ethernet** [<slotnum>/<portnum> | **loopback** <num> | **ve** <num>]

The <slotnum> variable is required on chassis devices.

The `<portnum>` variable is a valid port number.

The `<num>` variable is a loopback interface or virtual interface number.

The default is the lowest-numbered IP or IPv6 address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

SNTP packets

To specify the lowest-numbered IP address configured on a virtual interface as the device source for all SNTP packets, enter commands such as the following.

```
Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip address 10.0.0.5/24
Brocade(config-vif-1)# exit
Brocade(config)# ip sntp source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.5/24 to the interface, then designate the interface's address as the source address for all SNTP packets.

Syntax: `[no] ip sntp source-interface ethernet [<slotnum>]/<portnum> | loopback <num> | ve <num>`

The `<slotnum>` variable is required on chassis devices.

The `<portnum>` variable is a valid port number.

The `<num>` variable is a loopback interface or virtual interface number.

The default is the lowest-numbered IP or IPv6 address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

SSH packets

NOTE

When you specify a single SSH source, you can use only that source address to establish SSH management sessions with the Brocade device.

To specify the numerically lowest IP address configured on a loopback interface as the device source for all SSH packets, enter commands such as the following.

```
Brocade(config)# interface loopback 2
Brocade(config-lbif-2)# ip address 10.0.0.2/24
Brocade(config-lbif-2)# exit
Brocade(config)# ip ssh source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all SSH packets from the Layer 3 Switch.

Syntax: `[no] ip ssh source-interface ethernet [<slotnum>]/<portnum> | loopback <num> | ve <num>`

The `<slotnum>` parameter is required on chassis devices.

The `<portnum>` parameter is a valid port number.

The `<num>` parameter is a loopback interface or virtual interface number.

SNMP packets

To specify a loopback interface as the SNMP single source trap, enter commands such as the following.

```
Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 10.0.0.1/24
Brocade(config-lbif-1)# exit
Brocade(config)# snmp-server trap-source loopback 1
```

The commands in this example configure loopback interface 1, assign IP address 10.00.1/24 to the loopback interface, then designate the interface as the SNMP trap source for this device. Regardless of the port the Brocade device uses to send traps to the receiver, the traps always arrive from the same source IP address.

Syntax: [no] **snmp-server trap-source ethernet** [<slotnum>/<portnum> | **loopback** <num> | **ve** <num>]

The <slotnum> variable is required on chassis devices.

The <portnum> variable is a valid port number.

The <num> variable is a loopback interface or virtual interface number.

ARP parameter configuration

Address Resolution Protocol (ARP) is a standard IP protocol that enables an IP Layer 3 Switch to obtain the MAC address of another device interface when the Layer 3 Switch knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

NOTE

Brocade Layer 2 Switches also support ARP. The description in [“How ARP works”](#) also applies to ARP on Brocade Layer 2 Switches. However, the configuration options described later in this section apply only to Layer 3 Switches, not to Layer 2 Switches.

How ARP works

A Layer 3 Switch needs to know a destination MAC address when forwarding traffic, because the Layer 3 Switch encapsulates the IP packet in a Layer 2 packet (MAC layer packet) and sends the Layer 2 packet to a MAC interface on a device directly attached to the Layer 3 Switch. The device can be the packet final destination or the next-hop router toward the destination.

The Layer 3 Switch encapsulates IP packets in Layer 2 packets regardless of whether the ultimate destination is locally attached or is multiple router hops away. Since the Layer 3 Switch IP route table and IP forwarding cache contain IP address information but not MAC address information, the Layer 3 Switch cannot forward IP packets based solely on the information in the route table or forwarding cache. The Layer 3 Switch needs to know the MAC address that corresponds with the IP address of either the packet locally attached destination or the next-hop router that leads to the destination.

For example, to forward a packet whose destination is multiple router hops away, the Layer 3 Switch must send the packet to the next-hop router toward its destination, or to a default route or default network route if the IP route table does not contain a route to the packet destination. In each case, the Layer 3 Switch must encapsulate the packet and address it to the MAC address of a locally attached device, the next-hop router toward the IP packet destination.

To obtain the MAC address required for forwarding a datagram, the Layer 3 Switch does the following:

- First, the Layer 3 Switch looks in the ARP cache (not the static ARP table) for an entry that lists the MAC address for the IP address. The ARP cache maps IP addresses to MAC addresses. The cache also lists the port attached to the device and, if the entry is dynamic, the age of the entry. A dynamic ARP entry enters the cache when the Layer 3 Switch receives an ARP reply or receives an ARP request (which contains the sender IP address and MAC address). A static entry enters the ARP cache from the static ARP table (which is a separate table) when the interface for the entry comes up.

To ensure the accuracy of the ARP cache, each dynamic entry has its own age timer. The timer is reset to zero each time the Layer 3 Switch receives an ARP reply or ARP request containing the IP address and MAC address of the entry. If a dynamic entry reaches its maximum allowable age, the entry times out and the software removes the entry from the table. Static entries do not age out and can be removed only by you.

- If the ARP cache does not contain an entry for the destination IP address, the Layer 3 Switch broadcasts an ARP request out all its IP interfaces. The ARP request contains the IP address of the destination. If the device with the IP address is directly attached to the Layer 3 Switch, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the Layer 3 Switch. The Layer 3 Switch places the information from the ARP response into the ARP cache.

ARP requests contain the IP address and MAC address of the sender, so all devices that receive the request learn the MAC address and IP address of the sender and can update their own ARP caches accordingly.

NOTE

The ARP request broadcast is a MAC broadcast, which means the broadcast goes only to devices that are directly attached to the Layer 3 Switch. A MAC broadcast is not routed to other networks. However, some routers, including Brocade Layer 3 Switches, can be configured to reply to ARP requests from one network on behalf of devices on another network. Refer to [“Enabling proxy ARP”](#) on page 983.

NOTE

If the router receives an ARP request packet that it is unable to deliver to the final destination because of the ARP timeout and no ARP response is received (the Layer 3 Switch knows of no route to the destination address), the router sends an ICMP Host Unreachable message to the source.

Rate limiting ARP packets

You can limit the number of ARP packets the Brocade device accepts during each second. By default, the software does not limit the number of ARP packets the device can receive. Since the device sends ARP packets to the CPU for processing, if a device in a busy network receives a high number of ARP packets in a short period of time, some CPU processing might be deferred while the CPU processes the ARP packets.

To prevent the CPU from becoming flooded by ARP packets in a busy network, you can restrict the number of ARP packets the device will accept each second. When you configure an ARP rate limit, the device accepts up to the maximum number of packets you specify, but drops additional ARP packets received during the one-second interval. When a new one-second interval starts, the counter restarts at zero, so the device again accepts up to the maximum number of ARP packets you specified, but drops additional packets received within the interval.

To limit the number of ARP packets the device will accept each second, enter the **rate-limit-arp** command at the global CONFIG level of the CLI.

```
Brocade(config)# rate-limit-arp 100
```

This command configures the device to accept up to 100 ARP packets each second. If the device receives more than 100 ARP packets during a one-second interval, the device drops the additional ARP packets during the remainder of that one-second interval.

Syntax: [no] **rate-limit-arp** <num>

The <num> parameter specifies the number of ARP packets and can be from 0 through 100. If you specify 0, the device will not accept any ARP packets.

NOTE

If you want to change a previously configured the ARP rate limiting policy, you must remove the previously configured policy using the **no rate-limit-arp** <num> command before entering the new policy.

Changing the ARP aging period

When the Layer 3 Switch places an entry in the ARP cache, the Layer 3 Switch also starts an aging timer for the entry. The aging timer ensures that the ARP cache does not retain learned entries that are no longer valid. An entry can become invalid when the device with the MAC address of the entry is no longer on the network.

The ARP age affects dynamic (learned) entries only, not static entries. The default ARP age is ten minutes. On Layer 3 Switches, you can change the ARP age to a value from 0 through 240 minutes. You cannot change the ARP age on Layer 2 Switches. If you set the ARP age to zero, aging is disabled and entries do not age out.

To globally change the ARP aging parameter to 20 minutes, enter the **ip arp-age** command.

```
Brocade(config)# ip arp-age 20
```

Syntax: **ip arp-age** <num>

The <num> parameter specifies the number of minutes and can be from 0 through 240. The default is 10. If you specify 0, aging is disabled.

To override the globally configured IP ARP age on an individual interface, enter a command such as the following at the interface configuration level.

```
Brocade(config-if-e1000-1/1)# ip arp-age 30
```

Syntax: [no] **ip arp-age** <num>

The <num> parameter specifies the number of minutes and can be from 0 through 240. The default is the globally configured value, which is 10 minutes by default. If you specify 0, aging is disabled.

Enabling proxy ARP

Proxy ARP allows a Layer 3 Switch to answer ARP requests from devices on one network on behalf of devices in another network. Since ARP requests are MAC-layer broadcasts, they reach only the devices that are directly connected to the sender of the ARP request. Thus, ARP requests do not cross routers.

For example, if Proxy ARP is enabled on a Layer 3 Switch connected to two subnets, 10.10.10.0/24 and 20.20.20.0/24, the Layer 3 Switch can respond to an ARP request from 10.10.10.69 for the MAC address of the device with IP address 20.20.20.69. In standard ARP, a request from a device in the 10.10.10.0/24 subnet cannot reach a device in the 20.20.20.0 subnet if the subnets are on different network cables, and thus is not answered.

NOTE

An ARP request from one subnet can reach another subnet when both subnets are on the same physical segment (Ethernet cable), because MAC-layer broadcasts reach all the devices on the segment.

Proxy ARP is disabled by default on Brocade Layer 3 Switches. This feature is not supported on Brocade Layer 2 Switches.

You can enable proxy ARP at the Interface level, as well as at the Global CONFIG level, of the CLI.

NOTE

Configuring proxy ARP at the Interface level overrides the global configuration.

Enabling proxy ARP globally

To enable IP proxy ARP on a global basis, enter the **ip proxy-arp** command.

```
Brocade(config)# ip proxy-arp
```

To again disable IP proxy ARP on a global basis, enter the **no ip proxy-arp** command.

```
Brocade(config)# no ip proxy-arp
```

Syntax: [no] ip proxy-arp

Enabling IP ARP on an interface

NOTE

Configuring proxy ARP at the Interface level overrides the global configuration.

To enable IP proxy ARP on an interface, enter the following commands.

```
Brocade(config)# interface ethernet 5
Brocade(config-if-e1000-5)# ip proxy-arp enable
```

To again disable IP proxy ARP on an interface, enter the following command.

```
Brocade(config)# interface ethernet 5
Brocade(config-if-e1000-5)# ip proxy-arp disable
```

Syntax: [no] ip proxy-arp enable | disable

Enabling local proxy ARP

Brocade devices support Proxy Address Resolution Protocol (**Proxy ARP**), a feature that enables router ports to respond to ARP requests for subnets it can reach. However, router ports will not respond to ARP requests for IP addresses in the same subnet as the incoming ports, unless Local Proxy ARP per IP interface is enabled. **Local Proxy ARP** enables router ports to reply to ARP requests for IP addresses within the same subnet and to forward all traffic between hosts in the subnet.

When Local Proxy ARP is enabled on a router port, the port will respond to ARP requests for IP addresses within the same subnet, if it has ARP entries for the destination IP addresses in the ARP cache. If it does not have ARP entries for the IP addresses, the port will attempt to resolve them by broadcasting its own ARP requests.

Local Proxy ARP is disabled by default. To use Local Proxy ARP, Proxy ARP (**ip proxy-arp** command) must be enabled globally on the Brocade device. You can enter the CLI command to enable Local Proxy ARP even though Proxy ARP is not enabled, however, the configuration will not take effect until you enable Proxy ARP.

Use the **show run** command to view the ports on which Local Proxy ARP is enabled.

To enable Local Proxy ARP, enter commands such as the following.

```
Brocade(config)# interface ethernet 4
Brocade(config-if-e1000-4)# ip local-proxy-arp
```

Syntax: [no] **ip local-proxy-arp**

Use the **no** form of the command to disable Local Proxy ARP.

Creating static ARP entries

Brocade Layer 3 Switches have a static ARP table, in addition to the regular ARP cache. The static ARP table contains entries that you configure.

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the Layer 3 Switch, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the Brocade device receives an ARP request from the device that has the entry address.

NOTE

You cannot create static ARP entries on a Layer 2 Switch.

The maximum number of static ARP entries you can configure depends on the software version running on the device. Refer to [“Changing the maximum number of entries the static ARP table can hold”](#) on page 986.

To display the ARP cache and static ARP table, refer to the following:

- To display the ARP table, refer to [“Displaying the ARP cache”](#) on page 1070.
- To display the static ARP table, refer to [“Displaying the static ARP table”](#) on page 1072.

To create a static ARP entry, enter a command such as the following.

```
Brocade(config)# arp 1 192.53.4.2 1245.7654.2348 ethernet 1/2
```

Syntax: **arp** <num> <ip-addr> <mac-addr> **ethernet** <port>

The <num> parameter specifies the entry number. You can specify a number from 1 up to the maximum number of static entries allowed on the device.

The <ip-addr> parameter specifies the IP address of the device that has the MAC address of the entry.

The <mac-addr> parameter specifies the MAC address of the entry.

The **ethernet** <port> command specifies the port number attached to the device that has the MAC address of the entry. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Changing the maximum number of entries the static ARP table can hold

[Table 168](#) on page 986 lists the default maximum and configurable maximum number of entries in the static ARP table that are supported on a Brocade Layer 3 Switch. If you need to change the maximum number of entries supported on a Layer 3 Switch, use the method described in this section.

NOTE

The basic procedure for changing the static ARP table size is the same as the procedure for changing other configurable cache or table sizes. Refer to the section [“Displaying system parameter default values”](#) on page 578.

To increase the maximum number of static ARP table entries you can configure on a Brocade Layer 3 Switch, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# system-max ip-static-arp 1000
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload
```

NOTE

You must save the configuration to the startup-config file and reload the software after changing the static ARP table size to place the change into effect.

Syntax: `system-max ip-static-arp <num>`

The `<num>` parameter indicates the maximum number of static ARP entries and can be within one of the ranges shown in [Table 168](#), depending on the software version running on the device.

TABLE 168 Static ARP entry support

Default maximum	Configurable minimum	Configurable maximum
FastIron X Series and Brocade FCX Series devices		
512	512	6000
ICX 6450 devices		
256	64	1024

Configuring forwarding parameters

The following configurable parameters control the forwarding behavior of Brocade Layer 3 Switches:

- Time-To-Live (TTL) threshold
- Forwarding of directed broadcasts
- Forwarding of source-routed packets
- Ones-based and zero-based broadcasts

All these parameters are global and thus affect all IP interfaces configured on the Layer 3 Switch.

To configure these parameters, use the procedures in the following sections.

Changing the TTL threshold

The time to live (TTL) threshold prevents routing loops by specifying the maximum number of router hops an IP packet originated by the Layer 3 Switch can travel through. Each device capable of forwarding IP that receives the packet decrements (decreases) the packet TTL by one. If a device receives a packet with a TTL of 1 and reduces the TTL to zero, the device drops the packet.

The default TTL is 64. You can change the TTL to a value from 1 through 255.

To modify the TTL threshold to 25, enter the **ip ttl** command.

```
Brocade(config)# ip ttl 25
```

Syntax: **ip ttl** <1-255>

Enabling forwarding of directed broadcasts

A directed broadcast is an IP broadcast to all devices within a single directly-attached network or subnet. A net-directed broadcast goes to all devices on a given network. A subnet-directed broadcast goes to all devices within a given subnet.

NOTE

A less common type, the all-subnets broadcast, goes to all directly-attached subnets. Forwarding for this broadcast type also is supported, but most networks use IP multicasting instead of all-subnet broadcasting.

Forwarding for all types of IP directed broadcasts is disabled by default. You can enable forwarding for all types if needed. You cannot enable forwarding for specific broadcast types.

To enable forwarding of IP directed broadcasts, enter the **ip directed-broadcast** command.

```
Brocade(config)# ip directed-broadcast
```

Syntax: **[no] ip directed-broadcast**

Brocade software makes the forwarding decision based on the router's knowledge of the destination network prefix. Routers cannot determine that a message is unicast or directed broadcast apart from the destination network prefix. The decision to forward or not forward the message is by definition only possible in the last hop router.

To disable the directed broadcasts, enter the **no ip directed-broadcast** command in the CONFIG mode.

```
Brocade(config)# no ip directed-broadcast
```

To enable directed broadcasts on an individual interface instead of globally for all interfaces, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1
Brocade(config-if-1/1)# ip directed-broadcast
```

Syntax: **[no] ip directed-broadcast**

Disabling forwarding of IP source-routed packets

A source-routed packet specifies the exact router path for the packet. The packet specifies the path by listing the IP addresses of the router interfaces through which the packet must pass on its way to the destination. The Layer 3 Switch supports both types of IP source routing:

- Strict source routing – requires the packet to pass through only the listed routers. If the Layer 3 Switch receives a strict source-routed packet but cannot reach the next hop interface specified by the packet, the Layer 3 Switch discards the packet and sends an ICMP Source-Route-Failure message to the sender.

NOTE

The Layer 3 Switch allows you to disable sending of the Source-Route-Failure messages. Refer to [“Disabling ICMP messages”](#) on page 989.

- Loose source routing – requires that the packet pass through all of the listed routers but also allows the packet to travel through other routers, which are not listed in the packet.

The Layer 3 Switch forwards both types of source-routed packets by default. To disable the feature, use either of the following methods. You cannot enable or disable strict or loose source routing separately.

To disable forwarding of IP source-routed packets, enter the **no ip source-route** command.

```
Brocade(config)# no ip source-route
```

Syntax: [no] ip source-route

To re-enable forwarding of source-routed packets, enter the **ip source-route** command.

```
Brocade(config)# ip source-route
```

Enabling support for zero-based IP subnet broadcasts

By default, the Layer 3 Switch treats IP packets with all ones in the host portion of the address as IP broadcast packets. For example, the Layer 3 Switch treats IP packets with 209.157.22.255/24 as the destination IP address as IP broadcast packets and forwards the packets to all IP hosts within the 209.157.22.x subnet (except the host that sent the broadcast packet to the Layer 3 Switch).

Most IP hosts are configured to receive IP subnet broadcast packets with all ones in the host portion of the address. However, some older IP hosts instead expect IP subnet broadcast packets that have all zeros instead of all ones in the host portion of the address. To accommodate this type of host, you can enable the Layer 3 Switch to treat IP packets with all zeros in the host portion of the destination IP address as broadcast packets.

NOTE

When you enable the Layer 3 Switch for zero-based subnet broadcasts, the Layer 3 Switch still treats IP packets with all ones the host portion as IP subnet broadcasts too. Thus, the Layer 3 Switch can be configured to support all ones only (the default) or all ones **and** all zeroes.

NOTE

This feature applies only to IP subnet broadcasts, not to local network broadcasts. The local network broadcast address is still expected to be all ones.

To enable the Layer 3 Switch for zero-based IP subnet broadcasts in addition to ones-based IP subnet broadcasts, enter the following command.

```

Brocade(config)# ip broadcast-zero
Brocade(config)# write memory
Brocade(config)# end
Brocade# reload

```

NOTE

You must save the configuration and reload the software to place this configuration change into effect.

Syntax: [no] ip broadcast-zero

Disabling ICMP messages

Brocade devices are enabled to reply to ICMP echo messages and send ICMP Destination Unreachable messages by default.

You can selectively disable the following types of Internet Control Message Protocol (ICMP) messages:

- **Echo messages (ping messages)** – The Layer 3 Switch replies to IP pings from other IP devices.
- **Destination Unreachable messages** – If the Layer 3 Switch receives an IP packet that it cannot deliver to its destination, the Layer 3 Switch discards the packet and sends a message back to the device that sent the packet to the Layer 3 Switch. The message informs the device that the destination cannot be reached by the Layer 3 Switch.

Disabling replies to broadcast ping requests

By default, Brocade devices are enabled to respond to broadcast ICMP echo packets, which are ping requests.

To disable response to broadcast ICMP echo packets (ping requests), enter the following command.

```
Brocade(config)# no ip icmp echo broadcast-request
```

Syntax: [no] ip icmp echo broadcast-request

If you need to re-enable response to ping requests, enter the following command.

```
Brocade(config)# ip icmp echo broadcast-request
```

Disabling ICMP destination unreachable messages

By default, when a Brocade device receives an IP packet that the device cannot deliver, the device sends an ICMP Unreachable message back to the host that sent the packet. You can selectively disable a Brocade device response to the following types of ICMP Unreachable messages:

- **Administration** – The packet was dropped by the Brocade device due to a filter or ACL configured on the device.
- **Fragmentation-needed** – The packet has the Do not Fragment bit set in the IP Flag field, but the Brocade device cannot forward the packet without fragmenting it.
- **Host** – The destination network or subnet of the packet is directly connected to the Brocade device, but the host specified in the destination IP address of the packet is not on the network.
- **Port** – The destination host does not have the destination TCP or UDP port specified in the packet. In this case, the host sends the ICMP Port Unreachable message to the Brocade device, which in turn sends the message to the host that sent the packet.

- **Protocol** – The TCP or UDP protocol on the destination host is not running. This message is different from the Port Unreachable message, which indicates that the protocol is running on the host but the requested protocol port is unavailable.
- **Source-route-failure** – The device received a source-routed packet but cannot locate the next-hop IP address indicated in the packet Source-Route option.

You can disable the Brocade device from sending these types of ICMP messages on an individual basis. To do so, use the following CLI method.

NOTE

Disabling an ICMP Unreachable message type does not change the Brocade device ability to forward packets. Disabling ICMP Unreachable messages prevents the device from generating or forwarding the Unreachable messages.

To disable all ICMP Unreachable messages, enter the **no ip icmp unreachable** command.

```
Brocade(config)# no ip icmp unreachable
```

Syntax: [no] ip icmp unreachable [host | protocol | administration | fragmentation-needed | port | source-route-fail]

- If you enter the command without specifying a message type (as in the example above), all types of ICMP Unreachable messages listed above are disabled. If you want to disable only specific types of ICMP Unreachable messages, you can specify the message type. To disable more than one type of ICMP message, enter the **no ip icmp unreachable** command for each messages type.
- The **administration** parameter disables ICMP Unreachable (caused by Administration action) messages.
- The **fragmentation-needed** parameter disables ICMP Fragmentation-Needed But Do not-Fragment Bit Set messages.
- The **host** parameter disables ICMP Host Unreachable messages.
- The **port** parameter disables ICMP Port Unreachable messages.
- The **protocol** parameter disables ICMP Protocol Unreachable messages.
- The **source-route-fail** parameter disables ICMP Unreachable (caused by Source-Route-Failure) messages.

To disable ICMP Host Unreachable messages but leave the other types of ICMP Unreachable messages enabled, enter the following commands instead of the command shown above.

```
Brocade(config)# no ip icmp unreachable host
```

If you have disabled all ICMP Unreachable message types but you want to re-enable certain types, for example ICMP Host Unreachable messages, you can do so by entering the following command.

```
Brocade(config)# ip icmp unreachable host
```

Disabling ICMP Redirect Messages

You can disable or re-enable ICMP redirect messages. By default, a Brocade Layer 3 Switch sends an ICMP redirect message to the source of a misdirected packet in addition to forwarding the packet to the appropriate router. You can disable ICMP redirect messages on a global basis or on an individual port basis.

NOTE

FESX and FSX devices do not generate ICMP redirect and network unreachable messages.

NOTE

The device forwards misdirected traffic to the appropriate router, even if you disable the redirect messages.

To disable ICMP redirect messages globally, enter the following command at the global CONFIG level of the CLI:

```
Brocade(config)# no ip icmp redirect
```

Syntax: [no] ip icmp redirects

To disable ICMP redirect messages on a specific interface, enter the following command at the configuration level for the interface:

```
Brocade(config)# interface ethernet 3/11
Brocade(config-if-e1000-3/11)# no ip redirect
```

Syntax: [no] ip redirect

Static routes configuration

The IP route table can receive routes from the following sources:

- **Directly-connected networks** – When you add an IP interface, the Layer 3 Switch automatically creates a route for the network the interface is in.
- **RIP** – If RIP is enabled, the Layer 3 Switch can learn about routes from the advertisements other RIP routers send to the Layer 3 Switch. If the route has a lower administrative distance than any other routes from different sources to the same destination, the Layer 3 Switch places the route in the IP route table.
- **OSPF** – Refer to RIP, but substitute “OSPF” for “RIP”.
- **BGP4** – Refer to RIP, but substitute “BGP4” for “RIP”.
- **Default network route** – A statically configured default route that the Layer 3 Switch uses if other default routes to the destination are not available. Refer to [“Configuring a default network route”](#) on page 1000.
- **Statically configured route** – You can add routes directly to the route table. When you add a route to the IP route table, you are creating a static IP route. This section describes how to add static routes to the IP route table.

Static route types

You can configure the following types of static IP routes:

- **Standard** – the static route consists of the destination network address and network mask, and the IP address of the next-hop gateway. You can configure multiple standard static routes with the same metric for load sharing or with different metrics to provide a primary route and backup routes.
- **Interface-based** – the static route consists of the destination network address and network mask, and the Layer 3 Switch interface through which you want the Layer 3 Switch to send traffic for the route. Typically, this type of static route is for directly attached destination networks.

- **Null** – the static route consists of the destination network address and network mask, and the “null0” parameter. Typically, the null route is configured as a backup route for discarding traffic if the primary route is unavailable.

Static IP route parameters

When you configure a static IP route, you must specify the following parameters:

- The IP address and network mask for the route destination network.
- The route path, which can be one of the following:
 - The IP address of a next-hop gateway
 - An Ethernet port
 - A virtual interface (a routing interface used by VLANs for routing Layer 3 protocol traffic among one another)
 - A “null” interface. The Layer 3 Switch drops traffic forwarded to the null interface.

You also can specify the following optional parameters:

- The metric for the route – The value the Layer 3 Switch uses when comparing this route to other routes in the IP route table to the same destination. The metric applies only to routes that the Layer 3 Switch has already placed in the IP route table. The default metric for static IP routes is 1.
- The administrative distance for the route – The value that the Layer 3 Switch uses to compare this route with routes from other route sources to the same destination before placing a route in the IP route table. This parameter does not apply to routes that are already in the IP route table. The default administrative distance for static IP routes is 1.

The default metric and administrative distance values ensure that the Layer 3 Switch always prefers static IP routes over routes from other sources to the same destination.

Multiple static routes to the same destination provide load sharing and redundancy

You can add multiple static routes for the same destination network to provide one or more of the following benefits:

- **IP load balancing** – When you add multiple IP static routes for the same destination to different next-hop gateways, and the routes each have the same metric and administrative distance, the Layer 3 Switch can load balance traffic to the routes’ destination. For information about IP load balancing, refer to [“Configuring IP load sharing”](#) on page 1001.
- **Path redundancy** – When you add multiple static IP routes for the same destination, but give the routes different metrics or administrative distances, the Layer 3 Switch uses the route with the lowest administrative distance by default, but uses another route to the same destination if the first route becomes unavailable.

Refer to the following sections for examples and configuration information:

- [“Configuring load balancing and redundancy using multiple static routes to the same destination”](#) on page 996
- [“Configuring standard static IP routes and interface or null static routes to the same destination”](#) on page 997

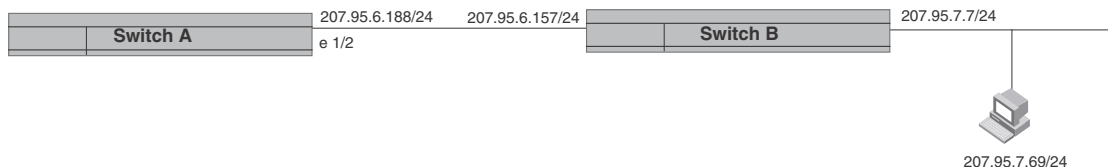
Static route states follow port states

IP static routes remain in the IP route table only so long as the port or virtual interface used by the route is available. If the port or virtual routing interface becomes unavailable, the software removes the static route from the IP route table. If the port or virtual routing interface becomes available again later, the software adds the route back to the route table.

This feature allows the Layer 3 Switch to adjust to changes in network topology. The Layer 3 Switch does not continue trying to use routes on unavailable paths but instead uses routes only when their paths are available.

Figure 113 shows an example of a network containing a static route. The static route is configured on Switch A, as shown in the CLI example following the figure.

FIGURE 113 Example of a static route



The following command configures a static route to 207.95.7.0, using 207.95.6.157 as the next-hop gateway.

```
Brocade(config)# ip route 207.95.7.0/24 207.95.6.157
```

When you configure a static IP route, you specify the destination address for the route and the next-hop gateway or Layer 3 Switch interface through which the Layer 3 Switch can reach the route. The Layer 3 Switch adds the route to the IP route table. In this case, Switch A knows that 207.95.6.157 is reachable through port 1/2, and also assumes that local interfaces within that subnet are on the same port. Switch A deduces that IP interface 207.95.7.188 is also on port 1/2.

The software automatically removes a static IP route from the IP route table if the port used by that route becomes unavailable. When the port becomes available again, the software automatically re-adds the route to the IP route table.

Configuring a static IP route

To configure an IP static route with a destination address of 192.0.0.0 255.0.0.0 and a next-hop router IP address of 195.1.1.1, enter a command such as the following.

```
Brocade(config)# ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

To configure a static IP route with an Ethernet port instead of a next-hop address, enter a command such as the following.

```
Brocade(config)# ip route 192.128.2.69 255.255.255.0 ethernet 4/1
```

The command in the previous example configures a static IP route for destination network 192.128.2.69/24. Since an Ethernet port is specified instead of a gateway IP address as the next hop, the Layer 3 Switch always forwards traffic for the 192.128.2.69/24 network to port 4/1. The command in the following example configures an IP static route that uses virtual interface 3 as its next hop.

```
Brocade(config)# ip route 192.128.2.71 255.255.255.0 ve 3
```

The command in the following example configures an IP static route that uses port 2/2 as its next hop.

```
Brocade(config)# ip route 192.128.2.73 255.255.255.0 ethernet 2/2
```

Syntax: **ip route** *<dest-ip-addr>* *<dest-mask>*
 <next-hop-ip-addr> |
 ethernet [*<slotnum>/<portnum>*] | **ve** *<num>*
 [*<metric>*] [**distance** *<num>*]

or

Syntax: **ip route** *<dest-ip-addr>/<mask-bits>*
 <next-hop-ip-addr> |
 ethernet [*<slotnum>/<portnum>*] | **ve** *<num>*
 [*<metric>*] [**distance** *<num>*]

The *<dest-ip-addr>* is the route destination. The *<dest-mask>* is the network mask for the route destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24.

The *<next-hop-ip-addr>* is the IP address of the next-hop router (gateway) for the route.

If you do not want to specify a next-hop IP address, you can instead specify a port or interface number on the Layer 3 Switch. The *<num>* parameter is a virtual interface number. If you instead specify an Ethernet port, the *<portnum>* is the port number (including the slot number, if you are configuring a Chassis device). In this case, the Layer 3 Switch forwards packets destined for the static route destination network to the specified interface. Conceptually, this feature makes the destination network like a directly connected network, associated with a specific Layer 3 Switch interface.

NOTE

The port or virtual interface you use for the static route next hop must have at least one IP address configured on it. The address does not need to be in the same subnet as the destination network.

The *<metric>* parameter can be a number from 1 through 16. The default is 1.

NOTE

If you specify 16, RIP considers the metric to be infinite and thus also considers the route to be unreachable.

The **distance** *<num>* parameter specifies the administrative distance of the route. When comparing otherwise equal routes to a destination, the Layer 3 Switch prefers lower administrative distances over higher ones, so make sure you use a low value for your default route. The default is 1.

NOTE

The Layer 3 Switch will replace the static route if the it receives a route with a lower administrative distance. Refer to [“Administrative distance”](#) on page 1265 for a list of the default administrative distances for all types of routes.

NOTE

You can also assign the default router as the destination by entering 0.0.0.0 0.0.0.0 xxx.xxx.xxx.xxx.

Configuring a “Null” route

You can configure the Layer 3 Switch to drop IP packets to a specific network or host address by configuring a “null” (sometimes called “null0”) static route for the address. When the Layer 3 Switch receives a packet destined for the address, the Layer 3 Switch drops the packet instead of forwarding it.

To configure a null static route, use the following CLI method.

To configure a null static route to drop packets destined for network 209.157.22.x, enter the following commands.

```
Brocade(config)# ip route 209.157.22.0 255.255.255.0 null0
Brocade(config)# write memory
```

Syntax: `ip route <ip-addr> <ip-mask> null0 [<metric>] [distance <num>]`

or

Syntax: `ip route <ip-addr>/<mask-bits> null0 [<metric>] [distance <num>]`

To display the maximum value for your device, enter the **show default values** command. The maximum number of static IP routes the system can hold is listed in the ip-static-route row in the System Parameters section of the display. To change the maximum value, use the **system-max ip-static-route <num>** command at the global CONFIG level.

The `<ip-addr>` parameter specifies the network or host address. The Layer 3 Switch will drop packets that contain this address in the destination field instead of forwarding them.

The `<ip-mask>` parameter specifies the network mask. Ones are significant bits and zeros allow any value. For example, the mask 255.255.255.0 matches on all hosts within the Class C subnet address specified by `<ip-addr>`. Alternatively, you can specify the number of bits in the network mask. For example, you can enter 209.157.22.0/24 instead of 209.157.22.0 255.255.255.0.

The **null0** parameter indicates that this is a null route. You must specify this parameter to make this a null route.

The `<metric>` parameter adds a cost to the route. You can specify from 1 through 16. The default is 1.

The distance `<num>` parameter configures the administrative distance for the route. You can specify a value from 1 through 255. The default is 1. The value 255 makes the route unusable.

NOTE

The last two parameters are optional and do not affect the null route, unless you configure the administrative distance to be 255. In this case, the route is not used and the traffic might be forwarded instead of dropped.

Configuring load balancing and redundancy using multiple static routes to the same destination

You can configure multiple static IP routes to the same destination, for the following benefits:

- **IP load sharing** – If you configure more than one static route to the same destination, and the routes have different next-hop gateways but have the same metrics, the Layer 3 Switch load balances among the routes using basic round-robin. For example, if you configure two static routes with the same metrics but to different gateways, the Layer 3 Switch alternates between the two routes. For information about IP load balancing, refer to [“Configuring IP load sharing”](#) on page 1001.
- **Backup Routes** – If you configure multiple static IP routes to the same destination, but give the routes different next-hop gateways and different metrics, the Layer 3 Switch will always use the route with the lowest metric. If this route becomes unavailable, the Layer 3 Switch will fail over to the static route with the next-lowest metric, and so on.

NOTE

You also can bias the Layer 3 Switch to select one of the routes by configuring them with different administrative distances. However, make sure you do not give a static route a higher administrative distance than other types of routes, unless you want those other types to be preferred over the static route. For a list of the default administrative distances, refer to [“Administrative distance”](#) on page 1265.

The steps for configuring the static routes are the same as described in the previous section. The following sections provide examples.

To configure multiple static IP routes, enter commands such as the following.

```
Brocade(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
Brocade(config)# ip route 192.128.2.69 255.255.255.0 192.111.10.1
```

The commands in the previous example configure two static IP routes. The routes go to different next-hop gateways but have the same metrics. These commands use the default metric value (1), so the metric is not specified. These static routes are used for load sharing among the next-hop gateways.

The following commands configure static IP routes to the same destination, but with different metrics. The route with the lowest metric is used by default. The other routes are backups in case the first route becomes unavailable. The Layer 3 Switch uses the route with the lowest metric if the route is available.

```
Brocade(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
Brocade(config)# ip route 192.128.2.69 255.255.255.0 192.111.10.1 2
Brocade(config)# ip route 192.128.2.69 255.255.255.0 201.1.1.1 3
```

In this example, each static route has a different metric. The metric is not specified for the first route, so the default (1) is used. A metric is specified for the second and third static IP routes. The second route has a metric of two and the third route has a metric of 3. Thus, the second route is used only if the first route (which has a metric of 1) becomes unavailable. Likewise, the third route is used only if the first and second routes (which have lower metrics) are both unavailable.

For complete syntax information, refer to [“Configuring a static IP route”](#) on page 993.

Configuring standard static IP routes and interface or null static routes to the same destination

You can configure a null0 or interface-based static route to a destination and also configure a normal static route to the same destination, so long as the route metrics are different.

When the Layer 3 Switch has multiple routes to the same destination, the Layer 3 Switch always prefers the route with the lowest metric. Generally, when you configure a static route to a destination network, you assign the route a low metric so that the Layer 3 Switch prefers the static route over other routes to the destination.

This feature is especially useful for the following configurations. These are not the only allowed configurations but they are typical uses of this enhancement:

- When you want to ensure that if a given destination network is unavailable, the Layer 3 Switch drops (forwards to the null interface) traffic for that network instead of using alternate paths to route the traffic. In this case, assign the normal static route to the destination network a lower metric than the null route.
- When you want to use a specific interface by default to route traffic to a given destination network, but want to allow the Layer 3 Switch to use other interfaces to reach the destination network if the path that uses the default interface becomes unavailable. In this case, give the interface route a lower metric than the normal static route.

NOTE

You cannot add a null or interface-based static route to a network if there is already a static route of any type with the same metric you specify for the null or interface-based route.

Figure 114 shows an example of two static routes configured for the same destination network. In this example, one of the routes is a standard static route and has a metric of 1. The other static route is a null route and has a higher metric than the standard static route. The Layer 3 Switch always prefers the static route with the lower metric. In this example, the Layer 3 Switch always uses the standard static route for traffic to destination network 192.168.7.0/24, unless that route becomes unavailable, in which case the Layer 3 Switch sends traffic to the null route instead.

FIGURE 114 Standard and null static routes to the same destination network

Configuring IP parameters – Layer 3 Switches

Two static routes to 192.168.7.0/24:

--Standard static route through gateway 192.168.6.157, with metric 1

--Null route, with metric 2

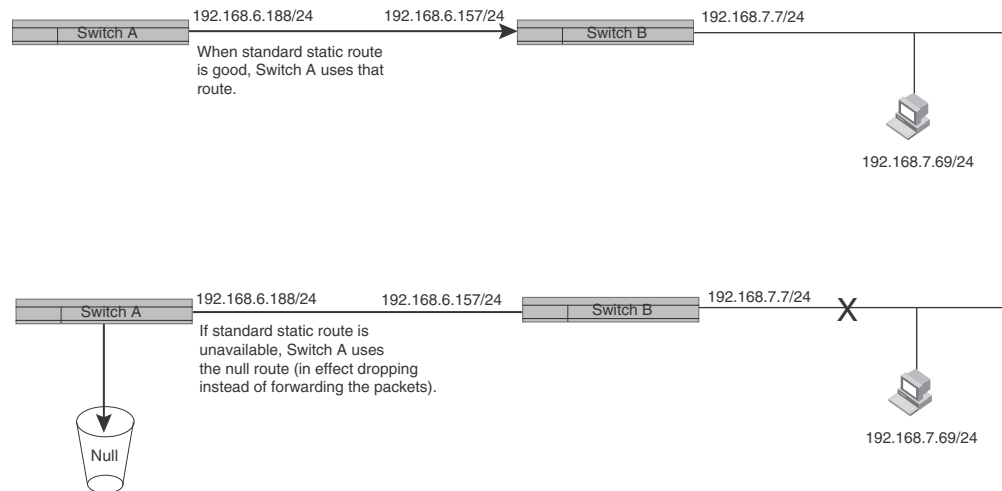
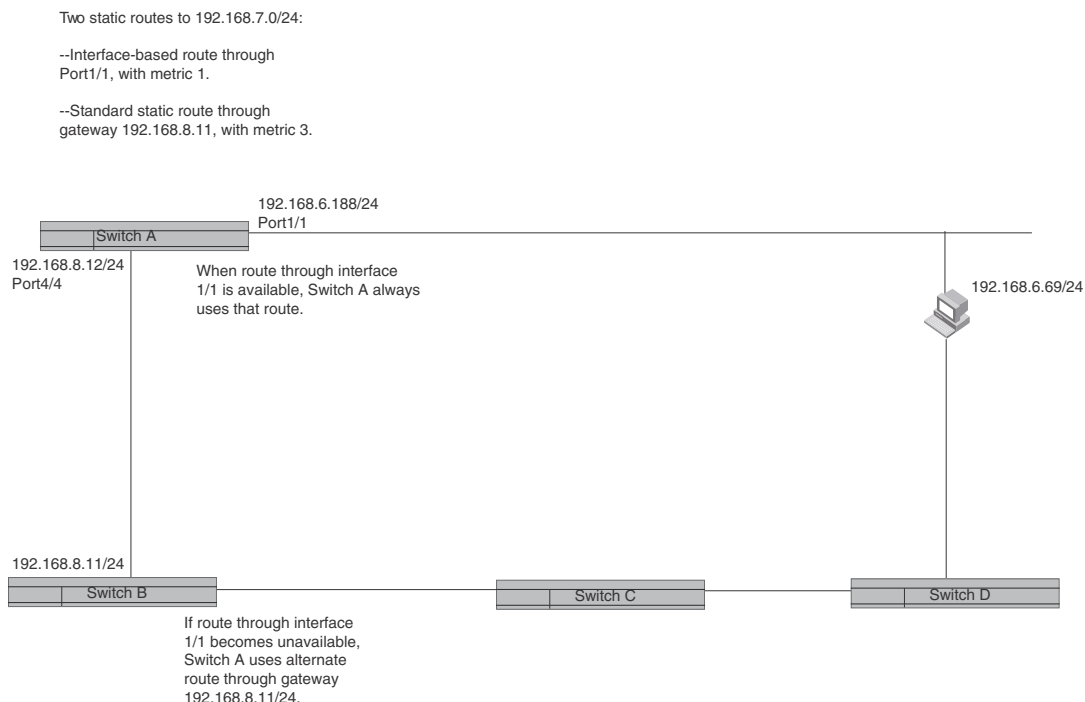


Figure 115 shows another example of two static routes. In this example, a standard static route and an interface-based static route are configured for destination network 192.168.6.0/24. The interface-based static route has a lower metric than the standard static route. As a result, the Layer 3 Switch always prefers the interface-based route when the route is available. However, if the interface-based route becomes unavailable, the Layer 3 Switch still forwards the traffic toward the destination using an alternate route through gateway 192.168.8.11/24.

FIGURE 115 Standard and interface routes to the same destination network

To configure a standard static IP route and a null route to the same network as shown in [Figure 114](#) on page 997, enter commands such as the following.

```
Brocade(config)# ip route 192.168.7.0/24 192.168.6.157/24 1
Brocade(config)# ip route 192.168.7.0/24 null0 3
```

The first command configures a standard static route, which includes specification of the next-hop gateway. The command also gives the standard static route a metric of 1, which causes the Layer 3 Switch to always prefer this route when the route is available.

The second command configures another static route for the same destination network, but the second route is a null route. The metric for the null route is 3, which is higher than the metric for the standard static route. If the standard static route is unavailable, the software uses the null route.

For complete syntax information, refer to [“Configuring a static IP route”](#) on page 993.

To configure a standard static route and an interface-based route to the same destination, enter commands such as the following.

```
Brocade(config)# ip route 192.168.6.0/24 ethernet 1/1 1
Brocade(config)# ip route 192.168.6.0/24 192.168.8.11/24 3
```

The first command configured an interface-based static route through Ethernet port 1/1. The command assigns a metric of 1 to this route, causing the Layer 3 Switch to always prefer this route when it is available. If the route becomes unavailable, the Layer 3 Switch uses an alternate route through the next-hop gateway 192.168.8.11/24.

Configuring a default network route

The Layer 3 Switch enables you to specify a candidate default route without the need to specify the next hop gateway. If the IP route table does not contain an explicit default route (for example, 0.0.0.0/0) or propagate an explicit default route through routing protocols, the software can use the default network route as a default route instead.

When the software uses the default network route, it also uses the default network route's next hop gateway as the gateway of last resort.

This feature is especially useful in environments where network topology changes can make the next hop gateway unreachable. This feature allows the Layer 3 Switch to perform default routing even if the default network route's default gateway changes.

The feature thus differs from standard default routes. When you configure a standard default route, you also specify the next hop gateway. If a topology change makes the gateway unreachable, the default route becomes unusable.

For example, if you configure 10.10.10.0/24 as a candidate default network route, if the IP route table does not contain an explicit default route (0.0.0.0/0), the software uses the default network route and automatically uses that route's next hop gateway as the default gateway. If a topology change occurs and as a result the default network route's next hop gateway changes, the software can still use the default network route. To configure a default network route, use the following CLI method.

If you configure more than one default network route, the Layer 3 Switch uses the following algorithm to select one of the routes.

1. Use the route with the lowest administrative distance.
2. If the administrative distances are equal:
 - Are the routes from different routing protocols (RIP, OSPF, or BGP4)? If so, use the route with the lowest IP address.
 - If the routes are from the same routing protocol, use the route with the best metric. The meaning of “best” metric depends on the routing protocol:
 - **RIP** – The metric is the number of hops (additional routers) to the destination. The best route is the route with the fewest hops.
 - **OSPF** – The metric is the path cost associated with the route. The path cost does not indicate the number of hops but is instead a numeric value associated with each route. The best route is the route with the lowest path cost.
 - **BGP4** – The metric is the Multi-exit Discriminator (MED) associated with the route. The MED applies to routes that have multiple paths through the same AS. The best route is the route with the lowest MED.

Configuring a default network route

You can configure up to four default network routes.

To configure a default network route, enter commands such as the following.

```
Brocade(config)# ip default-network 209.157.22.0
Brocade(config)# write memory
```

Syntax: `ip default-network <ip-addr>`

The `<ip-addr>` parameter specifies the network address.

To verify that the route is in the route table, enter the following command at any level of the CLI.

```
Brocade# show ip route
Total number of IP routes: 2
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF  *:Candidate default
      Destination      NetMask      Gateway      Port    Cost    Type
1      209.157.20.0      255.255.255.0  0.0.0.0      lb1      1      D
2      209.157.22.0      255.255.255.0  0.0.0.0      4/11     1      *D
```

This example shows two routes. Both of the routes are directly attached, as indicated in the Type column. However, one of the routes is shown as type “*D”, with an asterisk (*). The asterisk indicates that this route is a candidate default network route.

Configuring IP load sharing

The IP route table can contain more than one path to a given destination. When this occurs, the Layer 3 Switch selects the path with the lowest cost as the path for forwarding traffic to the destination. If the IP route table contains more than one path to a destination and the paths each have the lowest cost, then the Layer 3 Switch uses **IP load sharing** to select a path to the destination.¹

IP load sharing uses a hashing algorithm based on the source IP address, destination IP address, and protocol field in the IP header, TCP, and UDP information.

NOTE

IP load sharing is based on next-hop routing, and not on source routing.

NOTE

The term “path” refers to the next-hop router to a destination, not to the entire route to a destination. Thus, when the software compares multiple equal-cost paths, the software is comparing paths that use different next-hop routers, with equal costs, to the same destination.

In many contexts, the terms “route” and “path” mean the same thing. Most of the user documentation uses the term “route” throughout. The term “path” is used in this section to refer to an individual next-hop router to a destination, while the term “route” refers collectively to the multiple paths to the destination. Load sharing applies when the IP route table contains multiple, equal-cost paths to a destination.

NOTE

Brocade devices also perform load sharing among the ports in aggregate links. Refer to [“Trunk group load sharing”](#) on page 705.

How multiple equal-cost paths enter the IP route table

IP load sharing applies to equal-cost paths in the IP route table. Routes that are eligible for load sharing can enter the table from any of the following sources:

- IP static routes
- Routes learned through RIP
- Routes learned through OSPF

1. IP load sharing is also called “Equal-Cost Multi-Path (ECMP)” load sharing or just “ECMP”

- Routes learned through BGP4

Administrative distance for each IP route

The administrative distance is a unique value associated with each type (source) of IP route. Each path has an administrative distance. The administrative distance is not used when performing IP load sharing, but the administrative distance is used when evaluating multiple equal-cost paths to the same destination from different sources, such as RIP, OSPF and so on.

The value of the administrative distance is determined by the source of the route. The Layer 3 Switch is configured with a unique administrative distance value for each IP route source.

When the software receives multiple paths to the same destination and the paths are from different sources, the software compares the administrative distances of the paths and selects the path with the lowest distance. The software then places the path with the lowest administrative distance in the IP route table. For example, if the Layer 3 Switch has a path learned from OSPF and a path learned from RIP for a given destination, only the path with the lower administrative distance enters the IP route table.

Here are the default administrative distances on the Brocade Layer 3 Switch:

- Directly connected – 0 (this value is not configurable)
- Static IP route – 1 (applies to all static routes, including default routes and default network routes)
- Exterior Border Gateway Protocol (EBGP) – 20
- OSPF – 110
- RIP – 120
- Interior Gateway Protocol (IBGP) – 200
- Local BGP – 200
- Unknown – 255 (the router will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the router receives routes for the same network from OSPF and from RIP, the router will prefer the OSPF route by default.

NOTE

You can change the administrative distances individually. Refer to the configuration chapter for the route source for information.

Since the software selects only the path with the lowest administrative distance, and the administrative distance is determined by the path source, IP load sharing does not apply to paths from different route sources. IP load sharing applies only when the IP route table contains multiple paths to the same destination, from the same IP route source.

IP load sharing does not apply to paths that come from different sources.

Path cost

The cost parameter provides a common basis of comparison for selecting from among multiple paths to a given destination. Each path in the IP route table has a cost. When the IP route table contains multiple paths to a destination, the Layer 3 Switch chooses the path with the lowest cost. When the IP route table contains more than one path with the lowest cost to a destination, the Layer 3 Switch uses IP load sharing to select one of the lowest-cost paths.

The source of a path cost value depends on the source of the path:

- **IP static route** – The value you assign to the metric parameter when you configure the route. The default metric is 1. Refer to [“Configuring load balancing and redundancy using multiple static routes to the same destination”](#) on page 996.
- **RIP** – The number of next-hop routers to the destination.
- **OSPF** – The Path Cost associated with the path. The paths can come from any combination of inter-area, intra-area, and external Link State Advertisements (LSAs).
- **BGP4** – The path Multi-Exit Discriminator (MED) value.

NOTE

If the path is redistributed between two or more of the above sources before entering the IP route table, the cost can increase during the redistribution due to settings in redistribution filters.

Static route, OSPF, and BGP4 load sharing

IP load sharing and load sharing for static routes, OSPF routes, and BGP4 routes are individually configured. Multiple equal-cost paths for a destination can enter the IP route table only if the source of the paths is configured to support multiple equal-cost paths. For example, if BGP4 allows only one path with a given cost for a given destination, the BGP4 route table cannot contain equal-cost paths to the destination. Consequently, the IP route table will not receive multiple equal-cost paths from BGP4.

[Table 169](#) lists the default and configurable maximum numbers of paths for each IP route source that can provide equal-cost paths to the IP route table. The table also lists where to find configuration information for the route source load sharing parameters.

The load sharing state for all the route sources is based on the state of IP load sharing. Since IP load sharing is enabled by default on all Brocade Layer 3 Switches, load sharing for static IP routes, RIP routes, OSPF routes, and BGP4 routes also is enabled by default.

TABLE 169 Default load sharing parameters for route sources

Route source	Default maximum number of paths	Maximum number of paths		See...
		FSX	FCX	
Static IP route	4 ¹	6 ¹	8 ¹	page 1004
RIP	4 ¹	6 ¹	8 ¹	page 1004
OSPF	4	6	8	page 1004
BGP4	1	4	4	page 1352

1. This value depends on the value for IP load sharing, and is not separately configurable.

How IP load sharing works

When the Layer 3 Switch receives traffic for a destination and the IP route table contains multiple, equal-cost paths to that destination, the device checks the IP forwarding cache for a forwarding entry for the destination. The IP forwarding cache provides a fast path for forwarding IP traffic, including load-balanced traffic. The cache contains entries that associate a destination host or network with a path (next-hop router).

- If the IP forwarding sharing cache contains a forwarding entry for the destination, the device uses the entry to forward the traffic.

- If the IP load forwarding cache does not contain a forwarding entry for the destination, the software selects a path from among the available equal-cost paths to the destination, then creates a forwarding entry in the cache based on the calculation. Subsequent traffic for the same destination uses the forwarding entry.

Response to path state changes

If one of the load-balanced paths to a cached destination becomes unavailable, or the IP route table receives a new equal-cost path to a cached destination, the software removes the unavailable path from the IP route table. Then the software selects a new path. Disabling or re-enabling load sharing

To disable IP load sharing, enter the following commands.

```
Brocade(config)# no ip load-sharing
```

Syntax: [no] ip load-sharing

Changing the maximum number of ECMP (load sharing) paths

You can change the maximum number of paths the Layer 3 Switch supports to a value from 2 through 8. Table 170 shows the maximum number of paths supported per device.

TABLE 170 Maximum number of ECMP load sharing paths per device

FESX FSX 800 FSX 1600	FWS	FCX
6	2	8

For optimal results, set the maximum number of paths to a value at least as high as the maximum number of equal-cost paths your network typically contains. For example, if the Layer 3 Switch you are configuring for IP load sharing has six next-hop routers, set the maximum paths value to six.

NOTE

If the setting for the maximum number of paths is lower than the actual number of equal-cost paths, the software does not use all the paths for load sharing for RIP routes. Run the **clear ip route** command to fix this issue.

To change the number of IP load sharing paths, enter a command such as the following.

```
Brocade(config)# ip load-sharing 6
```

Syntax: [no] ip load-sharing [<num>]

The <num> parameter specifies the number of paths and can be from 2 through 8, depending on the device you are configuring.

ICMP Router Discovery Protocol configuration

The ICMP Router Discovery Protocol (IRDP) is used by Brocade Layer 3 Switches to advertise the IP addresses of its router interfaces to directly attached hosts. IRDP is disabled by default. You can enable the feature on a global basis or on an individual port basis:

- If you enable the feature globally, all ports use the default values for the IRDP parameters.

- If you leave the feature disabled globally but enable it on individual ports, you also can configure the IRDP parameters on an individual port basis.

NOTE

You can configure IRDP parameters only on an individual port basis. To do so, IRDP must be disabled globally and enabled only on individual ports. You cannot configure IRDP parameters if the feature is globally enabled.

When IRDP is enabled, the Layer 3 Switch periodically sends Router Advertisement messages out the IP interfaces on which the feature is enabled. The messages advertise the Layer 3 Switch IP addresses to directly attached hosts who listen for the messages. In addition, hosts can be configured to query the Layer 3 Switch for the information by sending Router Solicitation messages.

Some types of hosts use the Router Solicitation messages to discover their default gateway. When IRDP is enabled on the Brocade Layer 3 Switch, the Layer 3 Switch responds to the Router Solicitation messages. Some clients interpret this response to mean that the Layer 3 Switch is the default gateway. If another router is actually the default gateway for these clients, leave IRDP disabled on the Brocade Layer 3 Switch.

IRDP parameters

IRDP uses the following parameters. If you enable IRDP on individual ports instead of enabling the feature globally, you can configure these parameters on an individual port basis:

- **Packet type** – The Layer 3 Switch can send Router Advertisement messages as IP broadcasts or as IP multicasts addressed to IP multicast group 224.0.0.1. The packet type is IP broadcast.
- **Maximum message interval and minimum message interval** – When IRDP is enabled, the Layer 3 Switch sends the Router Advertisement messages every 450 – 600 seconds by default. The time within this interval that the Layer 3 Switch selects is random for each message and is not affected by traffic loads or other network factors. The random interval minimizes the probability that a host will receive Router Advertisement messages from other routers at the same time. The interval on each IRDP-enabled Layer 3 Switch interface is independent of the interval on other IRDP-enabled interfaces. The default maximum message interval is 600 seconds. The default minimum message interval is 450 seconds.
- **Hold time** – Each Router Advertisement message contains a hold time value. This value specifies the maximum amount of time the host should consider an advertisement to be valid until a newer advertisement arrives. When a new advertisement arrives, the hold time is reset. The hold time is always longer than the maximum advertisement interval. Therefore, if the hold time for an advertisement expires, the host can reasonably conclude that the router interface that sent the advertisement is no longer available. The default hold time is three times the maximum message interval.
- **Preference** – If a host receives multiple Router Advertisement messages from different routers, the host selects the router that sent the message with the highest preference as the default gateway. The preference can be a number from 0-4294967296 to 0-4294967295. The default is 0.

Enabling IRDP globally

To globally enable IRDP, enter the following command.

```
Brocade(config)# ip irdp
```

This command enables IRDP on the IP interfaces on all ports. Each port uses the default values for the IRDP parameters. The parameters are not configurable when IRDP is globally enabled.

Enabling IRDP on an individual port

To enable IRDP on an individual interface and change IRDP parameters, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3
Brocade(config-if-1/3)# ip irdp maxadvertinterval 400
```

This example shows how to enable IRDP on a specific port and change the maximum advertisement interval for Router Advertisement messages to 400 seconds.

NOTE

To enable IRDP on individual ports, you must leave the feature globally disabled.

Syntax: [no] ip irdp [broadcast | multicast] [holdtime <seconds>] [maxadvertinterval <seconds>] [minadvertinterval <seconds>] [preference <number>]

The **broadcast** | **multicast** parameter specifies the packet type the Layer 3 Switch uses to send Router Advertisement:

- **broadcast** – The Layer 3 Switch sends Router Advertisement as IP broadcasts. This is the default.
- **multicast** – The Layer 3 Switch sends Router Advertisement as multicast packets addressed to IP multicast group 224.0.0.1.

The **holdtime** <seconds> parameter specifies how long a host that receives a Router Advertisement from the Layer 3 Switch should consider the advertisement to be valid. When a host receives a new Router Advertisement message from the Layer 3 Switch, the host resets the hold time for the Layer 3 Switch to the hold time specified in the new advertisement. If the hold time of an advertisement expires, the host discards the advertisement, concluding that the router interface that sent the advertisement is no longer available. The value must be greater than the value of the **maxadvertinterval** parameter and cannot be greater than 9000. The default is three times the value of the **maxadvertinterval** parameter.

The **maxadvertinterval** parameter specifies the maximum amount of time the Layer 3 Switch waits between sending Router Advertisements. You can specify a value from 1 to the current value of the **holdtime** parameter. The default is 600 seconds.

The **minadvertinterval** parameter specifies the minimum amount of time the Layer 3 Switch can wait between sending Router Advertisements. The default is three-fourths (0.75) the value of the **maxadvertinterval** parameter. If you change the **maxadvertinterval** parameter, the software automatically adjusts the **minadvertinterval** parameter to be three-fourths the new value of the **maxadvertinterval** parameter. If you want to override the automatically configured value, you can specify an interval from 1 to the current value of the **maxadvertinterval** parameter.

The **preference** <number> parameter specifies the IRDP preference level of this Layer 3 Switch. If a host receives Router Advertisements from multiple routers, the host selects the router interface that sent the message with the highest interval as the host default gateway. The valid range is 0-4294967296 to 0-4294967295. The default is 0.

Reverse Address Resolution Protocol configuration

The Reverse Address Resolution Protocol (RARP) provides a simple mechanism for directly-attached IP hosts to boot over the network. RARP allows an IP host that does not have a means of storing its IP address across power cycles or software reloads to query a directly-attached router for an IP address.

RARP is enabled by default. However, you must create a RARP entry for each host that will use the Layer 3 Switch for booting. A RARP entry consists of the following information:

- The entry number – the entry sequence number in the RARP table.
- The MAC address of the boot client.
- The IP address you want the Layer 3 Switch to give to the client.

When a client sends a RARP broadcast requesting an IP address, the Layer 3 Switch responds to the request by looking in the RARP table for an entry that contains the client MAC address:

- If the RARP table contains an entry for the client, the Layer 3 Switch sends a unicast response to the client that contains the IP address associated with the client MAC address in the RARP table.
- If the RARP table does not contain an entry for the client, the Layer 3 Switch silently discards the RARP request and does not reply to the client.

How RARP Differs from BootP and DHCP

RARP and BootP/DHCP are different methods for providing IP addresses to IP hosts when they boot. These methods differ in the following ways:

- Location of configured host addresses:
 - RARP requires static configuration of the host IP addresses on the Layer 3 Switch. The Layer 3 Switch replies directly to a host request by sending an IP address you have configured in the RARP table.
 - The Layer 3 Switch forwards BootP and DHCP requests to a third-party BootP/DHCP server that contains the IP addresses and other host configuration information.
- Connection of host to boot source (Layer 3 Switch or BootP/DHCP server):
 - RARP requires the IP host to be directly attached to the Layer 3 Switch.
 - An IP host and the BootP/DHCP server can be on different networks and on different routers, so long as the routers are configured to forward (“help”) the host boot request to the boot server.
 - You can centrally configure other host parameters on the BootP/DHCP server, in addition to the IP address, and supply those parameters to the host along with its IP address.

To configure the Layer 3 Switch to forward BootP/DHCP requests when boot clients and the boot servers are on different subnets on different Layer 3 Switch interfaces, refer to [“BootP and DHCP relay parameter configuration”](#) on page 1011.

Disabling RARP

RARP is enabled by default. To disable RARP, enter the following command at the global CONFIG level.

```
Brocade(config)# no ip rarp
```

Syntax: [no] ip rarp

To re-enable RARP, enter the following command.

```
Brocade(config)# ip rarp
```

Creating static RARP entries

You must configure the RARP entries for the RARP table. The Layer 3 Switch can send an IP address in reply to a client RARP request only if create a RARP entry for that client.

To assign a static IP RARP entry for static routes on a Brocade router, enter a command such as the following.

```
Brocade(config)# rarp 1 1245.7654.2348 192.53.4.2
```

This command creates a RARP entry for a client with MAC address 1245.7654.2348. When the Layer 3 Switch receives a RARP request from this client, the Layer 3 Switch replies to the request by sending IP address 192.53.4.2 to the client.

Syntax: rarp <number> <mac-addr>.<ip-addr>

The <number> parameter identifies the RARP entry number. You can specify an unused number from 1 to the maximum number of RARP entries supported on the device. To determine the maximum number of entries supported on the device, refer to the section [“Displaying and modifying system parameter default settings”](#) on page 578.

The <mac-addr> parameter specifies the MAC address of the RARP client.

The <ip-addr> parameter specifies the IP address the Layer 3 Switch will give the client in response to the client RARP request.

Changing the maximum number of static RARP entries supported

The number of RARP entries the Layer 3 Switch supports depends on how much memory the Layer 3 Switch has. To determine how many RARP entries your Layer 3 Switch can have, display the system default information using the procedure in the section [“Displaying system parameter default values”](#) on page 578.

If your Layer 3 Switch allows you to increase the maximum number of RARP entries, you can use a procedure in the same section to do so.

NOTE

You must save the configuration to the startup-config file and reload the software after changing the RARP cache size to place the change into effect.

Configuring UDP broadcast and IP helper parameters

Some applications rely on client requests sent as limited IP broadcasts addressed to the UDP application port. If a server for the application receives such a broadcast, the server can reply to the client. Routers do not forward subnet directed broadcasts, so the client and server must be on the same network for the broadcast to reach the server. If the client and server are on different networks (on opposite sides of a router), the client request cannot reach the server.

You can configure the Layer 3 Switch to forward clients' requests to UDP application servers. To do so:

- Enable forwarding support for the UDP application port, if forwarding support is not already enabled.
- Configure a helper address on the interface connected to the clients. Specify the helper address to be the IP address of the application server or the subnet directed broadcast address for the IP subnet the server is in. A helper address is associated with a specific interface and applies only to client requests received on that interface. The Layer 3 Switch forwards client requests for any of the application ports the Layer 3 Switch is enabled to forward to the helper address.

Forwarding support for the following application ports is enabled by default:

- bootps (port 67)
- dns (port 53)
- tftp (port 69)
- time (port 37)
- netbios-ns (port 137)
- netbios-dgm (port 138)
- tacacs (port 65)

NOTE

The application names are the names for these applications that the Layer 3 Switch software recognizes, and might not match the names for these applications on some third-party devices. The numbers listed in parentheses are the UDP port numbers for the applications. The numbers come from RFC 1340.

NOTE

Forwarding support for BootP/DHCP is enabled by default. If you are configuring the Layer 3 Switch to forward BootP/DHCP requests, refer to [“BootP and DHCP relay parameter configuration”](#) on page 1011.

You can enable forwarding for other applications by specifying the application port number.

You also can disable forwarding for an application.

NOTE

If you disable forwarding for a UDP application, forwarding of client requests received as broadcasts to helper addresses is disabled. Disabling forwarding of an application does not disable other support for the application. For example, if you disable forwarding of Telnet requests to helper addresses, other Telnet support on the Layer 3 Switch is not also disabled.

Enabling forwarding for a UDP application

If you want the Layer 3 Switch to forward client requests for UDP applications that the Layer 3 Switch does not forward by default, you can enable forwarding support for the port. To enable forwarding support for a UDP application, use the following method. You also can disable forwarding for an application using this method.

NOTE

You also must configure a helper address on the interface that is connected to the clients for the application. The Layer 3 Switch cannot forward the requests unless you configure the helper address. Refer to [“Configuring an IP helper address”](#) on page 1012.

To enable the forwarding of NTP broadcasts, enter the following command.

```
Brocade(config)# ip forward-protocol udp ntp
```

Syntax: [no] ip forward-protocol udp <udp-port-name> | <udp-port-num>

The <udp-port-name> parameter can have one of the following values. For reference, the corresponding port numbers from RFC 1340 are shown in parentheses. If you specify an application name, enter the name only, not the parentheses or the port number shown here:

- bootpc (port 68)
- bootps (port 67)
- discard (port 9)
- dns (port 53)
- dnsix (port 90)
- echo (port 7)
- mobile-ip (port 434)
- netbios-dgm (port 138)
- netbios-ns (port 137)
- ntp (port 123)
- tacacs (port 65)
- talk (port 517)
- time (port 37)
- tftp (port 69)

In addition, you can specify any UDP application by using the application UDP port number.

The <udp-port-num> parameter specifies the UDP application port number. If the application you want to enable is not listed above, enter the application port number. You also can list the port number for any of the applications listed above.

To disable forwarding for an application, enter a command such as the following.

```
Brocade(config)# no ip forward-protocol udp ntp
```

This command disables forwarding of SNMP requests to the helper addresses configured on Layer 3 Switch interfaces.

Configuring an IP helper address

To forward a client broadcast request for a UDP application when the client and server are on different networks, you must configure a helper address on the interface connected to the client. Specify the server IP address or the subnet directed broadcast address of the IP subnet the server is in as the helper address.

You can configure up to 16 helper addresses on each interface. You can configure a helper address on an Ethernet port or a virtual interface.

To configure a helper address on interface 2 on chassis module 1, enter the following commands.

```
Brocade(config)# interface ethernet 1/2
Brocade(config-if-1/2)# ip helper-address 1 207.95.7.6
```

The commands in this example change the CLI to the configuration level for port 1/2, then add a helper address for server 207.95.7.6 to the port. If the port receives a client request for any of the applications that the Layer 3 Switch is enabled to forward, the Layer 3 Switch forwards the client request to the server.

Syntax: `ip helper-address <num> <ip-addr>`

The `<num>` parameter specifies the helper address number and can be from 1 through 16.

The `<ip-addr>` command specifies the server IP address or the subnet directed broadcast address of the IP subnet the server is in.

BootP and DHCP relay parameter configuration

A host on an IP network can use BootP or DHCP to obtain its IP address from a BootP/DHCP server. To obtain the address, the client sends a BootP or DHCP request. The request is a subnet directed broadcast and is addressed to UDP port 67. A limited IP broadcast is addressed to IP address 255.255.255.255 and is not forwarded by the Brocade Layer 3 Switch or other IP routers.

When the BootP or DHCP client and server are on the same network, the server receives the broadcast request and replies to the client. However, when the client and server are on different networks, the server does not receive the client request, because the Layer 3 Switch does not forward the request.

You can configure the Layer 3 Switch to forward BootP/DHCP requests. To do so, configure a helper address on the interface that receives the client requests, and specify the BootP/DHCP server IP address as the address you are helping the BootP/DHCP requests to reach. Instead of the server IP address, you can specify the subnet directed broadcast address of the IP subnet the server is in.

BootP and DHCP relay parameters

The following parameters control the Layer 3 Switch forwarding of BootP and DHCP requests:

- **Helper address** – The BootP/DHCP server IP address. You must configure the helper address on the interface that receives the BootP/DHCP requests from the client. The Layer 3 Switch cannot forward a request to the server unless you configure a helper address for the server.

- **Gateway address** – The Layer 3 Switch places the IP address of the interface that received the BootP/DHCP request in the request packet Gateway Address field (sometimes called the Router ID field). When the server responds to the request, the server sends the response as a unicast packet to the IP address in the Gateway Address field. (If the client and server are directly attached, the Gateway ID field is empty and the server replies to the client using a unicast or broadcast packet, depending on the server.)

By default, the Layer 3 Switch uses the lowest-numbered IP address on the interface that receives the request as the Gateway address. You can override the default by specifying the IP address you want the Layer 3 Switch to use.

- **Hop count** – Each router that forwards a BootP/DHCP packet increments the hop count by 1. Routers also discard a forwarded BootP/DHCP request instead of forwarding the request if the hop count is greater than the maximum number of BootP/DHCP hops allowed by the router. By default, a Brocade Layer 3 Switch forwards a BootP/DHCP request if its hop count is four or less, but discards the request if the hop count is greater than four. You can change the maximum number of hops the Layer 3 Switch will allow to a value from 1 through 15.

NOTE

The BootP/DHCP hop count is not the TTL parameter.

Configuring an IP helper address

The procedure for configuring a helper address for BootP/DHCP requests is the same as the procedure for configuring a helper address for other types of UDP broadcasts. Refer to [“Configuring an IP helper address”](#) on page 1011.

Configuring the BOOTP and DHCP reply source address

NOTE

This feature is supported on FastIron X Series and Brocade FCX Series devices. This feature is **not** supported on FastIron WS devices.

You can configure the Brocade device so that a BOOTP/DHCP reply to a client contains the server IP address as the source address instead of the router IP address. To do so, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)# ip helper-use-responder-ip
```

Syntax: [no] ip helper-use-responder-ip

Changing the IP address used for stamping BootP and DHCP requests

When the Layer 3 Switch forwards a BootP/DHCP request, the Layer 3 Switch “stamps” the Gateway Address field. The default value the Layer 3 Switch uses to stamp the packet is the lowest-numbered IP address configured on the interface that received the request. If you want the Layer 3 Switch to use a different IP address to stamp requests received on the interface, use either of the following methods to specify the address.

The BootP/DHCP stamp address is an interface parameter. Change the parameter on the interface that is connected to the BootP/DHCP client.

To change the IP address used for stamping BootP/DHCP requests received on interface 1/1, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1
Brocade(config-if-1/1)# ip bootp-gateway 109.157.22.26
```

These commands change the CLI to the configuration level for port 1/1, then change the BootP/DHCP stamp address for requests received on port 1/1 to 192.157.22.26. The Layer 3 Switch will place this IP address in the Gateway Address field of BootP/DHCP requests that the Layer 3 Switch receives on port 1/1 and forwards to the BootP/DHCP server.

Syntax: `ip bootp-gateway <ip-addr>`

Changing the maximum number of hops to a BootP relay server

Each BootP or DHCP request includes a field Hop Count field. The Hop Count field indicates how many routers the request has passed through. When the Layer 3 Switch receives a BootP/DHCP request, the Layer 3 Switch looks at the value in the Hop Count field:

- If the hop count value is equal to or less than the maximum hop count the Layer 3 Switch allows, the Layer 3 Switch increments the hop count by one and forwards the request.
- If the hop count is greater than the maximum hop count the Layer 3 Switch allows, the Layer 3 Switch discards the request.

To change the maximum number of hops the Layer 3 Switch allows for forwarded BootP/DHCP requests, use either of the following methods.

NOTE

The BootP and DHCP hop count is not the TTL parameter.

To modify the maximum number of BootP/DHCP hops, enter the following command.

```
Brocade(config)#bootp-relay-max-hops 10
```

This command allows the Layer 3 Switch to forward BootP/DHCP requests that have passed through ten previous hops before reaching the Layer 3 Switch. Requests that have traversed 11 hops before reaching the switch are dropped. Since the hop count value initializes at zero, the hop count value of an ingressing DHCP Request packet is the number of Layer 3 routers that the packet has already traversed.

Syntax: `bootp-relay-max-hops <1 through 15>`

DHCP Server

All FastIron devices can be configured to function as DHCP Servers.

NOTE

The DHCP server is platform independent and has no differences in behavior or configuration across all FastIron platforms (FESX, FSX,FWS, FCX, and ICX).

Dynamic Host Configuration Protocol (DHCP) is a computer networking protocol used by devices (DHCP clients) to obtain leased (or permanent) IP addresses. DHCP is an extension of the Bootstrap Protocol (BOOTP). The differences between DHCP and BOOTP are the address allocation and renewal process.

DHCP introduces the concept of a lease on an IP address. Refer to [“How DHCP Client-Based Auto-Configuration and Flash image update works”](#) on page 1030. The DHCP server can allocate an IP address for a specified amount of time, or can extend a lease for an indefinite amount of time. DHCP provides greater control of address distribution within a subnet. This feature is crucial if the subnet has more devices than available IP address. In contrast to BOOTP, which has two types of messages that can be used for leased negotiation, DHCP provides 7 types of messages. Refer to [“Supported Options for DHCP Servers”](#) on page 1033.

DHCP allocates temporary or permanent network IP addresses to clients. When a client requests the use of an address for a time interval, the DHCP server guarantees not to reallocate that address within the requested time and tries to return the same network address each time the client makes a request. The period of time for which a network address is allocated to a client is called a lease. The client may extend the lease through subsequent requests. When the client is done with the address, they can release the address back to the server. By asking for an indefinite lease, clients may receive a permanent assignment.

In some environments, it may be necessary to reassign network addresses due to exhaustion of the available address pool. In this case, the allocation mechanism reuses addresses with expired leases.

Configuration Notes for configuring DHCP servers

- DHCP server is supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images.
- In the event of a controlled or forced switchover, a DHCP client will request from the DHCP server the same IP address and lease assignment that it had before the switchover. After the switchover, the DHCP Server feature will be automatically re-initialized on the new active controller or management module.
- For DHCP client hitless support in an IronStack, the **stack mac** command must be used to configure the IronStack MAC address, so that the MAC address does not change in the event of a switchover or failover. If **stack mac** is not configured, the MAC address/IP address pair assigned to a DHCP client will not match after a switchover or failover. Furthermore, in the Layer 3 router image, if the **stack mac** configuration is changed or removed and the management port has a dynamic IP address, when a DHCP client tries to renew its lease from the DHCP server, the DHCP server will assign a different IP address.
- If any address from the configured DHCP pool is used, for example by the DHCP server, TFTP server, etc., you must exclude the address from the network pool. For configuration instructions, refer to [“Specifying addresses to exclude from the address pool”](#) on page 1023.

DHCP Option 82 support

The DHCP relay agent information option (DHCP option 82) enables a DHCP relay agent to include information about itself when forwarding client-originated DHCP packets to a DHCP server. The DHCP server uses this information to implement IP address or other parameter-assignment policies.

In a metropolitan Ethernet-access environment, the DHCP server can centrally manage IP address assignments for a large number of subscribers. If DHCP option 82 is disabled, a DHCP policy can only be applied per subnet, rather than per physical port. When DHCP option 82 is enabled, a subscriber is identified by the physical port through which it connects to the network.

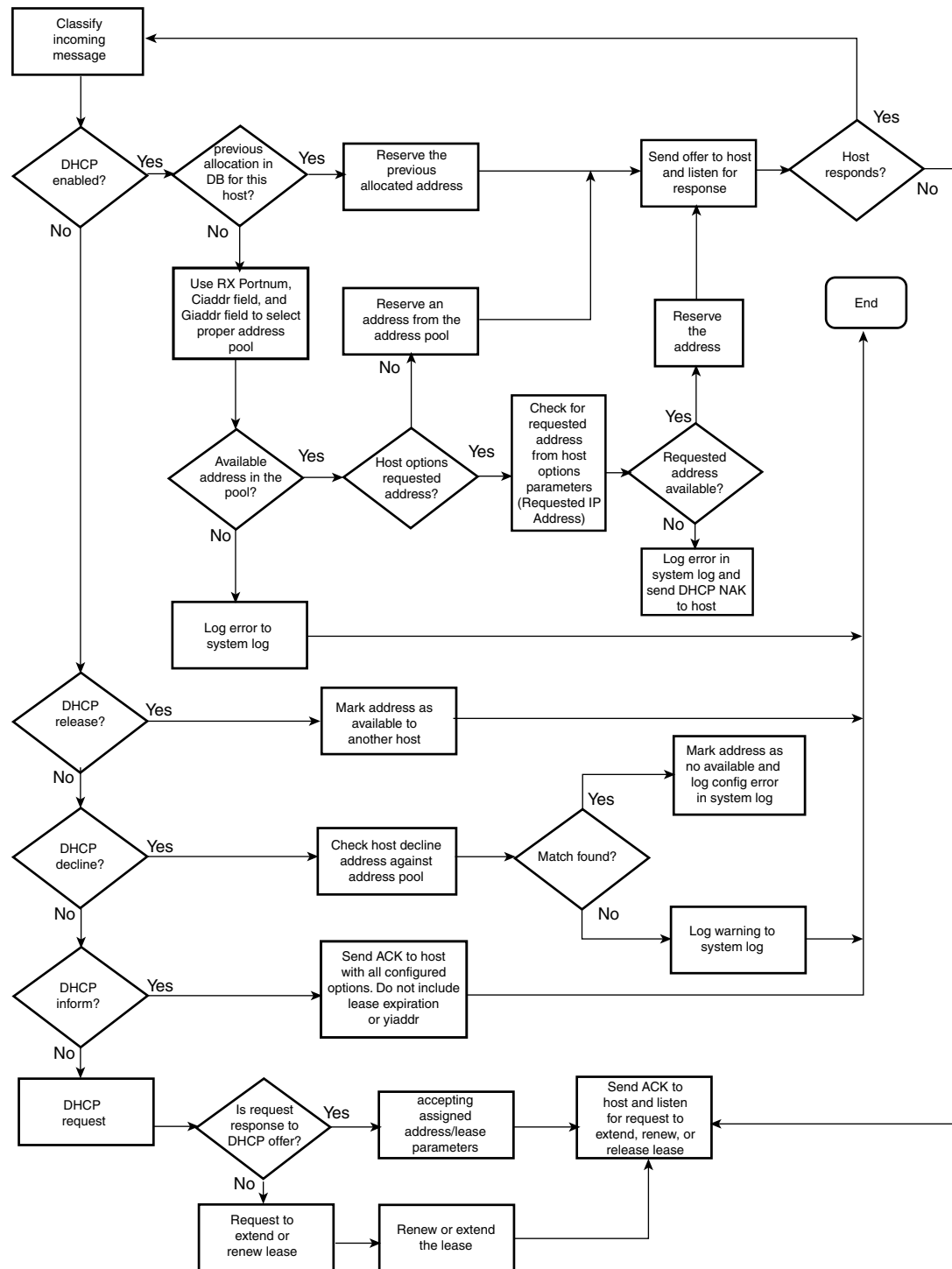
DHCP Server options

A FastIron configured as a DHCP server can support up to 1000 DHCP clients, offering them the following options:

- **NetBIOS over TCP/IP Name Server** - Specifies a list of RFC1001/1002 NBNS name servers listed in order of preference.
- **Domain Name Server** - Specifies a list of Domain Name System (RFC 1035) name servers available to the client. Servers are listed in order of preference.
- **Domain Name** - Specifies the domain name the client should use when resolving hostnames using the Domain Name system.
- **Router Option** - specifies a list of IP addresses for routers on the client subnet. Routers are listed in order of preference.
- **Subnet Mask** - Specifies the client subnet mask (per RFC950).
- **Vendor Specific Information** - Allows clients and servers to exchange vendor-specific information.
- **Boot File** - Specifies a boot image to be used by the client
- **Next Bootstrap Server** - Configures the IP address of the next server to be used for startup by the client.
- **TFTP Server** - Configures the address or name of the TFTP server available to the client.

A DHCP server assigns and manages IPv4 addresses from multiple address pools, using dynamic address allocation. The DHCP server also contains the relay agent to forward DHCP broadcast messages to network segments that do not support these types of messages.

FIGURE 116 DHCP Server configuration flow chart



Configuring DHCP Server on a device

Perform the following steps to configure the DHCP Server feature on your FastIron device:

1. Enable DHCP Server by entering a command similar to the following.

```
Brocade(config)# ip dhcp-server enable
```

2. Create a DHCP Server address pool by entering a command similar to the following.

```
Brocade(config)# ip dhcp-server pool cabo
```

3. Configure the DHCP Server address pool by entering commands similar to the following.

```
Brocade(config-dhcp-cabo)# network 172.16.1.0/24
Brocade(config-dhcp-cabo)# domain-name brocade.com
Brocade(config-dhcp-cabo)# dns-server 172.16.1.2 172.16.1.3
Brocade(config-dhcp-cabo)# netbios-name-server 172.16.1.2
Brocade(config-dhcp-cabo)# lease 0 0 5
```

4. To disable DHCP, enter a command similar to the following.

```
Brocade(config)# no ip dhcp-server enable
```

The following sections describe the default DHCP settings, CLI commands and the options you can configure for the DHCP Server feature.

Default DHCP server settings

Table 171 shows the default DHCP server settings.

TABLE 171 DHCP server default settings

Parameter	Default Value
DHCP server	Disabled
Lease database expiration time	86400 seconds
The duration of the lease for an assigned IP address	43200 seconds (one day)
Maximum lease database expiration time	86400 seconds
DHCP server with option 82	Disabled
DHCP server unknown circuit-ID for Option 82	Permit range lookup
IP distribution mechanism	Linear

DHCP server CLI commands**TABLE 172** DHCP server optional parameters commands

Command	Description
<code>dbexpire</code>	Specifies how long, in seconds, the DHCP server should wait before aborting a database transfer
<code>option domain-name</code>	Specifies the domain name for the DHCP clients.
<code>option domain-nameservers</code>	Specifies the Domain Name System (DNS) IP servers that are available to the DHCP clients.
<code>option merit-dump</code>	Specifies the path name of a file into which the client's core image should be placed in the event that the client crashes (the DHCP application issues an exception in case of errors such as division by zero).
<code>option root-path</code>	Specifies the name of the path that contains the client's root filesystem in NFS notation.
<code>option router</code>	Adds the default router and gateway for the DHCP clients.
<code>option subnet-mask</code>	Defines the subnet mask for the network.
<code>option broadcastaddress</code>	Defines a broadcast address for the network.
<code>option wins-server</code>	Defines the NetBIOS Windows Internet Naming Service (WINS) name servers that are available to Microsoft DHCP clients.
<code>option log-servers</code>	Defines a list of log servers available to the client.
<code>option bootstrapserver</code>	Specifies the IP address of the bootstrap server (the command fills the "siaddr" field in the DHCP packet).

This section describes the CLI commands that are available in the DHCP Server feature.

TABLE 173 DHCP Server CLI commands

Command	Description
ip dhcp-server arp-ping-timeout <#>	Specifies the time (in seconds) the server will wait for a response to an arp-ping packet before deleting the client from the binding database. The minimum setting is 5 seconds and the maximum time is 30 seconds. NOTE: Do not alter the default value unless it is necessary. Increasing the value of this timer may increase the time to get console access after a reboot.
clear ip dhcp-server binding	Deletes a specific, or all leases from the binding database. Refer to “Removing DHCP leases” on page 1020.
ip dhcp-server enable	Enables the DHCP server feature. Refer to “Enabling DHCP Server” on page 1020.
no ip dhcp-server mgmt	Disables DHCP server on the management port. Refer to “Disabling DHCP Server on the management port” on page 1020.
ip dhcp-server pool <name>	Switches to pool configuration mode (config-dhcp-name# prompt) and creates an address pool. Refer to “Creating an address pool” on page 1021.
ip dhcp-server relay-agent-echo enable	Enables relay agent echo (Option 82). Refer to “Enabling relay agent echo (Option 82)” on page 1021.
ip dhcp-server <server-id>	Specifies the IP address of the selected DHCP server. Refer to “Configuring the IP address of the DHCP server” on page 1021.
show ip dhcp-server binding [<address>]	Displays a specific lease entry, or all lease entries. Refer to “Displaying active lease entries” on page 1024.
show ip dhcp-server address-pool <name>	Displays a specific address pool or all address pools. Refer to “Displaying address-pool information” on page 1025.
show ip dhcp-server flash	Displays the lease binding database that is stored in flash memory. Refer to “Displaying lease-binding information in flash memory” on page 1026.
show ip dhcp-server summary	Displays a summary of active leases, deployed address pools, undeployed address pools, and server uptime. “Displaying summary DHCP server information” on page 1027.
bootfile <name>	Specifies a boot image to be used by the client. Refer to “Configuring the boot image” on page 1022.
deploy	Deploys an address pool configuration to the server. Refer to “Deploying an address pool configuration to the server” on page 1022.
dhcp-default-router <addresses>	Specifies the IP address of the default router or routers for a client. Refer to “Specifying default routers available to the client” on page 1022.
dns-server <addresses>	Specifies the IP addresses of a DNS server or servers available to the client. Refer to “Specifying DNS servers available to the client” on page 1022.
domain-name <domain>	Configures the domain name for the client. Refer to “Configuring the domain name for the client” on page 1022.
lease <days><hours><minutes>	Specifies the lease duration for an address pool. The default is a one-day lease. Refer to “Configuring the lease duration for the address pool” on page 1022.
excluded-address [<address> <address-low> <address-high>]	Specifies an address or range of addresses to be excluded from the address pool. Refer to “Specifying addresses to exclude from the address pool” on page 1023.

TABLE 173 DHCP Server CLI commands (Continued)

Command	Description
netbios-name-server <address> [<address2> <address3>]	Specifies the IP address of a NetBIOS WINS server or servers that are available to Microsoft DHCP clients. Refer to “Configuring the NetBIOS server for DHCP clients” on page 1023.
network <subnet>/<mask>	Configures the subnet network and mask of the DHCP address pool. Refer to “Configuring the subnet and mask of a DHCP address pool” on page 1023.
next-bootstrap-server <address>	Configures the IP address of the next server to be used for startup by the client. Refer to “Configuring a next-bootstrap server” on page 1023.
tftp-server <address> name <name>	Configures the address or name of the TFTP server available to the client. Refer to “Configuring the TFTP server” on page 1024.
vendor-class [<ascii> <ip> <hex> > <value>]	Specifies the vendor type and configuration value for the DHCP client. Refer to “Configuring a vendor type and configuration value for a DHCP client” on page 1024.

Removing DHCP leases

The **clear ip dhcp-server binding** command can be used to delete a specific lease, or all lease entries from the lease binding database.

```
Brocade(config)# clear ip dhcp-server binding *
```

Syntax: **clear ip dhcp-server binding** [<address> | <*>]

- <address> - The IP address to be deleted
- <*> - Clears all IP addresses

Enabling DHCP Server

The **ip dhcp-server enable** command enables DHCP Server, which is disabled by default.

Syntax: **[no] ip dhcp-server enable**

The **no** version of this command disables DHCP Server.

Disabling DHCP Server on the management port

By default, when DHCP Server is enabled, it responds to DHCP client requests received on the management port. If desired, you can prevent the response to DHCP client requests received on the management port, by disabling DHCP Server support on the port. When disabled, DHCP client requests that are received on the management port are silently discarded.

To disable DHCP Server on the management port, enter the following command at the global configuration level of the CLI.

```
Brocade(config)# no ip dhcp-server mgmt
```

To re-enable DHCP Server on the management port after it has been disabled, enter the **ip dhcp-server mgmt** command:

```
Brocade(config)# ip dhcp-server mgmt
```

Syntax: **[no] ip dhcp-server mgmt**

Setting the wait time for ARP-ping response

At startup, the server reconciles the lease-binding database by sending an ARP-ping packet out to every client. If there is no response to the ARP-ping packet within a set amount of time (set in seconds), the server deletes the client from the lease-binding database. The minimum setting is 5 seconds and the maximum is 30 seconds.

Syntax: `ip dhcp-server arp-ping-timeout <num>`

- `<num>` - The number of seconds to wait for a response to an ARP-ping packet.

NOTE

Do not alter the default value unless it is necessary. Increasing the value of this timer may increase the time to get console access after a reboot.

Creating an address pool

The `ip dhcp-server pool` command puts you in pool configuration mode, and allows you to create an address pool.

```
Brocade(config)# ip dhcp-server pool
Brocade(config-dhcp-name)# ip dhcp-server pool monterey
Brocade(config-dhcp-monterey)#
```

These commands create an address pool named monterey.

Syntax: `ip dhcp-server pool <name>`

Configuration notes for creating an address pool

- If the DHCP server address is part of a configured DHCP address pool, you must exclude the DHCP server address from the network pool. Refer to [“Specifying addresses to exclude from the address pool”](#) on page 1023.
- While in DHCP server pool configuration mode, the system will place the DHCP server pool in *pending* mode and the DHCP server will not use the address pool to distribute information to clients. To activate the pool, use the **deploy** command. Refer to [“Deploying an address pool configuration to the server”](#) on page 1022.

Enabling relay agent echo (Option 82)

The `ip dhcp-server relay-agent-echo enable` command activates DHCP Option 82, and enables the DHCP server to echo relay agent information in all replies.

```
Brocade(config)# ip dhcp-server relay-agent-echo enable
```

Syntax: `ip dhcp-server relay-agent-echo enable`

Configuring the IP address of the DHCP server

The `ip dhcp-server` command specifies the IP address of the selected DHCP server, as shown in this example:

```
Brocade(config)# ip dhcp-server 102.1.1.144
```

Syntax: `ip dhcp-server <server-identifier>`

- *<server-identifier>* - The IP address of the DHCP server

This command assigns an IP address to the selected DHCP server.

Configuring the boot image

The **bootfile** command specifies a boot image name to be used by the DHCP client.

```
Brocade(config-dhcp-cabo)# bootfile foxhound
```

In this example, the DHCP client should use the boot image called “foxhound”.

Syntax: **bootfile** *<name>*

Deploying an address pool configuration to the server

The **deploy** command sends an address pool configuration to the DHCP server.

```
Brocade(config-dhcp-cabo)# deploy
```

Syntax: **deploy**

Specifying default routers available to the client

The **dhcp-default-router** command specifies the ip addresses of the default routers for a client.

Syntax: **dhcp-default-router** *<address>* [*<address>*, *<address>*]

Specifying DNS servers available to the client

The **dns-server** command specifies DNS servers that are available to DHCP clients.

```
Brocade(config-dhcp-cabo)# dns-server 102.2.1.143, 101.2.2.142
```

Syntax: **dns-server** *<address>* [*<address>*, *<address>*]

Configuring the domain name for the client

The **domain-name** command configures the domain name for the client.

```
Brocade(config-dhcp-cabo)# domain-name sierra
```

Syntax: **domain-name** *<domain>*

Configuring the lease duration for the address pool

The **lease** command specifies the lease duration for the address pool. The default is a one-day lease.

```
Brocade(config-dhcp-cabo)# lease 1 4 32
```

In this example, the lease duration has been set to one day, four hours, and 32 minutes. You can set a lease duration for just days, just hours, or just minutes, or any combination of the three.

Syntax: **lease** *<days>* *<hours>* *<minutes>*

Specifying addresses to exclude from the address pool

The **excluded-address** command specifies either a single address, or a range of addresses that are to be excluded from the address pool.

```
Brocade(config-dhcp-cabo)# excluded-address 101.2.3.44
```

Syntax: **excluded-address** [<address> | <address-low address-high>]>

- <address> - Specifies a single address
- <address-low address-high> - Specifies a range of addresses

Configuring the NetBIOS server for DHCP clients

The **netbios-name-server** command specifies the IP address of a NetBIOS WINS server or servers that are available to Microsoft DHCP clients.

```
Brocade(config-dhcp-cabo)# netbios-name-server 192.168.1.55
```

Syntax: **netbios-name-server** <address> [<address2>, <address3>]

Configuring the subnet and mask of a DHCP address pool

This **network** command configures the subnet network and mask of the DHCP address pool.

```
Brocade(config-dhcp-cabo)# network 101.2.3.44/24
```

Syntax: **network** <subnet>/<mask>

Configuring a next-bootstrap server

The **next-bootstrap-server** command specifies the IP address of the next server the client should use for boot up.

```
Brocade(config-dhcp-cabo)# next-bootstrap-server 101.2.5.44
```

Syntax: **next-bootstrap-server** <address>

Configuring the TFTP server

The **tftp-server** command specifies the address or name of the TFTP server to be used by the DHCP clients.

To configure a TFTP server by specifying its IP address, enter a command similar to the following.

```
Brocade(config-dhcp-cabo)# tftp-server 101.7.5.48
```

To configure a TFTP server by specifying its server name, enter a command similar to the following.

```
Brocade(config-dhcp-cabo)# tftp-server tftp.domain.com
```

Syntax: **tftp-server** <address> | **name** <server-name>

- <address> is the IP address of the TFTP server.
- **name** configures the TFTP server specified by <server-name>.

If DHCP options 66 (TFTP server name) and 150 (TFTP server IP address) are both configured, the DHCP client ignores option 150 and tries to resolve the TFTP server name (option 66) using DNS.

Configuring a vendor type and configuration value for a DHCP client

The **vendor-class** command specifies the vendor-type and configuration value for a DHCP client.

```
Brocade(config-dhcp-cabo)# vendor class ascii waikiki
```

Syntax: **vendor-class** [<ascii> | <ip> | <hex>]> <value>

Displaying DHCP Server information

The following DHCP **show** commands can be entered from any level of the CLI.

Displaying active lease entries

The **show ip dhcp-server binding** command displays a specific active lease, or all active leases, as shown in the following example:

```
Brocade# show ip dhcp-server binding
```

The following output is displayed:

```
Brocade# show ip dhcp-server binding
```

Bindings from all pools:

IP Address	Client-ID/ Hardware address	Lease expiration	Type
192.168.1.2	001b.ed5d.a440	0d:0h:29m:31s	Automatic
192.168.1.3	0012.f2e1.26c0	0d:0h:29m:38s	Automatic

Syntax: **show ip dhcp-server binding** [<address>]

- <address> - Displays entries for this address only

The following table describes this output.

TABLE 174 CLI display of **show ip dhcp-server binding** command

Field	Description
IP address	The IP addresses currently in the binding database
Client ID/Hardware address	The hardware address for the client
Lease expiration	The time when this lease will expire
Type	The type of lease

Displaying address-pool information

This **show ip dhcp-server address-pool** command displays information about a specific address pool, or for all address pools.

```
Brocade# show ip dhcp-server address-pools
```

Output similar to the following is displayed, as shown here.

```
Showing all address pool(s):
```

```
Pool Name: one
Time elapsed since last save: 0d:0h:6m:52s
Total number of active leases: 2
Address Pool State: active
IP Address Exclusions: 192.168.1.45
IP Address Exclusions: 192.168.1.99 192.168.1.103
Pool Configured Options:
bootfile: example.bin
      dhcp-default-router: 192.168.1.1
      dns-server: 192.168.1.100
      domain-name: example.com
      lease: 0 0 30
      netbios-name-server: 192.168.1.101
      network: 192.168.1.0 255.255.255.0
      next-bootstrap-server: 192.168.1.102
      tftp-server: 192.168.1.103
```

Syntax: **show ip dhcp-server address-pool[s] [<name>]**

- **address-pool[s]** - If you enter address-pools, the display will show all address pools
- **<name>** - Displays information about a specific address pool

The following table describes this output.

TABLE 175 CLI display of **show ip dhcp-server address pools** command

Field	Description
Pool name	The name of the address pool
Time elapsed since last save	The time that has elapsed since the last save.
Total number of active leases	The number of leases that are currently active.
Address pool state	The state of the address pool (active or inactive).
IP Address exclusions	IP addresses that are not included in the address pool
Pool configured options	
bootfile	The name of the bootfile
dhcp-server-router	The address of the DHCP server router
dns-server	The address of the dns server
domain-name	The name of the domain
lease	The identifier for the lease
netbios-name server	The address of the netbios name server
network	The address of the network
next-bootstrap-server	The address of the next-bootstrap server
tftp-server	The address of the TFTP server

Displaying lease-binding information in flash memory

The **show ip dhcp-server flash** command displays the lease-binding database that is stored in flash memory.

```
Brocade# show ip dhcp-server flash
```

The following information is displayed.

```
Brocade# show ip dhcp-server flash
```

Address Pool Binding:

IP Address	Client-ID/ Hardware address	Lease expiration	Type
192.168.1.2	001b.ed5d.a440	0d:0h:18m:59s	Automatic
192.168.1.3	0012.f2e1.26c0	0d:0h:19m:8s	Automatic

Syntax: **show ip dhcp-server flash**

The following table describes this output.

TABLE 176 CLI display of **show ip dhcp-server flash** command

Field	Description
IP address	The IP address of the flash memory lease-binding database
Client-ID/Hardware address	The address of the client
Lease expiration	The time when the lease will expire
Type	The type of lease

Displaying summary DHCP server information

The **show ip dhcp-server summary** command displays information about active leases, deployed address-pools, undeployed address-pools, and server uptime.

```
Brocade# show ip dhcp-server summary
```

The following information is displayed.

DHCP Server Summary:

```

                Total number of active leases:  2
        Total number of deployed address-pools:  1
Total number of undeployed address-pools:  0
                Server uptime: 0d:0h:8m:27s
```

Syntax: **show ip dhcp-server summary**

The following table describes this output.

TABLE 177 CLI display of `show ip dhcp-server summary` command

Field	Description
Total number of active leases	Indicates the number of leases that are currently active
Total number of deployed address-pools	The number of address pools currently in use.
Total number of undeployed address-pools	The number of address-pools being held in reserve.
Server uptime	The amount of time that the server has been active.

TABLE 178 DHCP Server commands

Command	Description
<code>option bootstrapfilename</code>	Sets the name of the bootstrap file. The no form of this command removes the name of the bootstrap file.
<code>default-lease-time</code>	Specifies the duration of the lease for an IP address that is assigned from a DHCP server to a DHCP client.
<code>database tftp</code>	Defines the TFTP IP address server for storing the DHCP database, the name of the stored file and the time period at which the stored database is synchronized with the database on the device.
<code>database ftp</code>	Defines the FTP IP address server for storing the DHCP database, the name of the stored file and the time period at which the stored database is synchronized with the database on the device.
<code>max-lease-time</code>	Specifies the maximal duration of the leases in seconds.
<code>option bootfile-name</code>	Specifies the pathname of the boot file.
<code>option tftp-server</code>	Specifies the IP address of a TFTP server.

DHCP Client-Based Auto-Configuration and Flash image update

NOTE

The DHCP Client-Based Auto-Configuration and Flash image update are platform independent and have no differences in behavior or configuration across all platforms (FESX, FSX, FWS, and FCX).

DHCP Client-Based Auto-Configuration allows Layer 2 and Layer 3 devices to automatically obtain leased IP addresses through a DHCP server, negotiate address lease renewal, and obtain flash image and configuration files.

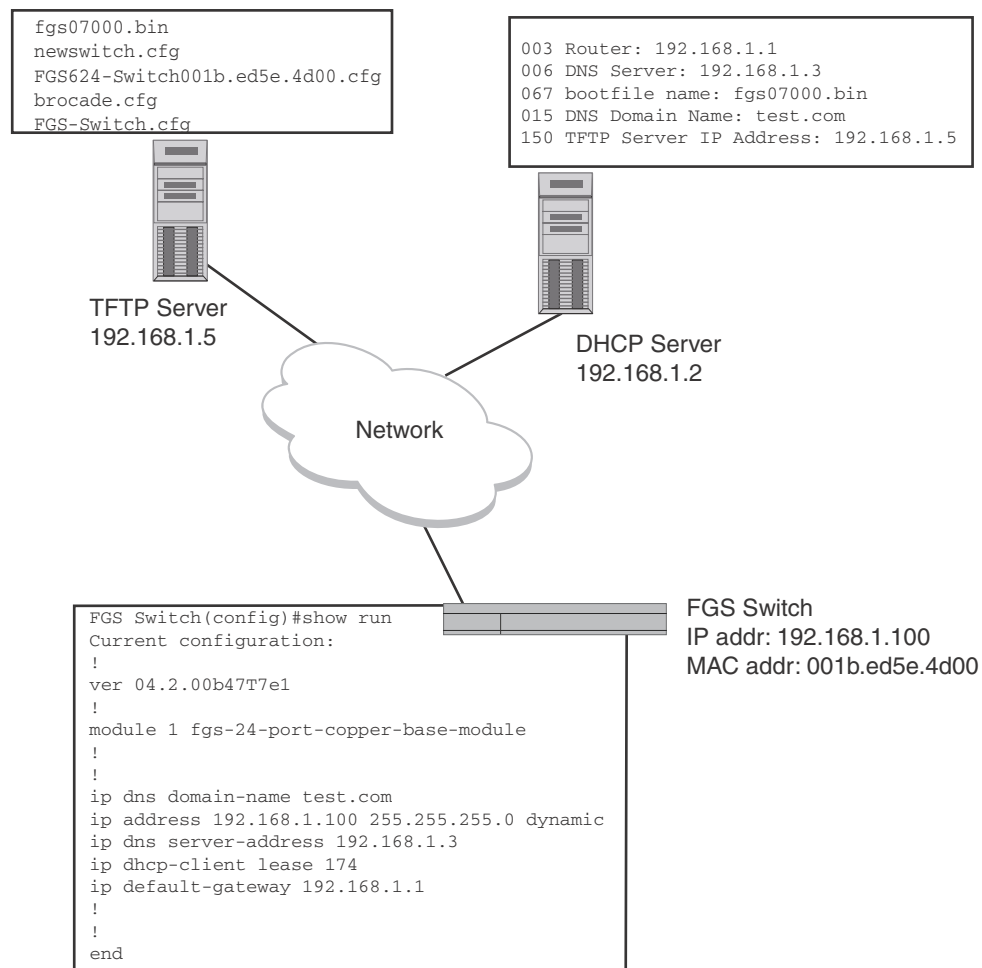
DHCP Client-Based Auto-Configuration occurs as follows.

1. The IP address validation and lease negotiation enables the DHCP client (a Brocade Layer 2 or Layer 3 device) to automatically obtain and configure an IP address, as follows:
 - One lease is granted for each Layer 2 device. If the device is configured with a static IP address, the DHCP Auto-Configuration feature is automatically disabled.
 - For a Layer 3 device, one leased address is granted (per device) to the interface that first receives a response from the DHCP server.

2. If **auto-update** is enabled, the TFTP flash image is downloaded and updated. The device compares the filename of the requested flash image with the image stored in flash. If the filenames are different, then the device will download the new image from a TFTP server, write the downloaded image to flash, then reload the device or stack.
3. In the final step, TFTP configuration download and update, the device downloads a configuration file from a TFTP server and saves it as the running configuration.

Figure 117 shows how DHCP Client-Based Auto Configuration works.

FIGURE 117 DHCP Client-Based Auto-Configuration



Configuration notes and feature limitations for DHCP client-based auto-configuration

- For Layer 3 devices, this feature is available for the default VLAN only. For Layer 2 devices, this feature is available for default VLANs and management VLANs. This feature is not supported on virtual interfaces (VEs), trunked ports, or LACP ports.
- Although the DHCP server may provide multiple addresses, only one IP address is installed at a time.

- This feature is not supported together with DHCP snooping.

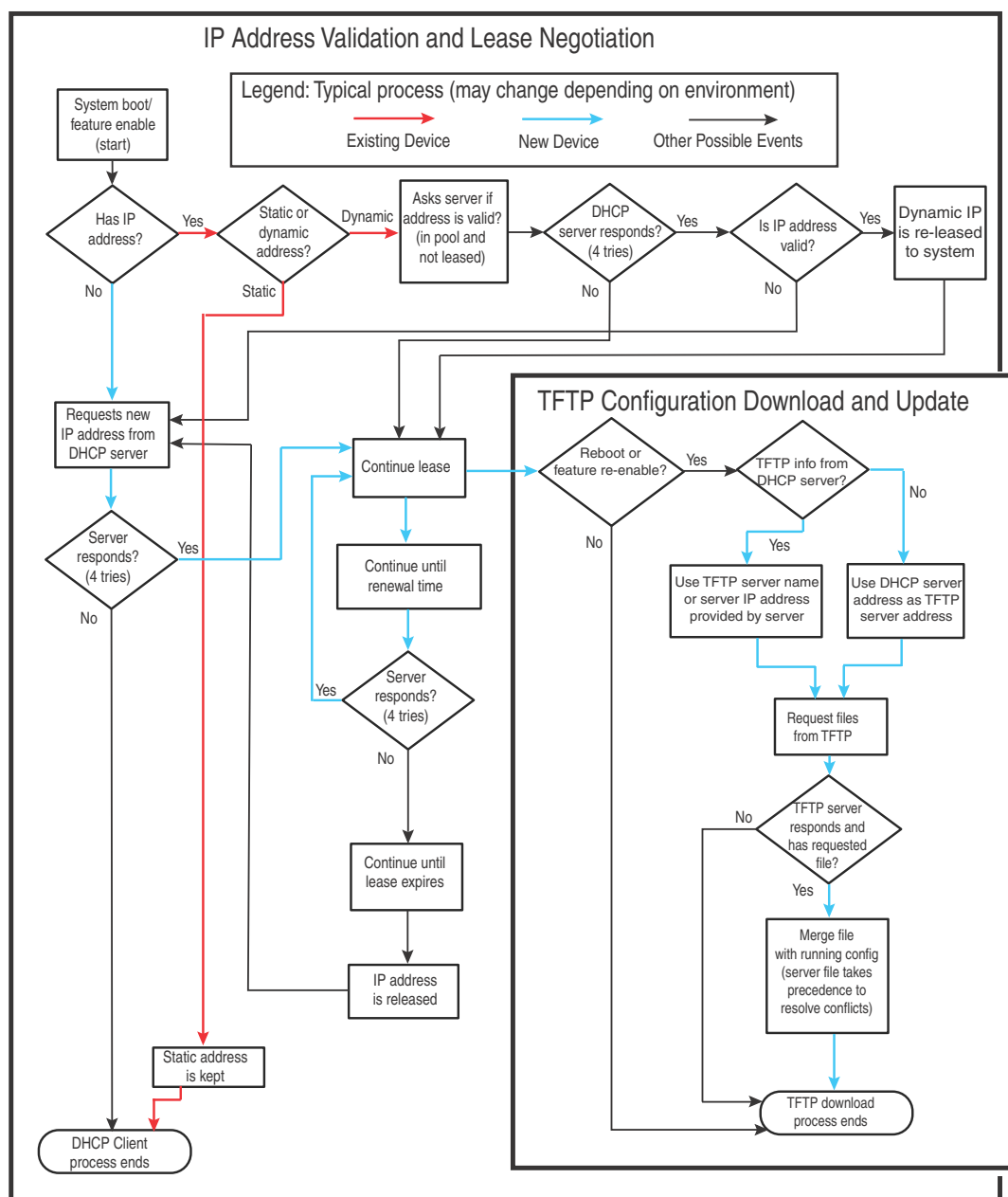
The following configuration rules apply to flash image update:

- To enable flash image update (**ip dhcp-client auto-update enable** command), also enable auto-configuration (**ip dhcp-client enable** command).
- The image filename to be updated must have the extension **.bin**.
- The DHCP option 067 bootfile name will be used for image update if it has the extension **.bin**.
- The DHCP option 067 bootfile name will be used for configuration download if it does not have the extension **.bin**.
- If the DHCP option 067 bootfile name is not configured or does not have the extension **.bin**, then the auto-update image will not occur.

How DHCP Client-Based Auto-Configuration and Flash image update works

Auto-Configuration and Auto-update are enabled by default. To disable this feature, refer to [“Disabling or re-enabling Auto-Configuration”](#) on page 1034 and [“Disabling or re-enabling Auto-Update”](#) on page 1034, respectively.

The steps of the Auto-Configuration and Auto-update process are described in [Figure 118](#), and in the description that follows the flowchart.

FIGURE 118 The DHCP Client-Based Auto-Configuration steps


Step 1. Validate the IP address and lease negotiation

1. At boot-up, the device automatically checks its configuration for an IP address.
2. If the device **does not have** a static IP address, it requests the lease of an address from the DHCP server:
 - If the server responds, it leases an IP address to the device for the specified lease period.
 - If the server does not respond (after four tries) the DHCP Client process is ended.

3. If the device **has a dynamic** address, the device asks the DHCP server to validate that address. If the server does not respond, the device will continue to use the existing address until the lease expires. If the server responds, and the IP address is outside of the DHCP address pool or has been leased to another device, it is automatically rejected, and the device receives a new IP address from the server. If the existing address is valid, the lease continues.

NOTE

The lease time interval is configured on the DHCP server, not on the client device. The **ip dhcp-client lease** command is set by the system, and is non-operational to a user.

4. If the existing address is **static**, the device keeps it and the DHCP Client process is ended.
5. For a leased IP address, when the lease interval reaches the renewal point, the device requests a renewal from the DHCP server:
 - If the device is **able** to contact the DHCP server at the renewal point in the lease, the DHCP server extends the lease. This process can continue indefinitely.
 - If the device is **unable** to reach the DHCP server after four attempts, it continues to use the existing IP address until the lease expires. When the lease expires, the dynamic IP address is removed and the device contacts the DHCP server for a new address. If the device is still unable to contact the DHCP server after four attempts, the process is ended.

The TFTP Flash image download and update step

NOTE

This process only occurs when the client device reboots, or when DHCP-client has been disabled and then re-enabled.

Once a lease is obtained from the server (described in [“Step 1. Validate the IP address and lease negotiation”](#) on page 1031), the device compares the filename of the requested flash image with the image stored in flash. In a stacking configuration, the device compares the filename with the image stored in the Active controller only.

- If the .bin filenames match, then the DHCP client skips the flash image download. If **auto-configuration** is enabled, the DHCP client proceeds with downloading the configuration files as described in [“The TFTP configuration download and update step”](#).
- If the .bin filenames are different, then the DHCP client downloads the new image from a TFTP server, then writes the downloaded image to flash. In a stacking configuration, the device copies the flash image to flash in all stack member units.

The code determines which flash (i.e., primary or secondary) to use based on how the device is booted. In a stacking configuration, the member units use the same flash as the Active controller. Once the flash is updated with the newer flash image, the device is reloaded and any member units in a stacking configuration are reloaded as well. If **auto-configuration** is enabled, the DHCP client then proceeds to download the configuration files described in [“The TFTP configuration download and update step”](#).

NOTE

In a stacking environment, the DHCP client flash image download waits 5 minutes for all member units to join and update. Then the DHCP client downloads the new image from the TFTP server using the TFTP server IP address (option 150), if it is available. If the TFTP server IP address is not available, the DHCP client requests the TFTP file from the DHCP server.

The TFTP configuration download and update step

NOTE

This process only occurs when the client device reboots, or when Auto-Configuration has been disabled and then re-enabled.

1. When the device reboots, or the Auto-Configuration feature has been disabled and then re-enabled, the device uses information from the DHCP server to contact the TFTP server to update the running-configuration file:
 - If the DHCP server provides a TFTP server name or IP address, the device uses this information to request files from the TFTP server.
 - If the DHCP server does not provide a TFTP server name or IP address, the device requests the configuration files from the DHCP server.
2. The device requests the configuration files from the TFTP server by asking for filenames in the following order:
 - **bootfile name provided by the DHCP server** (if configured)
 - **hostnameMAC-config.cfg**, for example:
FGS624p-Switch001b.ed5e.4d00-config.cfg
 - **hostnameMAC.cfg**, for example:
FGS624p-Switch001b.ed5e.4d00.cfg
 - **brocade.cfg** (applies to all devices), for example:
brocade.cfg
 - **<fcx | fgs | fls>-<switch | router>.cfg** (applies to Layer 2 or base Layer 3 devices), for example:

```
fgs-switch.cfg(FGS Layer 2)
fls-switch.cfg(FLS Layer 2)
fgs-router.cfg(FGS Base-Layer 3)
fls-router.cfg(FLS Base Layer 3)
fcx-switch.cfg(FCX Layer 2)
fcx-router.cfg(FCX Layer 3)
```

If the device is successful in contacting the TFTP server and the server has the configuration file, the files are merged. If there is a conflict, the server file takes precedence.

If the device is **unable** to contact the TFTP server or if the files are not found on the server, the TFTP part of the configuration download process ends.

Supported Options for DHCP Servers

DHCP Client supports the following options:

- 001 - subnetmask
- 003 - router ip
- 015 - domain name
- 006 - domain name server
- 012 - hostname (optional)
- 066 - TFTP server name (only used for Client-Based Auto Configuration)

- 067 - bootfile name
- 150 - TFTP server IP address (private option, datatype = IP Address)

Configuration notes for DHCP servers

- When using DHCP on a router, if you have a DHCP address for one interface, and you want to connect to the DHCP server from another interface, you must disable DHCP on the first interface, then enable DHCP on the second interface.
- When DHCP is disabled, and then re-enabled, or if the system is rebooted, the TFTP process requires approximately three minutes to run in the background before file images can be downloaded manually.
- Once a port is assigned a leased IP address, it is bound by the terms of the lease regardless of the link state of the port.

Disabling or re-enabling Auto-Configuration

For a switch, you can disable or enable this feature using the following commands.

```
Brocade(config)# ip dhcp-client enable
Brocade(config)# no ip dhcp-client enable
```

For a router, you can disable or enable this feature using the following commands.

```
Brocade(config-if-e1000-0/1/1)# ip dhcp-client enable
Brocade(config-if-e1000-0/1/1)# no ip dhcp-client enable
```

Syntax: [no] ip dhcp-client enable

Disabling or re-enabling Auto-Update

Auto-update is enabled by default. To disable it, use the following command.

```
Brocade(config)# no ip dhcp-client auto-update enabled
```

To re-enable auto-update after it has been disabled, use the following command.

```
Brocade(config)# ip dhcp-client auto-update enabled
```

Syntax: [no] ip dhcp-client auto-update enabled

Displaying DHCP configuration information

The following example shows output from the **show ip** command for Layer 2 devices.

```
Brocade(config)# show ip

Switch IP address: 10.44.16.116

Subnet mask: 255.255.255.0

Default router address: 10.44.16.1
TFTP server address: 10.44.16.41
Configuration filename: foundry.cfg
Image filename: None
```


The following example shows output from the **show ip address** command for a Layer 2 device.

```
Brocade(config)# show ip address
  IP Address      Type      Lease Time      Interface
10.44.16.116      Dynamic    174              0/1/1
```

The following example shows output from the **show ip address** command for a Layer 3 device.

```
Brocade(config)# show ip address
  IP Address      Type      Lease Time      Interface
10.44.3.233        Dynamic    672651          0/1/2
  1.0.0.1          Static     N/A              0/1/15
```

The following example shows a Layer 2 device configuration as a result of the **show run** command.

```
Brocade(config)# show run
Current configuration:
!
ver 04.2.00b47T7e1
!
module 1 fls-24-port-copper-base-module
!
!
ip address 10.44.16.116 255.255.255.0 dynamic
ip dns server-address 10.44.16.41
ip dhcp-client lease 174
ip default-gateway 10.44.16.1
!
!
end
```

NOTE

The **ip dhcp-client lease** entry in the example above applies to FastIron X Series devices only.

The following example shows a Layer 3 device configuration as a result of the **show run** command.

```
Brocade(config)# show run
Current configuration:
!
ver 04.2.00b47T7e1
!
module 1 fgs-24-port-management-module
module 2 fgs-cx4-2-port-10g-module
module 3 fgs-xfp-1-port-10g-module
!
vlan 1 name DEFAULT-VLAN by port
!
ip dns domain-name test.com
ip dns server-address 10.44.3.111
interface ethernet 0/1/2
    ip address 10.44.3.233 255.255.255.0 dynamic
    ip dhcp-client lease 691109
!
interface ethernet 0/1/15
    ip address 1.0.0.1 255.0.0.0
    ip helper-address 1 10.44.3.111
!
end
```

NOTE

The **ip dhcp-client lease** entry in the example above applies to FastIron X Series devices only.

DHCP Log messages

The following DHCP notification messages are sent to the log file.

```
2d01h48m21s:I: DHCPC: existing ip address found, no further action needed by DHCPC
2d01h48m21s:I: DHCPC: Starting DHCP Client service
2d01h48m21s:I: DHCPC: Stopped DHCP Client service
2d01h48m21s:I: DHCPC: FGS624P Switch running-configuration changed
2d01h48m21s:I: DHCPC: sending TFTP request for bootfile name fgs-switch.cfg
2d01h48m21s:I: DHCPC: TFTP unable to download running-configuration
2d01h48m21s:I: DHCPC: Found static IP Address 1.1.1.1 subnet mask 255.255.255.0 on
port 0/1/5
2d01h48m21s:I: DHCPC: Client service found no DHCP server(s) on 3 possible subnet
2d01h48m21s:I: DHCPC: changing 0/1/3 protocol from stopped to running
```

Configuring IP parameters – Layer 2 Switches

The following sections describe how to configure IP parameters on a Brocade Layer 2 Switch.

NOTE

This section describes how to configure IP parameters for Layer 2 Switches. For IP configuration information for Layer 3 Switches, refer to [“Configuring IP parameters – Layer 3 Switches”](#) on page 964.

Configuring the management IP address and specifying the default gateway

To manage a Layer 2 Switch using Telnet or Secure Shell (SSH) CLI connections or the Web Management Interface, you must configure an IP address for the Layer 2 Switch. Optionally, you also can specify the default gateway.

Brocade devices support both classical IP network masks (Class A, B, and C subnet masks, and so on) and Classless Interdomain Routing (CIDR) network prefix masks:

- To enter a classical network mask, enter the mask in IP address format. For example, enter “209.157.22.99 255.255.255.0” for an IP address with a Class-C subnet mask.
- To enter a prefix network mask, enter a forward slash (/) and the number of bits in the mask immediately after the IP address. For example, enter “209.157.22.99/24” for an IP address that has a network mask with 24 significant bits (ones).

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the display to prefix format. Refer to [“Changing the network mask display to prefix format”](#) on page 1065.

Assigning an IP address to a Brocade Layer 2 switch

To assign an IP address to a Brocade Layer 2 Switch, enter a command such as the following at the global CONFIG level.

```
Brocade(config)# ip address 192.45.6.110 255.255.255.0
```

Syntax: `ip address <ip-addr> <ip-mask>`

or

Syntax: `ip address <ip-addr>/<mask-bits>`

You also can enter the IP address and mask in CIDR format, as follows.

```
Brocade(config)# ip address 192.45.6.1/24
```

To specify the Layer 2 Switch default gateway, enter a command such as the following.

```
Brocade(config)# ip default-gateway 192.45.6.1
```

Syntax: `ip default-gateway <ip-addr>`

NOTE

When configuring an IP address on a Layer 2 switch that has multiple VLANs, make sure the configuration includes a designated management VLAN that identifies the VLAN to which the global IP address belongs. Refer to [“Designated VLAN for Telnet management sessions to a Layer 2 Switch”](#) on page 120.

Configuring Domain Name Server (DNS) resolver

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a Brocade Layer 2 Switch or Layer 3 Switch and thereby recognize all hosts within that domain. After you define a domain name, the Brocade Layer 2 Switch or Layer 3 Switch automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain “newyork.com” is defined on a Brocade Layer 2 Switch or Layer 3 Switch and you want to initiate a ping to host “NYC01” on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping.

```
Brocade# ping nyc01
Brocade# ping nyc01.newyork.com
```

Defining a DNS entry

You can define up to four DNS servers for each DNS entry. The first entry serves as the primary default address. If a query to the primary address fails to be resolved after three attempts, the next gateway address is queried (also up to three times). This process continues for each defined gateway address until the query is resolved. The order in which the default gateway addresses are polled is the same as the order in which you enter them.

Suppose you want to define the domain name of newyork.com on a Layer 2 Switch and then define four possible default DNS gateway addresses. To do so, enter the following commands.

```
Brocade(config)# ip dns domain-name newyork.com
Brocade(config)# ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25
201.98.7.15
```

Syntax: `ip dns server-address <ip-addr> [<ip-addr>] [<ip-addr>] [<ip-addr>]`

In this example, the first IP address in the `ip dns server-address...` command becomes the primary gateway address and all others are secondary addresses. Because IP address 201.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

Using a DNS name to initiate a trace route

Suppose you want to trace the route from a Brocade Layer 2 Switch to a remote server identified as NYC02 on domain newyork.com. Because the newyork.com domain is already defined on the Layer 2 Switch, you need to enter only the host name, NYC02, as noted in the following command.

```
Brocade# traceroute nyc02
```

Syntax: `traceroute <host-ip-addr> [maxttl <value>] [minttl <value>] [numeric] [timeout <value>] [source-ip <ip addr>]`

The only required parameter is the IP address of the host at the other end of the route.

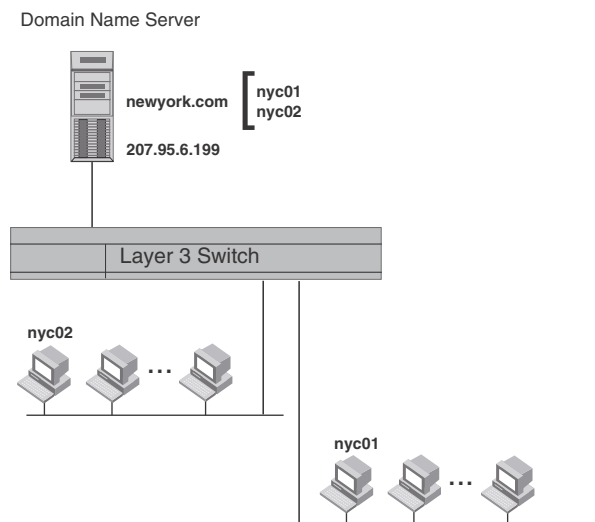
After you enter the command, a message indicating that the DNS query is in process and the current gateway address (IP address of the domain name server) being queried appear on the screen.

```
Type Control-c to abort
Sending DNS Query to 209.157.22.199
Tracing Route to IP node 209.157.22.80
To ABORT Trace Route, Please use stop-traceroute command.
Traced route to target IP node 209.157.22.80:
  IP Address      Round Trip Time1    Round Trip Time2
  207.95.6.30     93 msec            121 msec
```

NOTE

In the previous example, 209.157.22.199 is the IP address of the domain name server (default DNS gateway address), and 209.157.22.80 represents the IP address of the NYC02 host.

FIGURE 119 Querying a Host on the newyork.com Domain



Changing the TTL threshold

The time to live (TTL) threshold prevents routing loops by specifying the maximum number of router hops an IP packet originated by the Layer 2 Switch can travel through. Each device capable of forwarding IP that receives the packet decrements (decreases) the packet TTL by one. If a router receives a packet with a TTL of 1 and reduces the TTL to zero, the router drops the packet.

The default TTL is 64. You can change the TTL to a value from 1 through 255.

To modify the TTL threshold to 25, enter the following commands.

```
Brocade(config)# ip ttl 25
Brocade(config)# exit
```

Syntax: `ip ttl <1-255>`

DHCP Assist configuration

DHCP Assist allows a Brocade Layer 2 Switch to assist a router that is performing multi-netting on its interfaces as part of its DHCP relay function.

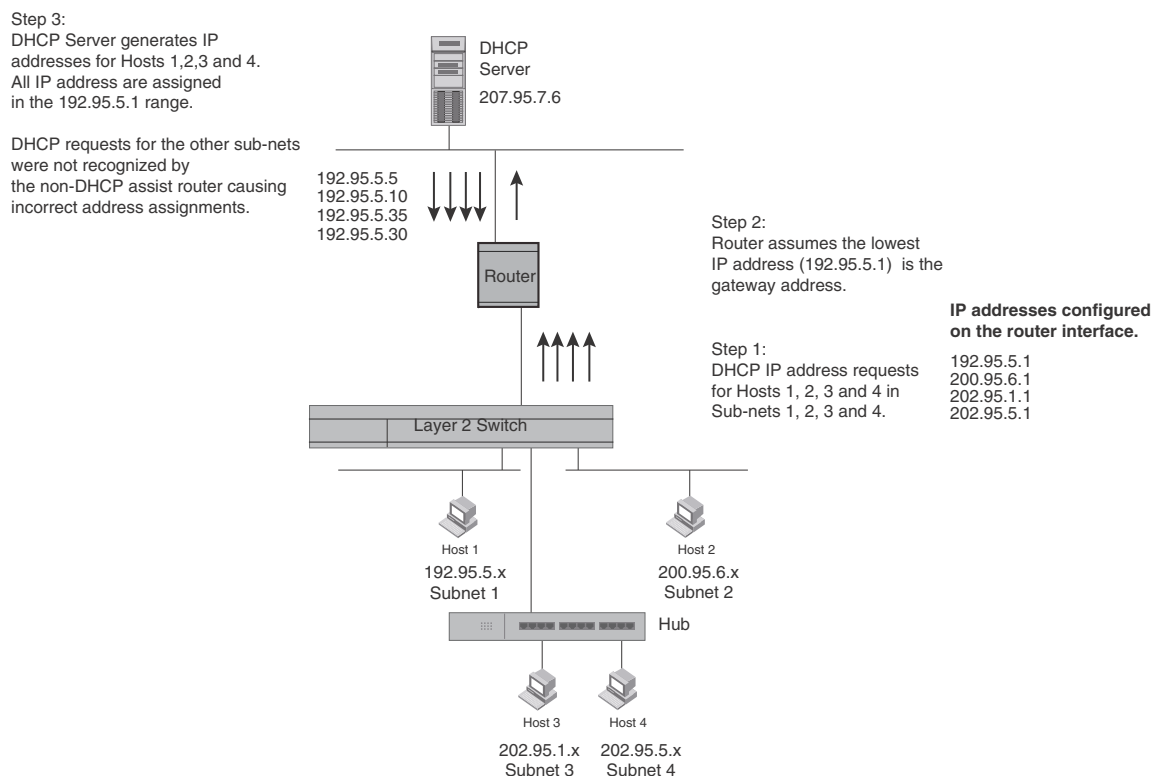
DHCP Assist ensures that a DHCP server that manages multiple IP subnets can readily recognize the requester IP subnet, even when that server is not on the client local LAN segment. The Brocade Layer 2 Switch does so by stamping each request with its IP gateway address in the DHCP discovery packet.

NOTE

Brocade Layer 3 Switches provide BootP/DHCP assistance by default on an individual port basis. Refer to [“Changing the IP address used for stamping BootP and DHCP requests”](#) on page 1012.

By allowing multiple subnet DHCP requests to be sent on the same wire, you can reduce the number of router ports required to support secondary addressing as well as reduce the number of DHCP servers required, by allowing a server to manage multiple subnet address assignments.

FIGURE 120 DHCP requests in a network without DHCP Assist on the Layer 2 Switch



In a network operating without DHCP Assist, hosts can be assigned IP addresses from the wrong subnet range because a router with multiple subnets configured on an interface cannot distinguish among DHCP discovery packets received from different subnets.

For example, in [Figure 120](#), a host from each of the four subnets supported on a Layer 2 Switch requests an IP address from the DHCP server. These requests are sent transparently to the router. Because the router is unable to determine the origin of each packet by subnet, it assumes the lowest IP address or the 'primary address' is the gateway for all ports on the Layer 2 Switch and stamps the request with that address.

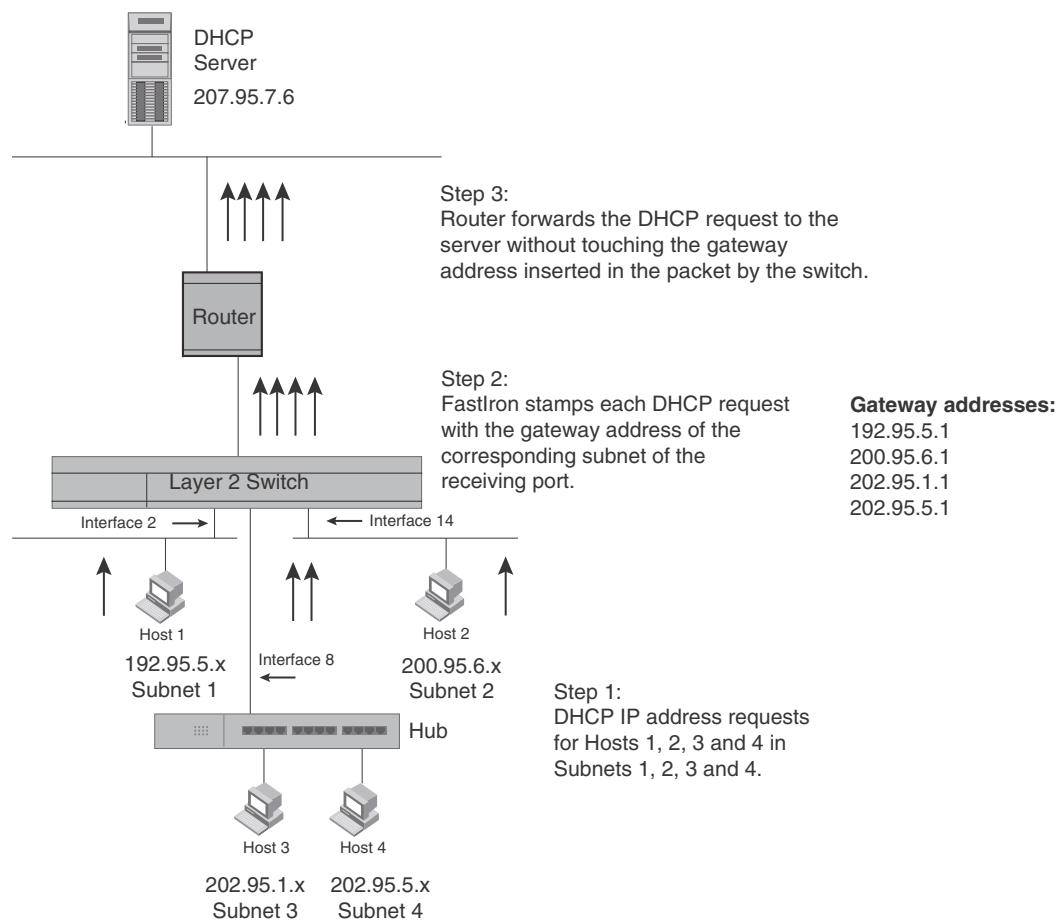
When the DHCP request is received at the server, it assigns all IP addresses within that range only.

With DHCP Assist enabled on a Brocade Layer 2 Switch, correct assignments are made because the Layer 2 Switch provides the stamping service.

How DHCP Assist works

Upon initiation of a DHCP session, the client sends out a DHCP discovery packet for an address from the DHCP server as seen in [Figure 121](#). When the DHCP discovery packet is received at a Brocade Layer 2 Switch with the DHCP Assist feature enabled, the gateway address configured on the receiving interface is inserted into the packet. This address insertion is also referred to as stamping.

FIGURE 121 DHCP requests in a network with DHCP Assist operating on a FastIron Switch



When the stamped DHCP discovery packet is then received at the router, it is forwarded to the DHCP server. The DHCP server then extracts the gateway address from each request and assigns an available IP address within the corresponding IP subnet (Figure 122). The IP address is then forwarded back to the workstation that originated the request.

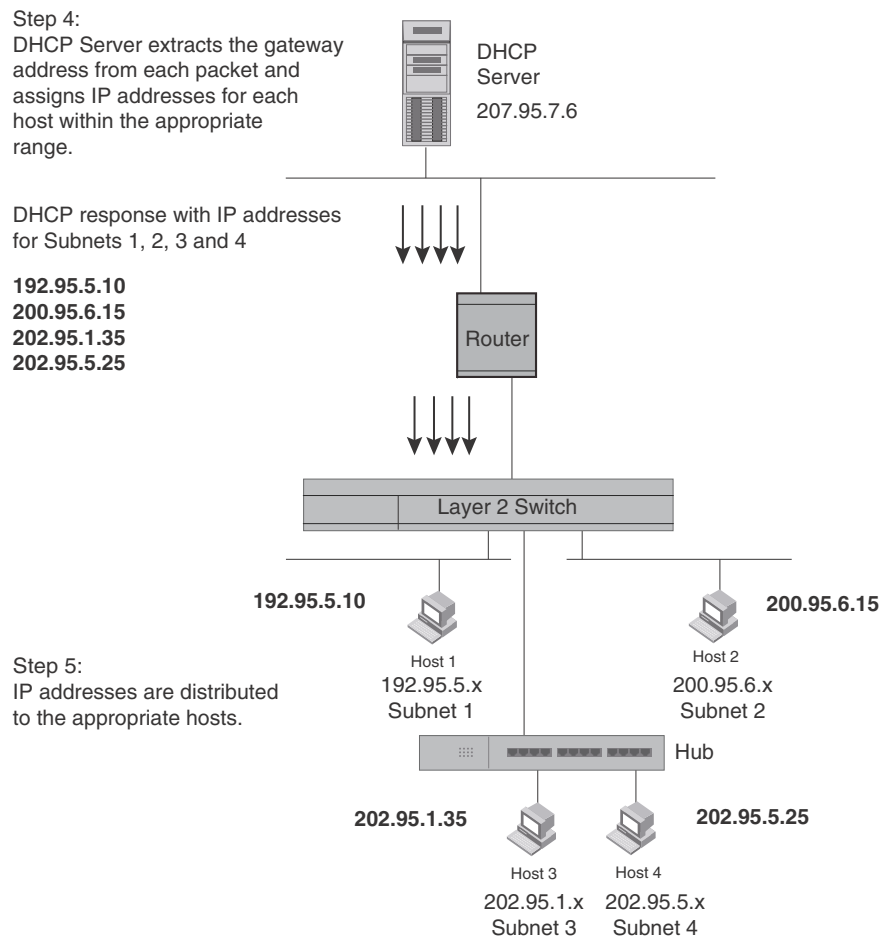
NOTE

When DHCP Assist is enabled on any port, Layer 2 broadcast packets are forwarded by the CPU. Unknown unicast and multicast packets are still forwarded in hardware, although selective packets such as IGMP, are sent to the CPU for analysis. When DHCP Assist is not enabled, Layer 2 broadcast packets are forwarded in hardware.

NOTE

The DHCP relay function of the connecting router must be turned on.

FIGURE 122 DHCP offers are forwarded back toward the requestors



NOTE

When DHCP Assist is enabled on any port, Layer 2 broadcast packets are forwarded by the CPU. Unknown unicast and multicast packets are still forwarded in hardware, although selective packets such as IGMP are sent to the CPU for analysis. When DHCP Assist is not enabled, Layer 2 broadcast packets are forwarded in hardware.

Configuring DHCP Assist

You can associate a gateway list with a port. You must configure a gateway list when DHCP Assist is enabled on a Brocade Layer 2 Switch. The gateway list contains a gateway address for each subnet that will be requesting addresses from a DHCP server. The list allows the stamping process to occur. Each gateway address defined on the Layer 2 Switch corresponds to an IP address of the Brocade router interface or other router involved.

Up to eight addresses can be defined for each gateway list in support of ports that are multi-homed. When multiple IP addresses are configured for a gateway list, the Layer 2 Switch inserts the addresses into the discovery packet in a round robin fashion.

Up to 32 gateway lists can be defined for each Layer 2 Switch.

Example

To create the configuration indicated in [Figure 121](#) and [Figure 122](#), enter commands such as the following.

```
Brocade(config)# dhcp-gateway-list 1 192.95.5.1
Brocade(config)# dhcp-gateway-list 2 200.95.6.1
Brocade(config)# dhcp-gateway-list 3 202.95.1.1 202.95.5.1
Brocade(config)# interface ethernet 2
Brocade(config-if-e1000-2)# dhcp-gateway-list 1
Brocade(config-if-e1000-2)# interface ethernet 8
Brocade(config-if-e1000-8)# dhcp-gateway-list 3
Brocade(config-if-e1000-8)# interface ethernet 14
Brocade(config-if-e1000-14)# dhcp-gateway-list 2
```

Syntax: `dhcp-gateway-list <num> <ip-addr>`

IPv4 point-to-point GRE tunnels

NOTE

This feature is supported on FCX and FastIron SX IPv6-capable devices only.

This section describes support for point-to-point Generic Routing Encapsulation (GRE) tunnels and how to configure them on a Brocade device.

GRE tunnels support includes, but is not limited to, the following:

- IPv4 over GRE tunnels. IPv6 over GRE tunnels is not supported.
- Static and dynamic unicast routing over GRE tunnels
- Multicast routing over GRE tunnels
- Hardware forwarding of IP data traffic across a GRE tunnel.
- Path MTU Discovery (PMTUD)

IPv4 GRE tunnel overview

Generic Routing Encapsulation is described in RFC 2784. Generally, GRE provides a way to encapsulate arbitrary packets (payload packet) inside of a transport protocol, and transmit them from one tunnel endpoint to another. The payload is encapsulated in a GRE packet. The resulting GRE packet is then encapsulated in a delivery protocol, then forwarded to the tunnel destination. At the tunnel destination, the packet is decapsulated to reveal the payload. The payload is then forwarded to its final destination.

Brocade IPv6-capable devices allow the tunneling of packets of the following protocols over an IPv4 network using GRE:

- OSPF V2
- BGP4
- RIP V1 and V2

NOTE

This is not supported on ICX 6450 devices.

GRE packet structure and header format

Figure 123 shows the structure of a GRE encapsulated packet.

FIGURE 123 GRE encapsulated packet structure

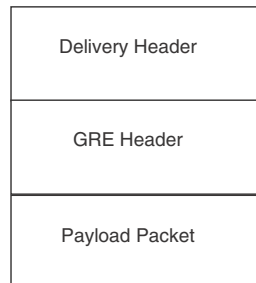


Figure 124 shows the GRE header format.

FIGURE 124 GRE header format

Checksum	Reserved0	Ver	Protocol Type	Checksum (optional)	Reserved (optional)
----------	-----------	-----	---------------	---------------------	---------------------

The GRE header has the following fields:

- **Checksum** – 1 bit. This field is assumed to be zero in this version. If set to 1, this means that the **Checksum (optional)** and **Reserved (optional)** fields are present and the **Checksum (optional)** field contains valid information.
- **Reserved0** – 12 bits. If bits 1 - 5 are non-zero, then a receiver must discard the packet unless RFC 1701 is implemented. Bits 6 - 12 are reserved for future use and must be set to zero in transmitted packets. This field is assumed to be zero in this version.

- **Ver** – 3 bits. The GRE protocol version. This field must be set to zero in this version.
- **Protocol Type** – 16 bits. The Ethernet protocol type of the packet, as defined in RFC 1700.
- **Checksum (optional)** – 16 bits. This field is optional. It contains the IP checksum of the GRE header and the payload packet.
- **Reserved (optional)** – 16 bits. This field is optional. It is reserved for Brocade internal use.

Path MTU Discovery (PMTUD) support

Brocade IronWare software supports the following RFCs for handling large packets over a GRE tunnel:

- RFC 1191, Path MTU Discovery
- RFC 4459, MTU and Fragmentation Issues with In-the-Network Tunneling

RFC 1191 describes a method for dynamically discovering the maximum transmission unit (MTU) of an arbitrary internet path. When a FastIron device receives an IP packet that has its Do not Fragment (DF) bit set, and the packet size is greater than the MTU value of the outbound interface, then the FastIron device returns an ICMP Destination Unreachable message to the source of the packet, with the code indicating "fragmentation needed and DF set". The ICMP Destination Unreachable message includes the MTU of the outbound interface. The source host can use this information to help determine the minimum MTU of a path to a destination.

RFC 4459 describes solutions for issues with large packets over a tunnel. The following methods, from RFC 4459, are supported in Brocade IronWare software:

- If a source attempts to send packets that are larger than the lowest MTU value along the path, PMTUD can signal to the source to send smaller packets. This method is described in Section 3.2 of RFC 4459.
- Inner packets can be fragmented before encapsulation, in such a manner that the encapsulated packet fits in the tunnel path MTU, which is discovered using PMTUD. This method is described in Section 3.4 of RFC 4459.

By default, PMTUD is enabled.

Configuration considerations for PMTUD support

Consider the following when configuring PMTUD support.

- On FCX devices, only eight different MTU values can be configured over the whole system. When the SX-FI48GPP module is installed in the FastIron SX device, the maximum number of different MTU values that can be configured is 16.
- On both FCX devices, and the SX-FI-24GPP, SX-FI48GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI-8XG modules, PMTUD will not be enabled on the device if the maximum number of MTU values has already been configured in the system.
 - When a new PMTUD value is discovered, and the maximum number of different MTU values for the system is already configured, the new value will search for the nearest, but smallest MTU value relative to its own value in the system. For example, in a FCX system, the new PMTUD value is 800, and the eight different MTU values configured in the system are 600, 700, 820, 1000, 1100, 1200, 1300, and 1500. The range of MTU values that can be configured is from 576 through 1500. The new PMTUD value 800 cannot be added to the system so the nearest, but smallest MTU value is used. In this example, the MTU value of 700 is considered as the nearest MTU value already configured in the system.

- When the new PMTUD value is smaller than all of the eight MTU values configured in the system, the PMTUD feature is disabled for the tunnel, and the value is not added to the system. For example, the new PMTUD value is 620 which is smaller in value than all of the eight, different MTU path values configured in the system. The following warning message is displayed on the CLI:

```
Warning - All MTU profiles used, disabling PMTU for tunnel <tunnel_id>; new PMTU was <new pmtu discovered>
```

Tunnel loopback ports for GRE tunnels

For SX-FI624C, SX-FI624P, SX-FI624HF, and SX-FI62XG modules a physical tunnel loopback port is required for routing a decapsulated packet. When a GRE-encapsulated packet is received on a tunnel interface, and the packet needs to be decapsulated, the packet is decapsulated and sent to the tunnel loopback port. The packet is then looped back and forwarded based on the payload packets.

If a tunnel loopback port is not configured, tunnel termination is performed by the CPU. Each GRE tunnel interface can have one assigned tunnel loopback port and the same tunnel loopback port can be used for multiple tunnels.

Tunnel loopback ports for GRE tunnels are supported on:

- untagged ports
- ports that are enabled by default
- 10 Gbps and 1 Gbps copper and fiber ports

Note the following hardware limitations for these port types:

- On 10 Gbps ports, the port LEDs will be ON (green) when the ports are configured as tunnel loopback ports for GRE tunnels. Also, the LEDs will blink when data packets are forwarded.
- On 1 Gbps fiber and copper ports, port LEDs will *not* be ON when the ports are configured as tunnel loopback ports for GRE tunnels, nor will the LEDs blink when data packets are forwarded.

Tunnel loopback ports for GRE tunnels are *not* applicable on:

- tagged ports
- trunk ports
- ports that are members of a VE
- ports that are disabled
- ports that have an IP address
- flow control
- the SX-FI48GPP module

Support for IPv4 multicast routing over GRE tunnels

PIM-DM and PIM-SM Layer 3 multicast protocols and multicast data traffic are supported over GRE tunnels. When a multicast protocol is enabled on both ends of a GRE tunnel, multicast packets can be sent from one tunnel endpoint to another. To accomplish this, the packets are encapsulated using the GRE unicast tunneling mechanism and forwarded like any other IPv4 unicast packet to

the destination endpoint of the tunnel. The router that terminates the tunnel (i.e., the router where the tunnel endpoint is an ingress interface) de-encapsulates the GRE tunneled packet to retrieve the native multicast data packets. After de-encapsulation, data packets are forwarded in the direction of its receivers, and control packets may be consumed. This creates a PIM-enabled virtual or logical link between the two GRE tunnel endpoints.

Strict RPF check for multicast protocols

IronWare software enforces strict Reverse Path Forwarding (RPF) check rules on an (s,g) entry on a GRE tunnel interface. The (s,g) entry uses the GRE tunnel as an RPF interface. During unicast routing transit, GRE tunnel packets may arrive at different physical interfaces. The strict RPF check limits GRE PIM tunnel interfaces to accept the (s,g) GRE tunnel traffic.

NOTE

For the FESX624 device, and the SX-FI624C, SX-FI624P, SX-FI624HF, and the SX-FI62XG modules loopback ports are required for de-encapsulating the GRE tunneled packet. On these hardware devices, when the GRE-encapsulated multicast packet is received, the unicast GRE mechanism takes care of de-encapsulating the packet. The packet then egresses and re-ingresses the tunnel interface loopback port as the native multicast packet. The hardware RPF check is done, not on the tunnel interface directly, but on the loopback port - the hardware compares this port number with the port number configured in the Multicast table (s,g) entry. If they match, the packet is routed. Otherwise it is sent to the CPU for error processing. In unicast, it is permissible for multiple tunnel interfaces to use a single loopback port. However, in multicast, this will not allow the hardware to determine the tunnel interface that the packet was received on in order to do an RPF check. Therefore, when IPv4 Multicast Routing is enabled on a GRE tunnel, the tunnel interface must have a dedicated loopback port.

GRE support with other features

This section describes how GRE tunnels may affect other features on FESX, FSX, and FCX devices.

Support for ECMP for routes through a GRE tunnel

Equal-Cost Multi-Path (ECMP) load sharing allows for load distribution of traffic among available routes. When GRE is enabled, a mix of GRE tunnels and normal IP routes is supported. If multiple routes are using GRE tunnels to a destination, packets are automatically load-balanced between tunnels, or between tunnels and normal IP routes.

ACL, QoS, and PBR support for traffic through a GRE tunnel

NOTE

PBR and ACL filtering for packets terminating on a GRE tunnel is not supported on FCX devices. However, PBR can be used to map IP traffic into a GRE tunnel, but it cannot be used to route GRE traffic. On FCX devices, QoS support for GRE encapsulated packets is limited to copying DSCP values from the inner header onto the outer header.

For FastIron SX devices only, traffic coming from a tunnel can be filtered by an ACL both before and after the tunnel is terminated and also redirected by PBR after tunnel is terminated. An ACL classifies and sets QoS for GRE traffic. If the ACL or PBR is applied to the tunnel loopback port, it would apply to the inner IP packet header (the payload packet) after the tunnel is terminated. If the ACL is applied to the tunnel ingress port, then the delivery header (outer header) would be classified or filtered before the tunnel is terminated.

NOTE

Restrictions for using ACLs in conjunction with GRE are noted in the section [“Configuration considerations for GRE IP tunnels”](#) on page 1048. PBR can be configured on tunnel loopback ports for tunnel interfaces with no restrictions. PBR with GRE tunnel is not supported on FSX 800 and FSX 1600 with the SX-FI48GPP module.

Syslog messages related to GRE IP tunnels

Syslog messages provide management applications with information related to GRE IP tunnels. The following Syslog message is supported.

```
Tunnel: TUN-RECURSIVE-DOWN tnnl 1, Tnl disabled due to recursive routing
```

Configuration considerations for GRE IP tunnels

Before configuring GRE tunnels and tunnel options, consider the configuration notes in this section.

- GRE tunnels are not supported in a mixed hardware configuration with 48-port 10/100/1000 Mbps Ethernet POE (SX-FI48GPP) interface modules, together with IPv6-capable interface modules, or management modules with user ports.
- The mix and match mode for GRE and IPv6 tunnels are not supported.
- Hitless management is supported for GRE tunnels on any FastIron devices. Hitless management is not supported for IPv6-over-IPv4 tunnels on all FastIron devices. When IPv6 tunnels are configured, the CLI commands that execute a hitless switchover (**switch-over-active-role** command and the **hitless reload** command) are disabled.
- When GRE is enabled on a Layer 3 switch, the following features are not supported on Virtual Ethernet (VE) ports, VE member ports (ports that have IP addresses), and GRE tunnel loopback ports:
 - ACL logging
 - ACL statistics (also called ACL counting)
 - MAC address filters
 - IPv6 filters

NOTE

The above features are supported on VLANs that do not have VE ports.

- Whenever multiple IP addresses are configured on a tunnel source, the primary address of the tunnel is always used for forming the tunnel connections. Therefore, carefully check the configurations when configuring the tunnel destination.

- When a GRE tunnel is configured, you cannot configure the same routing protocol on the tunnel through which you learn the route to the tunnel destination. For example, if the FastIron learns the tunnel destination route through the OSPF protocol, you cannot configure the OSPF protocol on the same tunnel and vice-versa. When a tunnel has OSPF configured, the FastIron cannot learn the tunnel destination route through OSPF. This could cause the system to become unstable.
- The tunnel destination cannot be resolved to the tunnel itself or any other local tunnel. This is called recursive routing. This scenario would cause the tunnel interface to flap and the Syslog message TUN-RECURSIVE-DOWN to be logged. To resolve this issue, create a static route for the tunnel destination.

Configuration considerations for tunnel loopback ports

NOTE

The configuration considerations for tunnel loopback ports are supported only on FESX and FSX devices.

Consider the following when configuring tunnel loopback ports for GRE tunnels:

- For multicast traffic over a GRE tunnel, each PIM-enabled tunnel interface must have a dedicated tunnel loopback port.
- For unicast traffic, a tunnel loopback port can be oversubscribed, meaning multiple GRE tunnels (up to the maximum supported) can use the same tunnel loopback port for traffic. When oversubscribed, proper traffic classification on the tunnel loopback port is necessary in order to avoid traffic congestion. In this case, Brocade recommends that you configure the trust level at the DSCP level for QoS by adding an ACL that maps DSCP 46 to priority 5. Otherwise, loss of loopback packets may flap the tunnel interface.
- By default, when you create a tunnel loopback port for a GRE tunnel on a port that is part of the default VLAN, the port will stay in the default VLAN. Before configuring a port as a tunnel loopback port for a GRE tunnel, if the port is in the default VLAN (VLAN 1), first create a VLAN, then add the port to the VLAN. Otherwise, an error message such as the following will appear on the console when you attempt to configure a router interface for the default VLAN.


```
ERROR: Router-interface cannot be applied because of GRE loopback port 1/2
```
- Configuration of tunnel loopback ports are not applicable on the SX-FI48GPP interface module.

GRE MTU configuration considerations

The default Maximum Transmission Unit (MTU) value for packets in a GRE tunnel is 1476 bytes, or 10194 bytes for jumbo packets. The MTU of the GRE tunnel is compared with the outgoing packet before the packet is encapsulated. After encapsulation, the packet size increases by 24 bytes. Therefore, when changing the GRE tunnel MTU, set the MTU to at least 24 bytes less than the IP MTU of the outgoing interface. If the MTU is not set to at least 24 bytes less than the IP MTU, the size of the encapsulated packet will exceed the IP MTU of the outgoing interface. This will cause the packet to either be sent to the CPU for fragmentation, or the packet will be dropped if the DF (Do-Not-Fragment) bit is set in the original IP packet, and an ICMP message is sent.

Configuration tasks for GRE tunnels

Brocade recommends that you perform the configuration tasks in the order listed in Table 179.

TABLE 179 Configuration tasks for GRE tunnels

Configuration tasks	Default behavior	For more information
Required tasks		
1 Create a tunnel interface	Not assigned	“Creating a tunnel interface” on page 1051
2 Configure the source address or source interface for the tunnel interface	Not assigned	“Configuring the source address or source interface for a tunnel interface” on page 1051
3 Configure the destination address of the tunnel interface	Not assigned	“Configuring the destination address for a tunnel interface” on page 1052
4 Enable GRE encapsulation on the tunnel interface	Disabled	“Enabling GRE encapsulation on a tunnel interface” on page 1052
NOTE: Step 4 must be performed before step 6.		
5 If packets need to be terminated in hardware, configure a tunnel loopback port for the tunnel interface	Not assigned	“Tunnel loopback ports for GRE tunnels” on page 1046
NOTE: Step 5 is not applicable to FCX devices.		
6 Configure an IP address for the tunnel interface	Not assigned	“Configuring an IP address for a tunnel interface” on page 1054
7 If a route to the tunnel destination (configured in Step 3) does not already exist, create a static route and specify that the route is through the tunnel interface.	Not assigned	“Configuring a static route to a tunnel destination” on page 1054
Optional tasks		
1 Change the maximum transmission unit (MTU) value for the tunnel interface	1476 bytes or 10194 bytes (jumbo mode)	“Changing the MTU value for a tunnel interface” on page 1055
2 Change the number of GRE tunnels supported on the device	Support for 32 GRE tunnels	“Changing the maximum number of tunnels supported” on page 1055
3 Enable and configure GRE link keepalive on the tunnel interface	Disabled	“Configuring GRE link keepalive” on page 1056
4 Change the Path MTU Discovery (PMTUD) configuration on the GRE tunnel interface	Enabled	“Configuring Path MTU Discovery (PMTUD)” on page 1056
5 Enable support for IPv4 multicast routing	Disabled	“Enabling IPv4 multicast routing over a GRE tunnel” on page 1057

The following features are also supported on GRE tunnel interfaces:

- Naming the tunnel interface (CLI command **port-name**) – for configuration details, refer to [“Assigning a port name”](#) on page 40.
- Changing the Maximum Transmission Unit (MTU) (CLI command **ip mtu**) – for configuration details, refer to [“Changing the MTU on an individual port”](#) on page 975.

- Increasing the cost of routes learned on the port (CLI command **ip metric**) – for configuration details, refer to [“Changing the cost of routes learned on a port”](#) on page 1203.

After performing the configuration steps listed in Table 179, you can view the GRE configuration and observe the routes that use GRE tunnels. For details, refer to [“Displaying GRE tunneling information”](#) on page 1059.

Creating a tunnel interface

To create a tunnel interface, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)#
```

Syntax: [no] interface tunnel <tunnel-number>

The <tunnel-number> is a numerical value that identifies the tunnel being configured.

NOTE

You can also use the **port-name** command to name the tunnel. To do so, follow the configuration instructions in [“Assigning a port name”](#) on page 40.

Configuring the source address or source interface for a tunnel interface

To configure the source for a tunnel interface, specify either a source address or a source interface.

NOTE

If the destination address for a tunnel interface is not resolved, Brocade recommends that you either configure *source interface* (instead of *source address*) as the source for a tunnel interface, or enable GRE link keepalive on the tunnel interface.

The tunnel **source address** should be one of the router IP addresses configured on a physical, loopback, or VE interface, through which the other end of the tunnel is reachable.

To configure the source address for a specific tunnel interface, enter commands such as the following.

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel source 36.0.8.108
```

The **source interface** should be the port number of the interface configured on a physical, loopback, or VE interface. The source interface should have at least one IP address configured on it. Otherwise, the interface will not be added to the tunnel configuration and an error message similar to the following will be displayed:

```
ERROR - Tunnel source interface 3/1 has no configured IP address.
```

To configure the source interface for a specific tunnel interface, enter commands such as the following.

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel source ethernet 3/1
```

Syntax: [no] tunnel source <ip-address> | ethernet <portnum> | ve <number> | loopback <number>

The <ip-address> variable is the source IP address being configured for the specified tunnel.

The ethernet <portnum> variable is the source slot (chassis devices only) and port number of the physical interface being configured for the specified tunnel, for example 3/1.

The ve <number> variable is the VE interface number being configured for the specified tunnel.

Deleting an IP address from an interface configured as a tunnel source

To delete an IP address from an interface that is configured as a tunnel source, first remove the tunnel source from the tunnel interface then delete the IP address, as shown in the following example.

```
Brocade(config-if-e1000-1/3)# interface tunnel 8
Brocade(config-tnif-8)# no tunnel source 45.1.83.15
Brocade(config-tnif-8)# interface ethernet 1/3
Brocade(config-if-e1000-1/3)# no ip address 45.1.83.15/24
```

If you attempt to delete an IP address without first removing the tunnel source, the console will display an error message, as shown in the following example.

```
Brocade# config terminal
Brocade(config)# interface ethernet 1/3
Brocade(config-if-e1000-1/3)# no ip address 45.1.83.15/24
Error - Please remove tunnel source from tnnl 8 before removing IP address
```

NOTE

The previous error message will also display on the CLI when an interface is part of a VLAN. A VLAN cannot be deleted until the tunnel source is first removed.

Configuring the destination address for a tunnel interface

The destination address should be the address of the IP interface of the device on the other end of the tunnel.

To configure the destination address for a specific tunnel interface, enter commands such as the following.

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel destination 131.108.5.2
```

Syntax: [no] tunnel destination <ip-address>

The <ip-address> variable is the destination IP address being configured for the specified tunnel.

NOTE

Ensure a route to the tunnel destination exists on the tunnel source device. Create a static route if necessary. For configuration details, refer to [“Configuring a static route to a tunnel destination”](#) on page 1054.

Enabling GRE encapsulation on a tunnel interface

To enable GRE encapsulation on a tunnel interface, enter commands such as the following.

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel mode gre ip
```

Syntax: [no] tunnel mode gre ip

- **gre** specifies that the tunnel will use GRE encapsulation (IP protocol 47).
- **ip** specifies that the tunneling protocol is IPv4.

NOTE

Before configuring a new GRE tunnel, the system should have at least one slot available for adding the default tunnel MTU value to the system tables. Depending on the configuration, the default tunnel MTU range is ((1500 or 10218) - 24) . To check for slot availability, or to see if the MTU value is already configured in the IP table, use the **show ip mtu** command. For more information on the **show ip mtu** command, refer to [“Displaying multicast protocols and GRE tunneling information”](#) on page 1062.

Configuring a tunnel loopback port for a tunnel interface

NOTE

Configuring a tunnel loopback port for a tunnel interface is not applicable on FCX devices, and SX-FI-24GPP, SX-FI48GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI-8XG modules.

For details and important configuration considerations regarding tunnel loopback ports for GRE tunnels, refer to [“Tunnel loopback ports for GRE tunnels”](#) on page 1046 and [“Configuration considerations for tunnel loopback ports”](#) on page 1049.

To configure a tunnel loopback port, enter commands such as the following:

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel loopback 3/1
```

Syntax: [no] tunnel loopback <portnum>

The <portnum> is the slot (chassis devices) and port number of the tunnel loopback port for the specified tunnel interface, for example 3/1.

Applying an ACL or PBR to a tunnel interface on a FastIron X Series module

To apply an ACL or PBR policy to a tunnel interface on a FastIron X Series module other than the SX-FI48GPP (48-port 10/100/1000 Mbps Ethernet POE interface module), enter commands such as the following:

Applying a PBR policy to a tunnel interface

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel mode gre ip
Brocade(config-tnif-1)# tunnel loopback 3
Brocade(config-tnif-1)# interface ethernet 3
Brocade(config-if-e1000-3)# ip policy route-map test-route
```

Applying an ACL policy to a tunnel interface

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel mode gre ip
Brocade(config-tnif-1)# tunnel loopback 3
Brocade(config-tnif-1)# interface ethernet 3
Brocade(config-if-e1000-3)# ip access-group 10 in
```

Applying an ACL or PBR to a tunnel interface on the SX-FI48GPP interface module

To apply an ACL or PBR policy to a tunnel interface on the SX-FI48GPP interface module, enter commands such as the following:

NOTE

Configuration of tunnel loopback ports are not applicable on the SX-FI48GPP interface module.

Applying a PBR policy to a tunnel interface

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel mode gre ip
Brocade(config-tnif-1)# ip policy route-map test-route
```

Applying an ACL policy to a tunnel interface

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel mode gre ip
Brocade(config-tnif-1)# ip access-group 10 in
```

Configuring an IP address for a tunnel interface

An IP address sets a tunnel interface as an IP port and allows the configuration of Layer 3 protocols, such as OSPF, BGP, and Multicast (PIM-DM and PIM-SM) on the port. Note that the subnet cannot overlap other subnets configured on other routing interfaces, and both ends of the tunnel should be in the same subnet, as illustrated in the GRE tunnel configuration example in [Figure 125](#) on page 1058.

To configure an IP address for a specified tunnel interface, enter commands such as the following.

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# ip address 10.10.3.1/24
```

Syntax: [no] ip address <ip-address>

The <ip-address> is the IP address being configured for the specified tunnel interface.

Configuring a static route to a tunnel destination

If a route to the tunnel destination does not already exist on the tunnel source, create a static route and set the route to go through the tunnel interface.

```
Brocade(config)# ip route 131.108.5.0/24 36.0.8.1
Brocade(config)# ip route 10.10.2.0/24 tunnel 1
```

Syntax: [no] ip route <ip-address> tunnel <tunnel-ID>

- The <ip-address> variable is the IP address of the tunnel interface.
- The <tunnel-ID> variable is a valid tunnel number or name.

Changing the MTU value for a tunnel interface

For important configuration considerations regarding this feature, refer to [“GRE MTU configuration considerations”](#) on page 1049.

You can set an MTU value for packets entering the tunnel. Packets that exceed either the default MTU value of 1476/10194 bytes (for jumbo case) or the value that you set using this command, are fragmented and encapsulated with IP/GRE headers for transit through the tunnel (if they do not have the DF bit set in the IP header). All fragments will carry the same DF bit as the incoming packet. Jumbo packets are supported, although they may be fragmented based on the configured MTU value.

NOTE

For the SX-FI8GMR6, SX-FI2XGMR6, SX-FI624HF, SX-FI624C, SX-FI624P, and the SX-FI62XG modules, all fragments will carry the same DF bit as the incoming packet. For the SX-FI-24GPP, SX-FI48GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI-8XG modules and the FCX modules, the DF bit on the outer IP header after encapsulation will be set if the PMTU is enabled. If PMTU is disabled, the DF bit will be unset irrespective of the DF bit of the incoming packet.

The following command allows you to change the MTU value for packets transiting “tunnel 1”:

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# ip mtu 1200
```

Syntax: `ip mtu <packet-size>`

The `<packet-size>` variable specifies the maximum size in bytes for the packets transiting the tunnel. Enter a value from 576 through 1476. The default value is 1476.

NOTE

To prevent packet loss after the 24 byte GRE header is added, make sure that any physical interface that is carrying GRE tunnel traffic has an IP MTU setting at least 24 bytes greater than the tunnel MTU setting. This configuration is only allowed on the system if the tunnel mode is set to GRE.

Changing the maximum number of tunnels supported

By default, FastIron IPv6 devices support up to 32 GRE tunnels. You can configure the device to support 16 – 64 GRE tunnels. To change the maximum number of tunnels supported, enter commands such as the following.

```
Brocade(config)# system-max gre-tunnels 16
Reload required. Please write memory and then reload or power cycle.
Brocade(config)# write memory
Brocade(config)# exit
Brocade# reload
```

NOTE

You must save the configuration (write memory) and reload the software to place the change into effect.

Syntax: `system-max gre-tunnels <number>`

The `<number>` variable specifies the number of GRE tunnels that can be supported on the device. The permissible range is 16 – 64. The **system-max gre-tunnels** command determines the interface range that is supported for an interface tunnel. For example, if the system-max value is reduced, it is possible that the configured interfaces may be rejected after a system reload.

Configuring GRE link keepalive

When GRE tunnels are used in combination with static routing or policy-based routing, and a dynamic routing protocol such as RIP, BGP, or OSPF is not deployed over the GRE tunnel, a configured tunnel does not have the ability to bring down the line protocol of either tunnel endpoint, if the far end becomes unreachable. Traffic sent on the tunnel cannot follow alternate paths because the tunnel is always UP. To avoid this scenario, enable GRE link keepalive, which will maintain or place the tunnel in an UP or DOWN state based upon the periodic sending of keepalive packets and the monitoring of responses to the packets. If the packets fail to reach the tunnel far end more frequently than the configured number of retries, the tunnel is placed in the DOWN state.

To enable GRE link keepalive, configure it on one end of the tunnel and ensure the other end of the tunnel has GRE enabled.

To configure GRE link keepalive, enter commands such as the following.

```
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# keepalive 12 4
```

These commands configure the device to wait for 4 consecutive lost keepalive packets before bringing the tunnel down. There will be a 12 second interval between each packet. Note that when the tunnel comes up, it would immediately (within one second) send the first keepalive packet.

Syntax: `[no] keepalive <seconds> <retries>`

Use the **no** form of the command to disable the keepalive option.

The `<seconds>` variable specifies the number of seconds between each initiation of a keepalive message. The range for this interval is 2 – 32767 seconds. The default value is 10 seconds.

The `<retries>` variable specifies the number of times that a packet is sent before the system places the tunnel in the DOWN state. Possible values are from 1 through 255. The default number of retries is 3.

Use the **show interface tunnel** and **show ip tunnel traffic** commands to view the GRE link keepalive configuration. For details, refer to [“Displaying GRE tunneling information”](#) on page 1059.

Configuring Path MTU Discovery (PMTUD)

Path MTU Discovery (PMTUD) support is described in the section [“Path MTU Discovery \(PMTUD\) support”](#) on page 1045. PMTUD is enabled by default on tunnel interfaces. This section describes how to disable and re-enable PMTUD on a tunnel interface, change the PMTUD age timer, manually clear the tunnel PMTUD, and view the PMTUD configuration.

NOTE

For the SX-FI8GMR6, SX-FI2XGMR6, SX-FI624HF, SX-FI624C, SX-FI624P, and the SX-FI62XG modules, all fragments will carry the same DF bit as the incoming packet. For the SX-FI-24GPP, SX-FI48GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI-8XG modules and the FCX modules, the DF bit on the outer IP header after encapsulation will be set if the PMTU is enabled. If PMTU is disabled, the DF bit will be unset irrespective of the DF bit of the incoming packet.

Disabling and re-enabling PMTUD

PMTUD is enabled by default. To disable it, enter the following command:

```
Brocade(config-tnif-1)# tunnel path-mtu-discovery disable
```

To re-enable PMTUD after it has been disabled, enter the following command:

```
Brocade(config-tnif-1)# no tunnel path-mtu-discovery disable
```

Syntax: [no] tunnel path-mtu-discovery disable

Changing the age timer for PMTUD

By default, when PMTUD is enabled on a tunnel interface, the path MTU is reset to its original value every 10 minutes. If desired, you can change the reset time (default age timer) to a value of up to 30 minutes. To do so, enter a command such as the following on the GRE tunnel interface.

```
Brocade(config-tnif-1)# tunnel path-mtu-discovery age-timer 20
```

This command configures the device to wait for 20 minutes before resetting the path MTU to its original value.

Syntax: [no] tunnel path-mtu-discovery age-timer <minutes> | infinite

For <minutes>, enter a value from 10 to 30.

Enter **infinite** to disable the timer.

Clearing the PMTUD dynamic value

To reset a dynamically-configured MTU on a tunnel Interface back to the configured value, enter the following command.

```
Brocade(config)# clear ip tunnel pmtud 1
```

Syntax: clear ip tunnel pmtud <tunnel-ID>

The <tunnel-ID> variable is a valid tunnel number or name.

Viewing PMTUD configuration details

Use the **show interface tunnel** command to view the PMTUD configuration and to determine whether PMTUD has reduced the size of the MTU. For details about the **show interface tunnel** command, refer to [“Displaying GRE tunneling information”](#) on page 1059.

Enabling IPv4 multicast routing over a GRE tunnel

This section describes how to enable IPv4 multicast protocols, PIM Sparse (PIM-SM) and PIM Dense (PIM-DM), on a GRE tunnel. Perform the procedures in this section after completing the required tasks in [Table 179](#) on page 1050.

For an overview of multicast routing support over a GRE tunnel, refer to [“Support for IPv4 multicast routing over GRE tunnels”](#) on page 1046. To view information about multicast protocols and GRE tunnel-specific information, refer to [“Displaying multicast protocols and GRE tunneling information”](#) on page 1062.

NOTE

For the FESX624 module, and the SX-FI624C, SX-FI624P, SX-FI624HF, and the SX-FI62XG modules, each PIM-enabled tunnel interface must have a *dedicated* tunnel loopback port. This differs from GRE tunnels that support unicast traffic only. For unicast traffic, multiple GRE tunnels can use the same tunnel loopback port for traffic.

Enabling PIM-SM on a GRE tunnel

To enable PIM-SM on a GRE tunnel interface, enter commands such as the following:

```
Brocade(config)# interface tunnel 10
Brocade(config-tunif-10)# ip pim-sparse
```

Syntax: [no] ip pim-sparse

Use the **no** form of the command to disable PIM-SM on the tunnel interface.

Enabling PIM-DM on a GRE tunnel interface

To enable PIM-DM on a GRE tunnel interface, enter commands such as the following:

```
Brocade(config)# interface tunnel 10
Brocade(config-tunif-10)# ip pim
```

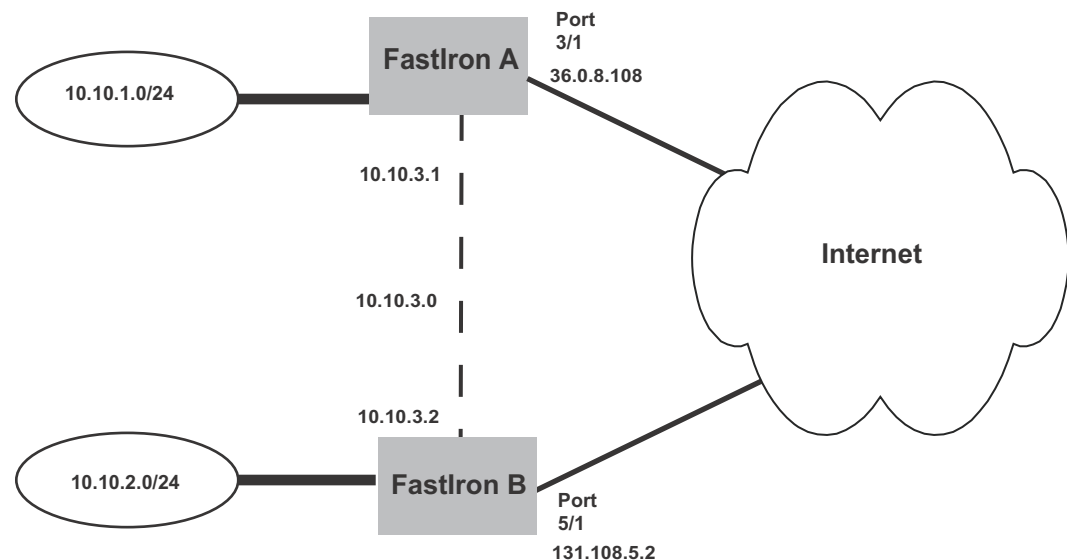
Syntax: [no] ip pim

Use the **no** form of the command to disable PIM-DM on the tunnel interface.

Example point-to-point GRE tunnel configuration

In the configuration example shown in Figure 125, a GRE Tunnel is configured between FastIron A and Brocade B. Traffic between networks 10.10.1.0/24 and 10.10.2.0/24 is encapsulated in a GRE packet sent through the tunnel on the 10.10.3.0 network, and unpacked and sent to the destination network. A static route is configured at each Layer 3 switch to go through the tunnel interface to the target network.

FIGURE 125 Point-to-point GRE tunnel configuration example



The following shows the configuration commands for the example shown in Figure 125.

NOTE

The configuration examples for FastIron A and FastIron B applies only to FastIron SX devices.

Configuring point-to-point GRE tunnel for FastIron A

```

Brocade (config)# interface ethernet 3/1
Brocade (config-if-e1000-3/1)# ip address 36.0.8.108/24
Brocade (config)# exit
Brocade (config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel source 36.0.8.108
Brocade(config-tnif-1)# tunnel destination 131.108.5.2
Brocade(config-tnif-1)# tunnel mode gre ip
Brocade(config-tnif-1)# tunnel loopback 4/1
Brocade(config-tnif-1)# ip address 10.10.3.1/24
Brocade(config-tnif-1)# exit
Brocade (config)# ip route 131.108.5.0/24 36.0.8.1
Brocade(config)# ip route 10.10.2.0/24 tunnel 1

```

Configuring point-to-point GRE tunnel for FastIron B

```

Brocade(config)# interface ethernet 5/1
Brocade(config-if-e1000-5/1)# ip address 131.108.5.2/24
Brocade(config)# exit
Brocade(config)# interface tunnel 1
Brocade(config-tnif-1)# tunnel source 131.108.5.2
Brocade(config-tnif-1)# tunnel destination 36.0.8.108
Brocade(config-tnif-1)# tunnel mode gre ip
Brocade(config-tnif-1)# tunnel loopback 1/1
Brocade(config-tnif-1)# ip address 10.10.3.2/24
Brocade(config-tnif-1)# exit
Brocade(config)# ip route 36.0.8.0/24 131.108.5.1
Brocade(config)# ip route 10.10.1.0/24 tunnel

```

Displaying GRE tunneling information

This section describes the **show** commands that display the GRE tunnels configuration, the link status of the GRE tunnels, and the routes that use GRE tunnels. To display information about multicast protocols and GRE tunnels, refer to [“Displaying multicast protocols and GRE tunneling information”](#) on page 1062.

To display GRE tunneling Information, use the following commands:

- **show ip interface**
- **show ip route**
- **show ip interface tunnel**
- **show ip tunnel traffic**
- **show interface tunnel**
- **show statistics tunnel**

The following shows an example output of the **show ip interface** command, which includes information about GRE tunnels.

```

Brocade# show ip interface

```

Interface	IP-Address	OK?	Method	Status	Protocol
Tunnel 1	10.10.3.1	YES	NVRAM	up	up

For field definitions, refer to [Table 183](#) on page 1069.

Syntax: show ip interface

The **show ip route** command displays routes that are pointing to a GRE tunnel as shown in the following example.

```
Brocade# show ip route
Total number of IP routes: 3, avail: 79996 (out of max 80000)
B:BGP D:Connected R:RIP S:Static O:OSPF *:Candidate default
      Destination      NetMask      Gateway      Port      Cost      Type
1      7.1.1.0          255.255.255.0  0.0.0.0      7          1          D
2      7.1.2.0          255.255.255.0  7.1.1.3      7          1          S
3      10.34.3.0         255.255.255.0  0.0.0.0      tn3         1          D
```

For field definitions, refer to [Table 187](#) on page 1076.

Syntax: show ip route

The **show ip interface tunnel** command displays the link status and IP address configuration for an IP tunnel interface as shown in the following example.

```
Brocade# show ip interface tunnel 1
Interface Tunnel 1
  port state: UP
  ip address: 21.21.21.2      subnet mask: 255.255.255.0
  encapsulation: GRE, mtu: 1476, metric: 1
  directed-broadcast-forwarding: disabled
  proxy-arp: disabled
  ip arp-age: 10 minutes
  No Helper Addresses are configured
  No inbound ip access-list is set
  No outgoing ip access-list is set
```

Syntax: show ip interface tunnel [<tunnel-ID>]

The **<tunnel-ID>** variable is a valid tunnel number or name.

The **show interface tunnel** command displays the GRE tunnel configuration.

```
Brocade# show int tunnel 3
Tunnel3 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source 7.1.1.1
  Tunnel destination is 7.1.2.3
  Tunnel loopback is not configured
  Port name is customer1001
  Internet address is 10.34.3.2/24, MTU 1476 bytes, encapsulation GRE
  Keepalive is Enabled : Interval 10, No.of Retries 3
  Total Keepalive Pkts Tx: 2, Rx: 2
  Path MTU Discovery: Enabled, MTU is 1476 bytes
```

Syntax: show interface tunnel [<tunnel-ID>]

This display shows the following information.

TABLE 180 CLI display of **show interface tunnel** command

Field	Definition
Hardware is Tunnel	The interface is a tunnel interface.
Tunnel source	The source address for the tunnel.
Tunnel destination	The destination address for the tunnel.

TABLE 180 CLI display of **show interface tunnel** command (Continued)

Field	Definition
Tunnel loopback	The tunnel loopback port for the tunnel (if applicable).
Port name	The port name (if applicable).
Internet address	The internet address.
MTU	The maximum transmission unit.
encapsulation GRE	GRE encapsulation is enabled on the port.
Keepalive	Indicates whether or not GRE link keepalive is enabled.
Interval	If GRE link keepalive is enabled, this field displays the number of seconds between each initiation of a GRE link keepalive message.
No. of Retries	If GRE link keepalive is enabled, this field displays the number of times that a GRE link keepalive packet is sent before the tunnel is placed in the DOWN state.
Total Keepalive Pkts	If GRE link keepalive is enabled, this field shows the total number of GRE link keepalive packets transmitted (TX) and received (RX) on the tunnel since it was last cleared by the administrator.
Path MTU Discovery	Indicates whether or not PMTUD is enabled. If PMTUD is enabled, the MTU value is also displayed.

The **show ip tunnel traffic** command displays the link status of the tunnel and the number of keepalive packets received and sent on the tunnel.

```
Brocade# show ip tunnel traffic
IP GRE Tunnels
      Tunnel Status  Packet Received  Packet Sent  KA recv  KA sent
  1   up/up         362             0           362      362
  3   up/up         0               0           0        0
 10  down/down      0               0           0        0
```

Syntax: show ip tunnel traffic

The **show statistics tunnel** [*<tunnel-ID>*] command displays GRE tunnel statistics for a specific tunnel ID number. The following shows an example output for tunnel ID 1.

```
Brocade(config-tnif-10)#show statistics tunnel 1
IP GRE Tunnels
      Tunnel Status  Packet Received  Packet Sent  KA recv  KA sent
  1   up/up         87120          43943      43208    43855
```

Syntax: show statistics tunnel [*<tunnel-ID>*]

The *<tunnel-ID>* variable specifies the tunnel ID number.

This display shows the following information.

TABLE 181 CLI display of **show ip tunnel traffic** command

Field	Description
Tunnel Status	Indicates whether the tunnel is up or down. Possible values are: <ul style="list-style-type: none"> • Up/Up – The tunnel and line protocol are up. • Up/Down – The tunnel is up and the line protocol is down. • Down/Up – The tunnel is down and the line protocol is up. • Down/Down – The tunnel and line protocol are down.
Packet Received	The number of packets received on the tunnel since it was last cleared by the administrator.
Packet Sent	The number of packets sent on the tunnel since it was last cleared by the administrator.
KA rcv	The number of keepalive packets received on the tunnel since it was last cleared by the administrator.
KA sent	The number of keepalive packets sent on the tunnel since it was last cleared by the administrator.

Displaying multicast protocols and GRE tunneling information

The following **show** commands display information about multicast protocols and GRE tunnels:

- **show ip pim interface**
- **show ip pim nbr**
- **show ip pim mcache**
- **show ip pim flow**
- **show statistics**
- **show ip mtu**

NOTE

All other **show** commands that are supported currently for Ethernet, VE, and IP loopback interfaces, are also supported for tunnel interfaces. To display information for a tunnel interface, specify the tunnel in the format **tn <num>**. For example, **show interface tn 1**. In some cases, the Ethernet port that the tunnel is using will be displayed in the format **tn<num>:e<port>**.

The following shows an example output of the **show ip pim interface** command. The lines in bold highlight the GRE tunnel-specific information.

```
Brocade# show ip pim interface
Interface e1
PIM Dense: V2
TTL Threshold: 1, Enabled, DR: itself
Local Address: 10.10.10.10

Interface tn1
PIM Dense: V2
TTL Threshold: 1, Enabled, DR: 1.1.1.20 on tn1:e2
Local Address: 1.1.1.10
Neighbor:
    1.1.1.20
```

Syntax: **show ip pim interface**

The following shows an example output of the **show ip pim nbr** command. The line in bold shows the GRE tunnel-specific information.

```
Brocade# show ip pim nbr
Total number of neighbors: 1 on 1 ports
Port   Phy_p      Neighbor      Holdtime Age    UpTime
tnl     tnl:e2      1.1.1.20      180      60     1740
```

Syntax: show ip pim nbr

The following shows an example output of the **show ip pim mcache** command. The line in bold shows the GRE tunnel-specific information.

```
Brocade# show ip pim mcache 230.1.1.1
1      (10.10.10.1 230.1.1.1) in e1 (e1), cnt=629
      Source is directly connected
      L3 (HW) 1: tnl:e2(VL1)
      fast=1 slow=0 pru=1 graft
      age=120s up-time=8m HW=1 L2-vidx=8191 has mll
```

Syntax: show ip pim mcache <ip-address>

The following shows an example output of the **show ip pim flow** command. The text in bold highlights the GRE tunnel-specific information.

```
Brocade# show ip pim flow 230.1.1.1

Multicast flow (10.10.10.1 230.1.1.1):
  Vidx for source vlan forwarding: 8191 (Blackhole, no L2 clients)
  Hardware MC Entry hit on devices: 0 1 2 3
  MC Entry[0x0c008040]: 00014001 000022ee 0ffc0001 00000000
  --- MLL contents read from Device 0 ---
  MLL Data[0x018c0010]: 0021ff8d 00000083 00000000 00000000
  First   : Last:1, outlif:60043fff1 00000000, TNL:1(e2)

1 flow printed
```

Syntax: show ip pim flow

The following shows an example output of the **show statistics** command. The following statistics demonstrate an example where the encapsulated multicast traffic ingresses a tunnel endpoint on port e 2, egresses and re-ingresses as native multicast traffic on the loopback port e 4, and is then forwarded to the outbound interface e 1.

```
Brocade# show statistics
```

Port	In Packets	Out Packets	In Errors	Out Errors
1	0	1670	0	0
2	1668	7	0	0
3	0	0	0	0
4	1668	1668	0	0

Syntax: show statistics

The **show ip mtu** command can be used to see if there is space available for the `ip_default_mtu_24` value in the system, or if the MTU value is already configured in the IP table. The following shows an example output of the **show ip mtu** command.

```
Brocade(config-tnif-10)#show ip mtu
idx   size  usage  ref-count
0     10218    1    default
1       800    0         1
2       900    0         1
3       750    0         1
4     10194    1         1
5     10198    0         1
```

Syntax: `show ip mtu`

Clearing GRE statistics

Use the **clear ip tunnel** command to clear statistics related to GRE tunnels.

To clear GRE tunnel statistics, enter a command such as the following.

```
Brocade(config)# clear ip tunnel stat 3
```

To reset a dynamically-configured MTU on a tunnel Interface back to the configured value, enter a command such as the following.

```
Brocade(config)#clear ip tunnel pmtud 3
```

Syntax: `clear ip tunnel [pmtud < tunnel-ID> | stat < tunnel-ID>]`

Use the **pmtud** option to reset a dynamically-configured MTU on a tunnel Interface back to the configured value.

Use the **stat** option to clear tunnel statistics.

The `<tunnel-ID>` variable is a valid tunnel number or name.

Use the **clear statistics tunnel** [`<tunnel-ID>`] command to clear GRE tunnel statistics for a specific tunnel ID number. To clear GRE tunnel statistics for tunnel ID 3, enter a command such as the following.

```
Brocade(config)# clear statistics tunnel 3
```

Syntax: `clear statistics tunnel [<tunnel-ID>]`

The `<tunnel-ID>` variable specifies the tunnel ID number.

Displaying IP configuration information and statistics

The following sections describe IP display options for Layer 3 Switches and Layer 2 Switches:

- To display IP information on a Layer 3 Switch, refer to [“Displaying IP information – Layer 3 Switches”](#) on page 1065.
- To display IP information on a Layer 2 Switch, refer to [“Displaying IP information – Layer 2 Switches”](#) on page 1080.

Changing the network mask display to prefix format

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the displays to prefix format (example: /18) on a Layer 3 Switch or Layer 2 Switch using the following CLI method.

NOTE

This option does not affect how information is displayed in the Web Management Interface.

To enable CIDR format for displaying network masks, entering the following command at the global CONFIG level of the CLI.

```
Brocade(config)# ip show-subnet-length
```

Syntax: [no] ip show-subnet-length

Displaying IP information – Layer 3 Switches

You can display the following IP configuration information statistics on Layer 3 Switches:

- Global IP parameter settings and IP access policies – refer to [“Displaying global IP configuration information”](#) on page 1066.
- CPU utilization statistics – refer to [“Displaying CPU utilization statistics”](#) on page 1067.
- IP interfaces – refer to [“Displaying IP interface information”](#) on page 1069.
- ARP entries – refer to [“Displaying ARP entries”](#) on page 1070.
- Static ARP entries – refer to [“Displaying ARP entries”](#) on page 1070.
- IP forwarding cache – refer to [“Displaying the forwarding cache”](#) on page 1073.
- IP route table – refer to [“Displaying the IP route table”](#) on page 1074.
- IP traffic statistics – refer to [“Displaying IP traffic statistics”](#) on page 1077.

The following sections describe how to display this information.

In addition to the information described below, you can display the following IP information. This information is described in other parts of this guide:

- RIP
- OSPF
- BGP4
- PIM
- DVMRP
- VRRP or VRRP-E

Displaying global IP configuration information

To display IP configuration information, enter the following command at any CLI level.

```
Brocade# show ip
```

Global Settings

```
ttl: 64, arp-age: 10, bootp-relay-max-hops: 4
router-id : 207.95.11.128
enabled : UDP-Broadcast-Forwarding IRDP Proxy-ARP RARP OSPF
disabled: BGP4 Load-Sharing RIP DVMRP FSRP VRRP
```

Static Routes

Index	IP Address	Subnet Mask	Next Hop Router	Metric	Distance
1	0.0.0.0	0.0.0.0	209.157.23.2	1	1

Policies

Index	Action	Source	Destination	Protocol	Port	Operator
1	deny	209.157.22.34	209.157.22.26	tcp	http	=
64	permit	any	any			

Syntax: show ip

NOTE

This command has additional options, which are explained in other sections in this guide, including the sections following this one.

This display shows the following information.

TABLE 182 CLI display of global IP configuration information – Layer 3 Switch

Field	Description
Global settings	
ttl	The Time-To-Live (TTL) for IP packets. The TTL specifies the maximum number of router hops a packet can travel before reaching the Brocade router. If the packet TTL value is higher than the value specified in this field, the Brocade router drops the packet. To change the maximum TTL, refer to “Changing the TTL threshold” on page 987.
arp-age	The ARP aging period. This parameter specifies how many minutes an inactive ARP entry remains in the ARP cache before the router ages out the entry. To change the ARP aging period, refer to “Changing the ARP aging period” on page 983.
bootp-relay-max-hops	The maximum number of hops away a BootP server can be located from the Brocade router and still be used by the router clients for network booting. To change this value, refer to “Changing the maximum number of hops to a BootP relay server” on page 1013.
router-id	The 32-bit number that uniquely identifies the Brocade router. By default, the router ID is the numerically lowest IP interface configured on the router. To change the router ID, refer to “Changing the router ID” on page 976.
enabled	The IP-related protocols that are enabled on the router.
disabled	The IP-related protocols that are disabled on the router.
Static routes	
Index	The row number of this entry in the IP route table.
IP Address	The IP address of the route destination.
Subnet Mask	The network mask for the IP address.

TABLE 182 CLI display of global IP configuration information – Layer 3 Switch (Continued)

Field	Description
Next Hop Router	The IP address of the router interface to which the Brocade router sends packets for the route.
Metric	The cost of the route. Usually, the metric represents the number of hops to the destination.
Distance	The administrative distance of the route. The default administrative distance for static IP routes in Brocade routers is 1. To list the default administrative distances for all types of routes or to change the administrative distance of a static route, refer to “Changing the administrative distance” on page 1204.
Policies	
Index	The policy number. This is the number you assigned the policy when you configured it.
Action	The action the router takes if a packet matches the comparison values in the policy. The action can be one of the following: <ul style="list-style-type: none"> deny – The router drops packets that match this policy. permit – The router forwards packets that match this policy.
Source	The source IP address the policy matches.
Destination	The destination IP address the policy matches.
Protocol	The IP protocol the policy matches. The protocol can be one of the following: <ul style="list-style-type: none"> ICMP IGMP IGRP OSPF TCP UDP
Port	The Layer 4 TCP or UDP port the policy checks for in packets. The port can be displayed by its number or, for port types the router recognizes, by the well-known name. For example, TCP port 80 can be displayed as HTTP. NOTE: This field applies only if the IP protocol is TCP or UDP.
Operator	The comparison operator for TCP or UDP port names or numbers. NOTE: This field applies only if the IP protocol is TCP or UDP.

Displaying CPU utilization statistics

You can display CPU utilization statistics for IP protocols using the **show process cpu** command.

The **show process cpu** command includes CPU utilization statistics for ACL, 802.1x, and L2VLAN. L2VLAN contains any packet transmitted to a VLAN by the CPU, including unknown unicast, multicast, broadcast, and CPU forwarded Layer 2 traffic.

To display CPU utilization statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI.

Displaying IP configuration information and statistics

```
Brocade# show process cpu
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ACL             0.00       0.00       0.00       0.00        0
ARP             0.01       0.01       0.01       0.01       714
BGP             0.00       0.00       0.00       0.00        0
DOT1X          0.00       0.00       0.00       0.00        0
GVRP           0.00       0.00       0.00       0.00        0
ICMP           0.00       0.00       0.00       0.00       161
IP             0.00       0.00       0.00       0.00       229
L2VLAN         0.01       0.00       0.00       0.01       673
OSPF           0.00       0.00       0.00       0.00        0
RIP            0.00       0.00       0.00       0.00        9
STP            0.00       0.00       0.00       0.00        7
VRRP           0.00       0.00       0.00       0.00        0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example.

```
Brocade# show process cpu
The system has only been up for 6 seconds.
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ACL             0.00       0.00       0.00       0.00        0
ARP             0.01       0.01       0.01       0.01       714
BGP             0.00       0.00       0.00       0.00        0
DOT1X          0.00       0.00       0.00       0.00        0
GVRP           0.00       0.00       0.00       0.00        0
ICMP           0.00       0.00       0.00       0.00       161
IP             0.00       0.00       0.00       0.00       229
L2VLAN         0.01       0.00       0.00       0.01       673
OSPF           0.00       0.00       0.00       0.00        0
RIP            0.00       0.00       0.00       0.00        9
STP            0.00       0.00       0.00       0.00        7
VRRP           0.00       0.00       0.00       0.00        0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following.

```
Brocade# show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name    Sec(%)    Time(ms)
ACL             0         0.00
ARP             1         0.01
BGP             0         0.00
DOT1X           0         0.00
GVRP            0         0.00
ICMP            0         0.00
IP              0         0.00
L2VLAN          1         0.01
OSPF            0         0.00
RIP             0         0.00
STP             0         0.00
VRRP            0         0.00
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: `show process cpu [<num>]`

The `<num>` parameter specifies the number of seconds and can be from 1 through 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying IP interface information

To display IP interface information, enter the following command at any CLI level.

```
Brocade# show ip interface
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet 1/1	207.95.6.173	YES	NVRAM	up	up
Ethernet 1/2	3.3.3.3	YES	manual	up	up
Loopback 1	1.2.3.4	YES	NVRAM	down	down

Syntax: `show ip interface [ethernet [<slotnum>/<portnum>] | [loopback <num>] | [ve <num>]`

This display shows the following information.

TABLE 183 CLI display of interface IP configuration information

Field	Description
Interface	The type and the slot and port number of the interface.
IP-Address	The IP address of the interface. NOTE: If an "s" is listed following the address, this is a secondary address. When the address was configured, the interface already had an IP address in the same subnet, so the software required the "secondary" option before the software could add the interface.
OK?	Whether the IP address has been configured on the interface.
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI or Web Management Interface, but have not saved the configuration, the entry for the interface in the Method field is "manual".
Status	The link status of the interface. If you have disabled the interface with the disable command, the entry in the Status field will be "administratively down". Otherwise, the entry in the Status field will be either "up" or "down".
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the protocol field will be "up". Otherwise the entry in the protocol field will be "down".

To display detailed IP information for a specific interface, enter a command such as the following.

```
Brocade# show ip interface ethernet 1/1
Interface Ethernet 1/1
  port state: UP
  ip address: 192.168.9.51      subnet mask: 255.255.255.0
  encapsulation: ETHERNET, mtu: 1500, metric: 1
  directed-broadcast-forwarding: disabled
  proxy-arp: disabled
  ip arp-age: 10 minutes
  Ip Flow switching is disabled
  No Helper Addresses are configured.
  No inbound ip access-list is set
  No outgoing ip access-list is set
```

Displaying ARP entries

You can display the ARP cache and the static ARP table. The ARP cache contains entries for devices attached to the Layer 3 Switch. The static ARP table contains the user-configured ARP entries. An entry in the static ARP table enters the ARP cache when the entry interface comes up.

The tables require separate display commands or Web management options.

Displaying the ARP cache

To display the contents of the ARP cache, enter the following command at any CLI level.

```
Brocade# show arp

Total number of ARP entries: 5, maximum capacity: 6000


| No. | IP Address   | MAC Address    | Type    | Age | Port | Status |
|-----|--------------|----------------|---------|-----|------|--------|
| 1   | 207.95.6.102 | 0800.5afc.ea21 | Dynamic | 0   | 6    | Valid  |
| 2   | 207.95.6.18  | 00a0.24d2.04ed | Dynamic | 3   | 6    | Pend   |
| 3   | 207.95.6.54  | 00a0.24ab.cd2b | Dynamic | 0   | 6    | Pend   |
| 4   | 207.95.6.101 | 0800.207c.a7fa | Dynamic | 0   | 6    | Valid  |
| 5   | 207.95.6.211 | 00c0.2638.ac9c | Dynamic | 0   | 6    | Valid  |


```

Syntax: `show arp [ethernet [<slotnum>]/<portnum> | mac-address <xxxx.xxxx.xxxx> [<mask>] | <ip-addr> [<ip-mask>]] [<num>]`

The `<slotnum>` parameter is required on chassis devices.

The `<portnum>` parameter lets you restrict the display to entries for a specific port.

The `mac-address <xxxx.xxxx.xxxx>` parameter lets you restrict the display to entries for a specific MAC address.

The `<mask>` parameter lets you specify a mask for the `mac-address <xxxx.xxxx.xxxx>` parameter, to display entries for multiple MAC addresses. Specify the MAC address mask as “f”s and “0”s, where “f”s are significant bits.

The `<ip-addr>` and `<ip-mask>` parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

NOTE

The `<ip-mask>` parameter and `<mask>` parameter perform different operations. The `<ip-mask>` parameter specifies the network mask for a specific IP address, whereas the `<mask>` parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

The `<num>` parameter lets you display the table beginning with a specific entry number.

NOTE

The entry numbers in the ARP cache are not related to the entry numbers for static ARP table entries.

This display shows the following information. The number in the left column of the CLI display is the row number of the entry in the ARP cache. This number is not related to the number you assign to static MAC entries in the static ARP table.

TABLE 184 CLI display of ARP cache

Field	Description
Total number of ARP Entries	The number of entries in the ARP cache.
Maximum capacity	The total number of ARP entries supported on the device.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Type	<p>The ARP entry type, which can be one of the following:</p> <ul style="list-style-type: none"> • Dynamic – The Layer 3 Switch learned the entry from an incoming packet. • Static – The Layer 3 Switch loaded the entry from the static ARP table when the device for the entry was connected to the Layer 3 Switch. • DHCP – The Layer 3 Switch learned the entry from the DHCP binding address table. <p>NOTE: If the type is DHCP, the port number will not be available until the entry gets resolved through ARP.</p>
Age	<p>The number of minutes before which the ARP entry was refreshed. If this value reaches the ARP aging period, the entry is removed from the table.</p> <p>To display the ARP aging period, refer to “Displaying global IP configuration information” on page 1066. To change the ARP aging interval, refer to “Changing the ARP aging period” on page 983.</p> <p>NOTE: Static entries do not age out.</p>
Port	<p>The port on which the entry was learned.</p> <p>NOTE: If the ARP entry type is DHCP, the port number will not be available until the entry gets resolved through ARP.</p>
Status	<p>The status of the entry, which can be one of the following:</p> <ul style="list-style-type: none"> • Valid – This a valid ARP entry. • Pend – The ARP entry is not yet resolved.

Displaying the static ARP table

To display the static ARP table instead of the ARP cache, enter the following command at any CLI level.

```
Brocade# show ip static-arp
```

```
Static ARP table size: 512, configurable from 512 to 1024
  Index  IP Address      MAC Address      Port
  1      207.95.6.111     0800.093b.d210   1/1
  3      207.95.6.123     0800.093b.d211   1/1
```

This example shows two static entries. Note that because you specify an entry index number when you create the entry, it is possible for the range of index numbers to have gaps, as shown in this example.

NOTE

The entry number you assign to a static ARP entry is not related to the entry numbers in the ARP cache.

Syntax: `show ip static-arp [ethernet [<slotnum>/]<portnum> | mac-address <xxxx.xxxx.xxxx> [<mask>] | <ip-addr> [<ip-mask>]] [<num>]`

The `<slotnum>` parameter is required on chassis devices.

The `<portnum>` parameter lets you restrict the display to entries for a specific port.

The `mac-address <xxxx.xxxx.xxxx>` parameter lets you restrict the display to entries for a specific MAC address.

The `<mask>` parameter lets you specify a mask for the `mac-address <xxxx.xxxx.xxxx>` parameter, to display entries for multiple MAC addresses. Specify the MAC address mask as “f”s and “0”s, where “f”s are significant bits.

The `<ip-addr>` and `<ip-mask>` parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

NOTE

The `<ip-mask>` parameter and `<mask>` parameter perform different operations. The `<ip-mask>` parameter specifies the network mask for a specific IP address, whereas the `<mask>` parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

The `<num>` parameter lets you display the table beginning with a specific entry number.

TABLE 185 CLI display of static ARP table

Field	Description
Static ARP table size	The maximum number of static entries that can be configured on the device using the current memory allocation. The range of valid memory allocations for static ARP entries is listed after the current allocation. To change the memory allocation for static ARP entries, refer to “Changing the maximum number of entries the static ARP table can hold” on page 986.
Index	The number of this entry in the table. You specify the entry number when you create the entry.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Port	The port attached to the device the entry is for.

Displaying the forwarding cache

To display the IP forwarding cache, enter the following command at any CLI level.

```
Brocade# show ip cache
```

```
Total number of cache entries: 3
D:Dynamic P:Permanent F:Forward U:Us C:Complex Filter
W:Wait ARP I:ICMP Deny K:Drop R:Fragment S:Snap Encap
      IP Address      Next Hop      MAC      Type  Port  Vlan  Pri
1      192.168.1.11    DIRECT      0000.0000.0000  PU    n/a    0
2      192.168.1.255    DIRECT      0000.0000.0000  PU    n/a    0
3      255.255.255.255  DIRECT      0000.0000.0000  PU    n/a    0
```

Syntax: `show ip cache [<ip-addr>] | [<num>]`

The `<ip-addr>` parameter displays the cache entry for the specified IP address.

The `<num>` parameter displays the cache beginning with the row following the number you enter. For example, to begin displaying the cache at row 10, enter the following command.

show ip cache 9

The **show ip cache** command displays the following information.

TABLE 186 CLI display of IP forwarding cache – Layer 3 Switch

Field	Description
IP Address	The IP address of the destination.
Next Hop	The IP address of the next-hop router to the destination. This field contains either an IP address or the value DIRECT. DIRECT means the destination is either directly attached or the destination is an address on this Brocade device. For example, the next hop for loopback addresses and broadcast addresses is shown as DIRECT.
MAC	The MAC address of the destination. NOTE: If the entry is type U (indicating that the destination is this Brocade device), the address consists of zeroes.

TABLE 186 CLI display of IP forwarding cache – Layer 3 Switch (Continued)

Field	Description
Type	The type of host entry, which can be one or more of the following: <ul style="list-style-type: none"> • D – Dynamic • P – Permanent • F – Forward • U – Us • C – Complex Filter • W – Wait ARP • I – ICMP Deny • K – Drop • R – Fragment • S – Snap Encap
Port	The port through which this device reaches the destination. For destinations that are located on this device, the port number is shown as “n/a”.
VLAN	Indicates the VLANs the listed port is in.
Pri	The QoS priority of the port or VLAN.

Displaying the IP route table

To display the IP route table, enter the **show ip route** command at any CLI level.

```

Brocade# show ip route
Total number of IP routes: 514
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
Destination      NetMask          Gateway          Port    Cost    Type
1.1.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.2.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.3.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.4.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.5.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.6.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.7.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.8.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.9.0.0           255.255.0.0      99.1.1.2        1/1     2       R
1.10.0.0          255.255.0.0      99.1.1.2        1/1     2       S

```

Syntax: **show ip route** [*<ip-addr>*] [*<ip-mask>*] [**longer**] [**none-bgp**] | *<num>* | **bgp** | **direct** | **ospf** | **rip** | **static**

The *<ip-addr>* parameter displays the route to the specified IP address.

The *<ip-mask>* parameter lets you specify a network mask or, if you prefer CIDR format, the number of bits in the network mask. If you use CIDR format, enter a forward slash immediately after the IP address, then enter the number of mask bits (for example: 209.157.22.0/24 for 209.157.22.0 255.255.255.0).

The **longer** parameter applies only when you specify an IP address and mask. This option displays only the routes for the specified IP address and mask. Refer to the following example.

The **none-bgp** parameter displays only the routes that did not come from BGP4.

The *<num>* option display the route table entry whose row number corresponds to the number you specify. For example, if you want to display the tenth row in the table, enter “10”.

The **bgp** option displays the BGP4 routes.

The **direct** option displays only the IP routes that are directly attached to the Layer 3 Switch.

The **ospf** option displays the OSPF routes.

The **rip** option displays the RIP routes.

The **static** option displays only the static IP routes.

The **default** routes are displayed first.

Here is an example of how to use the **direct** option. To display only the IP routes that go to devices directly attached to the Layer 3 Switch, enter the following command.

```
Brocade# show ip route direct
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
      Destination      NetMask      Gateway      Port    Cost    Type
      209.157.22.0      255.255.255.0  0.0.0.0      4/11    1        D
```

Notice that the route displayed in this example has “D” in the Type field, indicating the route is to a directly connected device.

Here is an example of how to use the **static** option. To display only the static IP routes, enter the following command.

```
Brocade# show ip route static
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
      Destination      NetMask      Gateway      Port    Cost    Type
      192.144.33.11     255.255.255.0  209.157.22.12  1/1     2        S
```

Notice that the route displayed in this example has “S” in the Type field, indicating the route is static.

Here is an example of how to use the **longer** option. To display only the routes for a specified IP address and mask, enter a command such as the following.

```
Brocade# show ip route 209.159.0.0/16 longer
Starting index: 1 B:BGP D:Directly-Connected R:RIP S:Static O:OSPF
Destination NetMask Gateway Port Cost Type

52 209.159.38.0 255.255.255.0 207.95.6.101 1/1 1 S
53 209.159.39.0 255.255.255.0 207.95.6.101 1/1 1 S
54 209.159.40.0 255.255.255.0 207.95.6.101 1/1 1 S
55 209.159.41.0 255.255.255.0 207.95.6.101 1/1 1 S
56 209.159.42.0 255.255.255.0 207.95.6.101 1/1 1 S
57 209.159.43.0 255.255.255.0 207.95.6.101 1/1 1 S
58 209.159.44.0 255.255.255.0 207.95.6.101 1/1 1 S
59 209.159.45.0 255.255.255.0 207.95.6.101 1/1 1 S
60 209.159.46.0 255.255.255.0 207.95.6.101 1/1 1 S
```

This example shows all the routes for networks beginning with 209.159. The mask value and **longer** parameter specify the range of network addresses to be displayed. In this example, all routes within the range 209.159.0.0 – 209.159.255.255 are listed.

The **summary** option displays a summary of the information in the IP route table. The following is an example of the output from this command.

Example

```
Brocade# show ip route summary

IP Routing Table - 35 entries:
  6 connected, 28 static, 0 RIP, 1 OSPF, 0 BGP, 0 ISIS, 0 MPLS
Number of prefixes:
 /0: 1 /16: 27 /22: 1 /24: 5 /32: 1
```

Syntax: show ip route summary

In this example, the IP route table contains 35 entries. Of these entries, 6 are directly connected devices, 28 are static routes, and 1 route was calculated through OSPF. One of the routes has a zero-bit mask (this is the default route), 27 have a 22-bit mask, 5 have a 24-bit mask, and 1 has a 32-bit mask.

The following table lists the information displayed by the **show ip route** command.

TABLE 187 CLI display of IP route table

Field	Description
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router. An asterisk (*) next to the next-hop router indicates that it is one of multiple Equal-Cost Multi-Path (ECMP) next hops for a given route. The asterisk will initially appear next to the first next hop for each route with multiple ECMP next hops. If the ARP entry for the <i>next hop</i> * ages out or is cleared, then the next packet to be routed through the Brocade device whose destination matches that route can cause the asterisk to move to the next hop down the list of ECMP next hops for that route. This means that if the <i>next hop</i> * goes down, the asterisk can move to another next hop with equal cost.
Port	The port through which this router sends packets to reach the route's destination.
Cost	The route's cost.
Type	The route type, which can be one of the following: <ul style="list-style-type: none"> • B – The route was learned from BGP. • D – The destination is directly connected to this Layer 3 Switch. • R – The route was learned from RIP. • S – The route is a static route. • * – The route and next-hop gateway are resolved through the ip default-network setting. • O – The route is an OSPF route. Unless you use the ospf option to display the route table, "O" is used for all OSPF routes. If you do use the ospf option, the following type codes are used: <ul style="list-style-type: none"> • O – OSPF intra area route (within the same area). • IA – The route is an OSPF inter area route (a route that passes from one area into another). • E1 – The route is an OSPF external type 1 route. • E2 – The route is an OSPF external type 2 route.

Clearing IP routes

If needed, you can clear the entire route table or specific individual routes.

To clear all routes from the IP route table, enter the following command.

```
Brocade# clear ip route
```

To clear route 209.157.22.0/24 from the IP routing table, enter the **clear ip route** command.

```
Brocade# clear ip route 209.157.22.0/24
```

Syntax: **clear ip route** [*<ip-addr>* *<ip-mask>*]

or

Syntax: **clear ip route** [*<ip-addr>/<mask-bits>*]

Displaying IP traffic statistics

To display IP traffic statistics, enter the **show ip traffic** command at any CLI level.

```
Brocade# show ip traffic
```

```
IP Statistics
```

```
  139 received, 145 sent, 0 forwarded
  0 filtered, 0 fragmented, 0 reassembled, 0 bad header
  0 no route, 0 unknown proto, 0 no buffer, 0 other errors
```

```
ICMP Statistics
```

```
Received:
```

```
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation
```

```
Sent:
```

```
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation
```

```
UDP Statistics
```

```
  1 received, 0 sent, 1 no port, 0 input errors
```

```
TCP Statistics
```

```
  0 active opens, 0 passive opens, 0 failed attempts
  0 active resets, 0 passive resets, 0 input errors
  138 in segments, 141 out segments, 4 retransmission
```

```
RIP Statistics
```

```
  0 requests sent, 0 requests received
  0 responses sent, 0 responses received
  0 unrecognized, 0 bad version, 0 bad addr family, 0 bad req format
  0 bad metrics, 0 bad resp format, 0 resp not from rip port
  0 resp from loopback, 0 packets rejected
```

The **show ip traffic** command displays the following information.

TABLE 188 CLI display of IP traffic statistics – Layer 3 Switch

Field	Description
IP statistics	
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
forwarded	The total number of IP packets received by the device and forwarded to other devices.
filtered	The total number of IP packets filtered by the device.
fragmented	The total number of IP packets fragmented by this device to accommodate the MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.
no buffer	This information is used by Brocade customer support.
other errors	The number of packets dropped due to error types other than those listed above.
ICMP statistics	
The ICMP statistics are derived from RFC 792, "Internet Control Message Protocol", RFC 950, "Internet Standard Subnetting Procedure", and RFC 1256, "ICMP Router Discovery Messages". Statistics are organized into Sent and Received. The field descriptions below apply to each.	
total	The total number of ICMP messages sent or received by the device.
errors	This information is used by Brocade customer support.
unreachable	The number of Destination Unreachable messages sent or received by the device.
time exceed	The number of Time Exceeded messages sent or received by the device.
parameter	The number of Parameter Problem messages sent or received by the device.
source quench	The number of Source Quench messages sent or received by the device.
redirect	The number of Redirect messages sent or received by the device.
echo	The number of Echo messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
timestamp	The number of Timestamp messages sent or received by the device.
timestamp reply	The number of Timestamp Reply messages sent or received by the device.
addr mask	The number of Address Mask Request messages sent or received by the device.
addr mask reply	The number of Address Mask Replies messages sent or received by the device.
irdp advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages sent or received by the device.
irdp solicitation	The number of IRDP Solicitation messages sent or received by the device.
UDP statistics	

TABLE 188 CLI display of IP traffic statistics – Layer 3 Switch (Continued)

Field	Description
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because they did not have a valid UDP port number.
input errors	This information is used by Brocade customer support.
TCP statistics	
The TCP statistics are derived from RFC 793, "Transmission Control Protocol".	
active opens	The number of TCP connections opened by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by this device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Brocade customer support.
active resets	The number of TCP connections this device reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections this device reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Brocade customer support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that this device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.
RIP statistics	
The RIP statistics are derived from RFC 1058, "Routing Information Protocol".	
requests sent	The number of requests this device has sent to another RIP router for all or part of its RIP routing table.
requests received	The number of requests this device has received from another RIP router for all or part of this device RIP routing table.
responses sent	The number of responses this device has sent to another RIP router request for all or part of this device RIP routing table.
responses received	The number of responses this device has received to requests for all or part of another RIP router routing table.
unrecognized	This information is used by Brocade customer support.
bad version	The number of RIP packets dropped by the device because the RIP version was either invalid or is not supported by this device.
bad addr family	The number of RIP packets dropped because the value in the Address Family Identifier field of the packet header was invalid.
bad req format	The number of RIP request packets this router dropped because the format was bad.
bad metrics	This information is used by Brocade customer support.
bad resp format	The number of responses to RIP request packets dropped because the format was bad.
resp not from rip port	This information is used by Brocade customer support.

TABLE 188 CLI display of IP traffic statistics – Layer 3 Switch (Continued)

Field	Description
resp from loopback	The number of RIP responses received from loopback interfaces.
packets rejected	This information is used by Brocade customer support.

Displaying IP information – Layer 2 Switches

You can display the following IP configuration information statistics on Layer 2 Switches:

- Global IP settings – refer to [“Displaying global IP configuration information”](#) on page 1080.
- ARP entries – refer to [“Displaying ARP entries”](#) on page 1081.
- IP traffic statistics – refer to [“To display IP traffic statistics on a Layer 2 Switch, enter the show ip traffic command at any CLI level.”](#) on page 1082.

Displaying global IP configuration information

To display the Layer 2 Switch IP address and default gateway, enter the **show ip** command.

```
Brocade# show ip

Switch IP address: 192.168.1.2

Subnet mask: 255.255.255.0

Default router address: 192.168.1.1
TFTP server address: None
Configuration filename: None
Image filename: None
```

Syntax: show ip

This display shows the following information.

TABLE 189 CLI display of global IP configuration information – Layer 2 Switch

Field	Description
IP configuration	
Switch IP address	The management IP address configured on the Layer 2 Switch. Specify this address for Telnet or Web management access.
Subnet mask	The subnet mask for the management IP address.
Default router address	The address of the default gateway, if you specified one.
Most recent TFTP access	
TFTP server address	The IP address of the most-recently contacted TFTP server, if the switch has contacted a TFTP server since the last time the software was reloaded or the switch was rebooted.

TABLE 189 CLI display of global IP configuration information – Layer 2 Switch

Field	Description
Configuration filename	The name under which the Layer 2 Switch startup-config file was uploaded or downloaded during the most recent TFTP access.
Image filename	The name of the Layer 2 Switch flash image (system software file) that was uploaded or downloaded during the most recent TFTP access.

Displaying ARP entries

To display the entries the Layer 2 Switch has placed in its ARP cache, enter the **show arp** command from any level of the CLI.

```
Brocade# show arp
```

```
Total Arp Entries : 1, maximum capacity: 1000
No.
1      IP           Mac           Port Age VlanId
      192.168.1.170  0010.5a11.d042   7    0     1
```

Syntax: show arp

This display shows the following information.

TABLE 190 CLI display of ARP cache

Field	Description
Total ARP Entries	The number of entries in the ARP cache.
Maximum capacity	The total number of ARP entries supported on the device.
IP	The IP address of the device.
Mac	The MAC address of the device. NOTE: If the MAC address is all zeros, the entry is for the default gateway, but the Layer 2 Switch does not have a link to the gateway.
Port	The port on which the entry was learned.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the cache.
VlanId	The VLAN the port that learned the entry is in. NOTE: If the MAC address is all zeros, this field shows a random VLAN ID, since the Layer 2 Switch does not yet know which port the device for this entry is attached to.

Displaying IP traffic statistics

To display IP traffic statistics on a Layer 2 Switch, enter the **show ip traffic** command at any CLI level.

```
Brocade# show ip traffic

IP Statistics
  27 received, 24 sent
  0 fragmented, 0 reassembled, 0 bad header
  0 no route, 0 unknown proto, 0 no buffer, 0 other errors

ICMP Statistics
Received:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp rely, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation
Sent:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp rely, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation

UDP Statistics
  0 received, 0 sent, 0 no port, 0 input errors

TCP Statistics
  1 current active tcbs, 4 tcbs allocated, 0 tcbs freed 0 tcbs protected
  0 active opens, 0 passive opens, 0 failed attempts
  0 active resets, 0 passive resets, 0 input errors
  27 in segments, 24 out segments, 0 retransmission
```

Syntax: show ip traffic

The **show ip traffic** command displays the following information.

TABLE 191 CLI display of IP traffic statistics – Layer 2 Switch

Field	Description
IP statistics	
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
fragmented	The total number of IP packets fragmented by this device to accommodate the MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.
no buffer	This information is used by Brocade customer support.
other errors	The number of packets that this device dropped due to error types other than the types listed above.

TABLE 191 CLI display of IP traffic statistics – Layer 2 Switch (Continued)

Field	Description
ICMP statistics	
The ICMP statistics are derived from RFC 792, “Internet Control Message Protocol”, RFC 950, “Internet Standard Subnetting Procedure”, and RFC 1256, “ICMP Router Discovery Messages”. Statistics are organized into Sent and Received. The field descriptions below apply to each.	
total	The total number of ICMP messages sent or received by the device.
errors	This information is used by Brocade customer support.
unreachable	The number of Destination Unreachable messages sent or received by the device.
time exceed	The number of Time Exceeded messages sent or received by the device.
parameter	The number of Parameter Problem messages sent or received by the device.
source quench	The number of Source Quench messages sent or received by the device.
redirect	The number of Redirect messages sent or received by the device.
echo	The number of Echo messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
timestamp	The number of Timestamp messages sent or received by the device.
timestamp reply	The number of Timestamp Reply messages sent or received by the device.
addr mask	The number of Address Mask Request messages sent or received by the device.
addr mask reply	The number of Address Mask Replies messages sent or received by the device.
irdp advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages sent or received by the device.
irdp solicitation	The number of IRDP Solicitation messages sent or received by the device.
UDP statistics	
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Brocade customer support.
TCP statistics	
The TCP statistics are derived from RFC 793, “Transmission Control Protocol”.	
current active tcbs	The number of TCP Control Blocks (TCBs) that are currently active.
tcbs allocated	The number of TCBs that have been allocated.
tcbs freed	The number of TCBs that have been freed.
tcbs protected	This information is used by Brocade customer support.
active opens	The number of TCP connections opened by this device by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by this device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Brocade customer support.
active resets	The number of TCP connections this device reset by sending a TCP RESET message to the device at the other end of the connection.

TABLE 191 CLI display of IP traffic statistics – Layer 2 Switch (Continued)

Field	Description
passive resets	The number of TCP connections this device reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Brocade customer support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that this device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

Disabling IP checksum check

NOTE

This command is supported on FastIron X Series IPv4 devices only.

The **disable-hw-ip-checksum-check** command traps a packet with bad checksum to the CPU. Previously, if the packet processor detected a packet with, for example, the checksum 0xFFFF, it would treat it as a bad checksum even if it was correct and it would drop the packet. Now, the command **disable-hw-ip-checksum-check** traps the packet at the CPU and if the checksum is correct, it forwards the packet.

To set disable hardware ip checksum check for all ports, enter the following command.

```
Brocade# )# disable-hw-ip-checksum-check
disable-ip-header-check set for all ports
```

To clear disable hardware ip checksum check on all ports, enter the following command.

```
Brocade# )# no disable-hw-ip-checksum-check ethernet 13
disable-hw-ip-checksum-check cleared for ports the 13 to 24
```

To set disable hardware ip checksum check on for example, port range 0-12, enter the following command.

```
Brocade# ))# disable-hw-ip-checksum-check ethernet 2
disable-ip-header-check set for ports ethe 1 to 12
```

To set disable hardware ip checksum check on, for example, port range 13-24, enter the following command.

```
Brocade# ))# disable-hw-ip-checksum-check ethernet 22
disable-ip-header-check set for ports ethe 13 to 24
```

To clear disable hardware ip checksum check on, for example, port range 13-24, enter the following command.

```
Brocade# )# no disable-hw-ip-checksum-check ethernet 13  
disable-hw-ip-checksum-check cleared for ports the 13 to 24
```

NOTE

The port range could be any consecutive range, it may not necessarily be a decimal number.

Syntax: [no] **disable-hw-ip-checksum-check ethernet** <portnum>

NOTE

This command only functions on the IPv4 platform.

Disabling IP checksum check

Spanning Tree Protocol

[Table 192](#) lists the individual Brocade FastIron switches and the Spanning Tree Protocol (STP) features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 192 Supported STP features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
802.1s Multiple Spanning Tree	Yes	Yes	Yes	Yes	Yes
802.1W Rapid Spanning Tree (RSTP)	Yes	Yes	Yes	Yes	Yes
802.1D Spanning Tree Support	Yes	Yes	Yes	Yes	Yes
Enhanced IronSpan support includes Fast Port Span, Fast Uplink Span, and Single-instance Span	Yes	Yes	Yes	Yes	Yes
FastIron Layer 2 devices (switches) support up to 254 spanning tree instances for VLANs.	Yes	Yes	Yes	Yes	32 for ICX 6430
Layer 3 devices (routers) support up to 254 spanning tree instances for VLANs.	Yes	Yes	Yes	Yes	ICX 6450 only
PVST/PVST+ compatibility	Yes	Yes	Yes	Yes	Yes
PVRST+ compatibility	Yes	Yes	Yes	Yes	Yes
BPDU Guard	Yes	Yes	Yes	Yes	Yes
Root Guard	Yes	Yes	Yes	Yes	Yes
Error disable recovery	Yes	Yes	Yes	Yes	Yes

STP overview

The Spanning Tree Protocol (STP) eliminates Layer 2 loops in networks, by selectively blocking some ports and allowing other ports to forward traffic, based on global (bridge) and local (port) parameters you can configure.

STP related features, such as RSTP and PVST, extend the operation of standard STP, enabling you to fine-tune standard STP and avoid some of its limitations.

You can enable or disable STP on a global basis (for the entire device), a port-based VLAN basis (for the individual Layer 2 broadcast domain), or an individual port basis.

Configuration procedures are provided for the standard STP bridge and port parameters as well as Brocade features listed in [Table 198](#).

Standard STP parameter configuration

Brocade Layer 2 Switches and Layer 3 Switches support standard STP as described in the IEEE 802.1D specification. STP is enabled by default on Layer 2 Switches but disabled by default on Layer 3 Switches.

By default, each port-based VLAN on a Brocade device runs a separate spanning tree (a separate instance of STP). A Brocade device has one port-based VLAN (VLAN 1) by default that contains all the device ports. Thus, by default each Brocade device has one spanning tree. However, if you configure additional port-based VLANs on a Brocade device, then each of those VLANs on which STP is enabled and VLAN 1 all run separate spanning trees.

If you configure a port-based VLAN on the device, the VLAN has the same STP state as the default STP state on the device. Thus, on Layer 2 Switches, new VLANs have STP enabled by default. On Layer 3 Switches, new VLANs have STP disabled by default. You can enable or disable STP in each VLAN separately. In addition, you can enable or disable STP on individual ports.

STP parameters and defaults

[Table 193](#) lists the default STP states for Brocade devices.

TABLE 193 Default STP states

Device type	Default STP type	Default STP state	Default STP state of new VLANs ¹
Layer 2 Switch	MSTP ²	Enabled	Enabled
Layer 3 Switch	MSTP	Disabled	Disabled

1. When you create a port-based VLAN, the new VLAN STP state is the same as the default STP state on the device. The new VLAN does not inherit the STP state of the default VLAN.

2. MSTP stands for "Multiple Spanning Tree Protocol". In this type of STP, each port-based VLAN, including the default VLAN, has its own spanning tree. References in this documentation to "STP" apply to MSTP. The Single Spanning Tree Protocol (SSTP) is another type of STP. SSTP includes all VLANs on which STP is enabled in a single spanning tree. Refer to ["Single Spanning Tree \(SSTP\)"](#) on page 1151.

[Table 194](#) lists the default STP bridge parameters. The bridge parameters affect the entire spanning tree. If you are using MSTP, the parameters affect the VLAN. If you are using SSTP, the parameters affect all VLANs that are members of the single spanning tree.

TABLE 194 Default STP bridge parameters

Parameter	Description	Default and valid values
Forward Delay	The period of time spent by a port in the listening and learning state before moving on to the learning or forwarding state, respectively. The forward delay value is also used for the age time of dynamic entries in the filtering database, when a topology change occurs.	15 seconds Possible values: 4 – 30 seconds
Maximum Age	The interval a bridge will wait for a configuration BPDU from the root bridge before initiating a topology change.	20 seconds Possible values: 6 – 40 seconds

TABLE 194 Default STP bridge parameters (Continued)

Parameter	Description	Default and valid values
Hello Time	The interval of time between each configuration BPDU sent by the root bridge.	2 seconds Possible values: 1 – 10 seconds
Priority	A parameter used to identify the root bridge in a spanning tree (instance of STP). The bridge with the lowest value has the highest priority and is the root. A higher numerical value means a lower priority; thus, the highest priority is 0.	32768 Possible values: 0 – 65535

NOTE

If you plan to change STP bridge timers, Brocade recommends that you stay within the following ranges, from section 8.10.2 of the IEEE STP specification.

$$2 * (\text{forward_delay} - 1) \geq \text{max_age}$$

$$\text{max_age} \geq 2 * (\text{hello_time} + 1)$$

Table 195 lists the default STP port parameters. The port parameters affect individual ports and are separately configurable on each port.

TABLE 195 Default STP port parameters

Parameter	Description	Default and valid values
Priority	The preference that STP gives this port relative to other ports for forwarding traffic out of the spanning tree. A higher numerical value means a lower priority.	128 Possible values: 0 – 240 (configurable in increments of 16)
Path Cost	The cost of using the port to reach the root bridge. When selecting among multiple links to the root bridge, STP chooses the link with the lowest path cost and blocks the other paths. Each port type has its own default STP path cost.	10 Mbps – 100 100 Mbps – 19 Gbps – 4 10 Gbps – 2 Possible values are 0 – 65535

Enabling or disabling the Spanning Tree Protocol (STP)

STP is *enabled* by default on devices running Layer 2 code. STP is *disabled* by default on devices running Layer 3 code.

You can enable or disable STP on the following levels:

- **Globally** – Affects all ports and port-based VLANs on the device.
- **Port-based VLAN** – Affects all ports within the specified port-based VLAN. When you enable or disable STP within a port-based VLAN, the setting overrides the global setting. Thus, you can enable STP for the ports within a port-based VLAN even when STP is globally disabled, or disable the ports within a port-based VLAN when STP is globally enabled.
- **Individual port** – Affects only the individual port. However, if you change the STP state of the primary port in a trunk group, the change affects all ports in the trunk group.

NOTE

The CLI converts the STP groups into topology groups when you save the configuration. For backward compatibility, you can still use the STP group commands. However, the CLI converts the commands into the topology group syntax. Likewise, the **show stp-group** command displays STP topology groups.

Enabling or disabling STP globally

Use the following method to enable or disable STP on a device on which you have not configured port-based VLANs.

NOTE

When you configure a VLAN, the VLAN inherits the global STP settings. However, once you begin to define a VLAN, you can no longer configure standard STP parameters globally using the CLI. From that point on, you can configure STP only within individual VLANs.

To enable STP for all ports in all VLANs on a Brocade device, enter the **spanning-tree** command.

```
Brocade(config)#spanning-tree
```

The **spanning-tree** command enables a separate spanning tree in each VLAN, including the default VLAN.

Syntax: [no] spanning-tree

Enabling or disabling STP in a port-based VLAN

Use the following procedure to disable or enable STP on a device on which you have configured a port-based VLAN. Changing the STP state in a VLAN affects only that VLAN.

To enable STP for all ports in a port-based VLAN, enter commands such as the following.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#spanning-tree
```

Syntax: [no] spanning-tree

Enabling or disabling STP on an individual port

Use the following procedure to disable or enable STP on an individual port.

NOTE

If you change the STP state of the primary port in a trunk group, it affects all ports in the trunk group.

To enable STP on an individual port, enter commands such as the following.

```
Brocade(config)#interface 1/1
Brocade(config-if-e1000-1/1)#spanning-tree
```

Syntax: [no] spanning-tree

Changing STP bridge and port parameters

[Table 194](#) on page 1088 and [Table 195](#) on page 1089 list the default STP parameters. If you need to change the default value for an STP parameter, use the following procedures.

Changing STP bridge parameters

NOTE

If you plan to change STP bridge timers, Brocade recommends that you stay within the following ranges, from section 8.10.2 of the IEEE STP specification.

$$2 * (\text{forward_delay} - 1) \geq \text{max_age}$$

$$\text{max_age} \geq 2 * (\text{hello_time} + 1)$$

To change a STP bridge priority on a Brocade device to the highest value to make the device the root bridge, enter the following command.

```
Brocade(config)#spanning-tree priority 0
```

The command in this example changes the priority on a device on which you have not configured port-based VLANs. The change applies to the default VLAN. If you have configured a port-based VLAN on the device, you can configure the parameters only at the configuration level for individual VLANs. Enter commands such as the following.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#spanning-tree priority 0
```

To make this change in the default VLAN, enter the following commands.

```
Brocade(config)#vlan 1
Brocade(config-vlan-1)#spanning-tree priority 0
```

Syntax: [no] spanning-tree [forward-delay <value>] | [hello-time <value>] | [maximum-age <value>] | [priority <value>]

The **forward-delay** <value> parameter specifies the forward delay and can be a value from 4 – 30 seconds. The default is 15 seconds.

NOTE

You can configure a Brocade device for faster convergence (including a shorter forward delay) using Fast Span or Fast Uplink Span. Refer to [“STP feature configuration”](#) on page 1103.

The **hello-time** <value> parameter specifies the hello time and can be a value from 1 – 10 seconds. The default is 2 seconds.

NOTE

This parameter applies only when this device or VLAN is the root bridge for its spanning tree.

The **maximum-age** <value> parameter specifies the amount of time the device waits for receipt of a configuration BPDU from the root bridge before initiating a topology change. You can specify from 6 – 40 seconds. The default is 20 seconds.

The **priority** <value> parameter specifies the priority and can be a value from 0 – 65535. A higher numerical value means a lower priority. Thus, the highest priority is 0. The default is 32768.

You can specify some or all of these parameters on the same command line. If you specify more than one parameter, you must specify them in the order shown above, from left to right.

Changing STP port parameters

To change the path and priority costs for a port, enter commands such as the following.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#spanning-tree ethernet 5 path-cost 15 priority 64
```

Syntax: **spanning-tree ethernet** <port> **path-cost** <value> | **priority** <value> | **disable** | **enable**

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **path-cost** <value> parameter specifies the port cost as a path to the spanning tree root bridge. STP prefers the path with the lowest cost. You can specify a value from 0 – 65535.

The default depends on the port type:

- 10 Mbps – 100
- 100 Mbps – 19
- Gbps – 4
- 10 Gbps – 2
- The **priority** <value> parameter specifies the preference that STP gives this port relative to other ports for forwarding traffic out of the spanning tree. If you are upgrading a device that has a configuration saved under an earlier software release, and the configuration contains a value from 0 – 7 for a port STP priority, the software changes the priority to the default when you save the configuration while running the new release.

The **disable** | **enable** parameter disables or re-enables STP on the port. The STP state change affects only this VLAN. The port STP state in other VLANs is not changed.

NOTE

When you enable or disable STP auto negotiated combo ports on FESX devices, the ports may flap for a few seconds before the link is up

STP protection enhancement

STP protection provides the ability to prohibit an end station from initiating or participating in an STP topology change.

The 802.1W Spanning Tree Protocol (STP) detects and eliminates logical loops in a redundant network by selectively blocking some data paths (ports) and allowing only the best data paths to forward traffic.

In an STP environment, switches, end stations, and other Layer 2 devices use Bridge Protocol Data Units (BPDUs) to exchange information that STP will use to determine the best path for data flow. When a Layer 2 device is powered ON and connected to the network, or when a Layer 2 device goes down, it sends out an STP BPDU, triggering an STP topology change.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP Protection feature on the Brocade port to which the end station is connected. STP Protection disables the connected device ability to initiate or participate in an STP topology change, by dropping all BPDUs received from the connected device.

Enabling STP protection

You can enable STP Protection on a per-port basis.

To prevent an end station from initiating or participating in STP topology changes, enter the following command at the Interface level of the CLI.

```
Brocade#(config) interface e 2
Brocade#(config-if-e1000-2)#stp-protect
```

This command causes the port to drop STP BPDUs sent from the device on the other end of the link.

Syntax: `[no] stp-protect`

Enter the **no** form of the command to disable STP protection on the port.

Clearing BPDU drop counters

For each port that has STP Protection enabled, the Brocade device counts and records the number of dropped BPDUs. You can use CLI commands to clear the BPDU drop counters for all ports on the device, or for a specific port on the device.

To clear the BPDU drop counters for all ports on the device that have STP Protection enabled, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)#clear stp-protect-statistics
```

To clear the BPDU drop counter for a specific port that has STP Protection enabled, enter the following command at the Global CONFIG level of the CLI.

```
Brocade#clear stp-protect-statistics e 2
```

Syntax: `clear stp-protect-statistics [ethernet [<port>]] | [ethernet [<port>]]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Viewing the STP Protection Configuration

You can view the STP Protection configuration for all ports on a device, or for a specific port only. The **show stp-protect** command output shows the port number on which STP Protection is enabled, and the number of BPDUs dropped by each port.

To view the STP Protection configuration for all ports on the device, enter the following command at any level of the CLI.

Standard STP parameter configuration

```
Brocade#show stp-protect
Port ID          BPDU Drop Count
3                478
5                213
6                0
12              31
```

To view STP Protection configuration for a specific port, enter the following command at any level of the CLI.

```
Brocade#show stp-protect e 3
STP-protect is enabled on port 3. BPDU drop count is 478
```

If you enter the **show stp-protect** command for a port that does not have STP protection enabled, the following message displays on the console.

```
Brocade#show stp-protect e 4
STP-protect is not enabled on port 4.
```

Syntax: **show stp-protect** [**ethernet** <port>]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying STP information

You can display the following Spanning Tree Protocol (STP) information:

- All the global and interface STP settings
- CPU utilization statistics
- Detailed STP information for each interface
- STP state information for a port-based VLAN
- STP state information for an individual interface

Displaying STP information for an entire device

To display STP information, enter the following command at any level of the CLI.

```
Brocade#show span
```

```
VLAN 1 BPDU cam_index is 3 and the Master DMA Are(HEX)
STP instance owned by VLAN 1
```

Global STP (IEEE 802.1D) Parameters:

VLAN ID	Root ID	Root Cost	Root Port	Prio rity Hex	Max Age sec	He- llo sec	Ho- ld sec	Fwd dly sec	Last Chang sec	Chg cnt	Bridge Address
1	800000e0804d4a00	0	Root	8000	20	2	1	15	689	1	00e0804d4a00

Port STP Parameters:

Port Num	Prio rity Hex	Path Cost	State	Fwd Trans	Design Cost	Designated Root	Designated Bridge
1	80	19	FORWARDING	1	0	800000e0804d4a00	800000e0804d4a00
2	80	0	DISABLED	0	0	0000000000000000	0000000000000000
3	80	0	DISABLED	0	0	0000000000000000	0000000000000000
4	80	0	DISABLED	0	0	0000000000000000	0000000000000000
5	80	19	FORWARDING	1	0	800000e0804d4a00	800000e0804d4a00
6	80	19	BLOCKING	0	0	800000e0804d4a00	800000e0804d4a00
7	80	0	DISABLED	0	0	0000000000000000	0000000000000000

<lines for remaining ports excluded for brevity>

Syntax: `show span [vlan <vlan-id>] | [pvst-mode] | [<num>] | [detail [vlan <vlan-id> [ethernet [<port>] | <num>]]]`

The **vlan <vlan-id>** parameter displays STP information for the specified port-based VLAN.

The **pvst-mode** parameter displays STP information for the device Per VLAN Spanning Tree (PVST+) compatibility configuration. Refer to [“PVST/PVST+ compatibility”](#) on page 1157

Specify the **port** variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **<num>** parameter displays only the entries after the number you specify. For example, on a device with three port-based VLANs, if you enter 1, then information for the second and third VLANs is displayed, but information for the first VLAN is not displayed. Information is displayed according to VLAN number, in ascending order. The entry number is not the same as the VLAN number. For example, if you have port-based VLANs 1, 10, and 2024, then the command output has three STP entries. To display information for VLANs 10 and 2024 only, enter **show span 1**.

The **detail** parameter and its additional optional parameters display detailed information for individual ports. Refer to [“Displaying detailed STP information for each interface”](#) on page 1099.

The **show span** command shows the following information.

TABLE 196 CLI display of STP information

Field	Description
Global STP parameters	
VLAN ID	The port-based VLAN that contains this spanning tree (instance of STP). VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all STP information is for VLAN 1.
Root ID	The ID assigned by STP to the root bridge for this spanning tree.
Root Cost	The cumulative cost from this bridge to the root bridge. If this device is the root bridge, then the root cost is 0.
Root Port	The port on this device that connects to the root bridge. If this device is the root bridge, then the value is "Root" instead of a port number.
Priority Hex	This device or VLAN STP priority. The value is shown in hexadecimal format. NOTE: If you configure this value, specify it in decimal format. Refer to "Changing STP bridge parameters" on page 1091.
Max age sec	The number of seconds this device or VLAN waits for a configuration BPDU from the root bridge before deciding the root has become unavailable and performing a reconvergence.
Hello sec	The interval between each configuration BPDU sent by the root bridge.
Hold sec	The minimum number of seconds that must elapse between transmissions of consecutive Configuration BPDUs on a port.
Fwd dly sec	The number of seconds this device or VLAN waits following a topology change and consequent reconvergence.
Last Chang sec	The number of seconds since the last time a topology change occurred.
Chg cnt	The number of times the topology has changed since this device was reloaded.
Bridge Address	The STP address of this device or VLAN. NOTE: If this address is the same as the Root ID, then this device or VLAN is the root bridge for its spanning tree.
Port STP parameters	
Port Num	The port number.
Priority Hex	The port STP priority, in hexadecimal format. NOTE: If you configure this value, specify it in decimal format. Refer to "Changing STP port parameters" on page 1092.
Path Cost	The port STP path cost.

TABLE 196 CLI display of STP information (Continued)

Field	Description
State	<p>The port STP state. The state can be one of the following:</p> <ul style="list-style-type: none"> • BLOCKING – STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED – The port is not participating in STP. This can occur when the port is disconnected or STP is disabled on the port. • FORWARDING – STP is allowing the port to send and receive frames. • LISTENING – STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING – The port has passed through the LISTENING state and will change to the FORWARDING state, depending on the results of STP reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
Fwd Trans	The number of times STP has changed the state of this port between BLOCKING and FORWARDING.
Design Cost	The cost to the root bridge as advertised by the designated bridge that is connected to this port. If the designated bridge is the root bridge itself, then the cost is 0. The identity of the designated bridge is shown in the Design Bridge field.
Designated Root	The root bridge as recognized on this port. The value is the same as the root bridge ID listed in the Root ID field.
Designated Bridge	The designated bridge to which this port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.

Displaying CPU utilization statistics

You can display CPU utilization statistics for STP and the IP protocols.

To display CPU utilization statistics for STP for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI.

```

Brocade#show process cpu
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ARP             0.01       0.03       0.09       0.22        9
BGP             0.04       0.06       0.08       0.14       13
GVRP           0.00       0.00       0.00       0.00        0
ICMP           0.00       0.00       0.00       0.00        0
IP             0.00       0.00       0.00       0.00        0
OSPF           0.00       0.00       0.00       0.00        0
RIP            0.00       0.00       0.00       0.00        0
STP            0.00       0.03       0.04       0.07        4
VRRP           0.00       0.00       0.00       0.00        0

```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example.

Standard STP parameter configuration

```
Brocade#show process cpu
The system has only been up for 6 seconds.
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ARP              0.01        0.00        0.00        0.00         0
BGP              0.00        0.00        0.00        0.00         0
GVRP             0.00        0.00        0.00        0.00         0
ICMP             0.01        0.00        0.00        0.00         1
IP               0.00        0.00        0.00        0.00         0
OSPF             0.00        0.00        0.00        0.00         0
RIP              0.00        0.00        0.00        0.00         0
STP              0.00        0.00        0.00        0.00         0
VRRP            0.00        0.00        0.00        0.00         0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following.

```
Brocade#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name    Sec(%)    Time(ms)
ARP             0.00        0
BGP             0.00        0
GVRP            0.00        0
ICMP            0.01        1
IP              0.00        0
OSPF            0.00        0
RIP             0.00        0
STP             0.01        0
VRRP           0.00        0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: `show process cpu [<num>]`

The `<num>` parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying the STP state of a port-based VLAN

When you display information for a port-based VLAN, that information includes the STP state of the VLAN.

To display information for a port-based VLAN, enter a command such as the following at any level of the CLI. The STP state is shown in bold type in this example.


```
Brocade#show vlans
```

```
Total PORT-VLAN entries: 2
Maximum PORT-VLAN entries: 16
```

```
legend: [S=Slot]
```

```
PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree On
Untagged Ports: (S3)  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
Untagged Ports: (S3) 17 18 19 20 21 22 23 24
Untagged Ports: (S4)  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
Untagged Ports: (S4) 18 19 20 21 22 23 24
Tagged Ports: None
Uplink Ports: None
```

```
PORT-VLAN 2, Name greenwell, Priority level0, Spanning tree Off
Untagged Ports: (S1)  1  2  3  4  5  6  7  8
Untagged Ports: (S4)  1
Tagged Ports: None
Uplink Ports: None
```

Syntax: `show vlan [<vlan-id> | ethernet <port>]`

The `<vlan-id>` parameter specifies a VLAN for which you want to display the configuration information.

The `ethernet <port>` parameter specifies a port. If you use this parameter, the command lists all the VLAN memberships for the port. Specify the `port` variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying detailed STP information for each interface

To display the detailed STP information, enter the following command at any level of the CLI.

```
Brocade#show span detail
=====
VLAN 1 - MULTIPLE SPANNING TREE (MSTP) ACTIVE
=====
Bridge identifier      - 0x800000e0804d4a00
Active global timers - Hello: 0

Port 1/1 is FORWARDING
  Port - Path cost: 19, Priority: 128, Root: 0x800000e052a9bb00
  Designated - Bridge: 0x800000e052a9bb00, Interface: 1, Path cost: 0
  Active Timers - None
  BPDUs - Sent: 11, Received: 0
Port 1/2 is DISABLED
Port 1/3 is DISABLED
Port 1/4 is DISABLED
<lines for remaining ports excluded for brevity>
```

NOTE

The line in the above output, **VLAN 1 - MULTIPLE SPANNING TREE (MSTP) ACTIVE**, is not the 802.1s standard. It is the same Global STP (IEEE 802.1D) type as shown in the output of the **show span** CLI command.

If a port is disabled, the only information shown by this command is “DISABLED”. If a port is enabled, this display shows the following information.

Syntax: **show span detail** [**vlan** <vlan-id>] [**ethernet** <port>] <num>]

The **vlan** <vlan-id> parameter specifies a VLAN.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The <num> parameter specifies the number of VLANs you want the CLI to skip before displaying detailed STP information. For example, if the device has six VLANs configured (VLAN IDs 1, 2, 3, 99, 128, and 256) and you enter the command **show span detail 4**, detailed STP information is displayed for VLANs 128 and 256 only.

NOTE

If the configuration includes VLAN groups, the **show span detail** command displays the master VLANs of each group but not the member VLANs within the groups. However, the command does indicate that the VLAN is a master VLAN. The **show span detail vlan** <vlan-id> command displays the information for the VLAN even if it is a member VLAN. To list all the member VLANs within a VLAN group, enter the **show vlan-group** [<group-id>] command.

The **show span detail** command shows the following information.

TABLE 197 CLI display of detailed STP information for ports

Field	Description
Active Spanning Tree protocol	<p>The VLAN that contains the listed ports and the active Spanning Tree protocol. The STP type can be one of the following:</p> <ul style="list-style-type: none"> • MULTIPLE SPANNING TREE (MSTP) • GLOBAL SINGLE SPANNING TREE (SSTP) <p>NOTE: If STP is disabled on a VLAN, the command displays the following message instead: “Spanning-tree of port-vlan <vlan-id> is disabled.”</p>
Bridge identifier	The STP identity of this device.
Active global timers	<p>The global STP timers that are currently active, and their current values. The following timers can be listed:</p> <ul style="list-style-type: none"> • Hello – The interval between Hello packets. This timer applies only to the root bridge. • Topology Change (TC) – The amount of time during which the topology change flag in Hello packets will be marked, indicating a topology change. This timer applies only to the root bridge. • Topology Change Notification (TCN) – The interval between Topology Change Notification packets sent by a non-root bridge toward the root bridge. This timer applies only to non-root bridges.

TABLE 197 CLI display of detailed STP information for ports (Continued)

Field	Description
Port number and STP state	<p>The internal port number and the port STP state.</p> <p>The internal port number is one of the following:</p> <ul style="list-style-type: none"> • The port interface number, if the port is the designated port for the LAN. • The interface number of the designated port from the received BPDU, if the interface is not the designated port for the LAN. <p>The state can be one of the following:</p> <ul style="list-style-type: none"> • BLOCKING – STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED – The port is not participating in STP. This can occur when the port is disconnected or STP is administratively disabled on the port. • FORWARDING – STP is allowing the port to send and receive frames. • LISTENING – STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING – The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table. <p>NOTE: If the state is DISABLED, no further STP information is displayed for the port.</p>
Port Path cost	The STP path cost for the port.
Port Priority	This STP priority for the port. The value is shown as a hexadecimal number.
Root	The ID assigned by STP to the root bridge for this spanning tree.
Designated Bridge	The MAC address of the designated bridge to which this port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.
Designated Port	The port number sent from the designated bridge.
Designated Path Cost	The cost to the root bridge as advertised by the designated bridge that is connected to this port. If the bridge is the root bridge itself, then the cost is 0. The identity of the designated bridge is shown in the Designated Bridge field.
Active Timers	<p>The current values for the following timers, if active:</p> <ul style="list-style-type: none"> • Message age – The number of seconds this port has been waiting for a hello message from the root bridge. • Forward delay – The number of seconds that have passed since the last topology change and consequent reconvergence. • Hold time – The number of seconds that have elapsed since transmission of the last Configuration BPDU.
BPDUs Sent and Received	The number of BPDUs sent and received on this port since the software was reloaded.

Displaying detailed STP information for a single port in a specific VLAN

Enter a command such as the following to display STP information for an individual port in a specific VLAN.

```
Brocade#show span detail vlan 1 ethernet 7/1
Port 7/1 is FORWARDING
  Port - Path cost: 19, Priority: 128, Root: 0x800000e052a9bb00
  Designated - Bridge: 0x800000e052a9bb00, Interface: 7, Path cost: 0
  Active Timers - None
  BPDUs - Sent: 29, Received: 0
```

Syntax: `show span detail [vlan <vlan-id> ethernet <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying STP state information for an individual interface

To display STP state information for an individual port, you can use the methods in [“Displaying STP information for an entire device”](#) on page 1095 or [“Displaying detailed STP information for each interface”](#) on page 1099. You also can display STP state information for a specific port using the following method.

To display information for a specific port, enter a command such as the following at any level of the CLI.

```
Brocade#show interface ethernet 3/11

FastEthernet3/11 is up, line protocol is up
  Hardware is FastEthernet, address is 00e0.52a9.bb49 (bia 00e0.52a9.bb49)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  STP configured to ON, priority is level0, flow control enabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1518 bytes, encapsulation ethernet
  5 minute input rate: 352 bits/sec, 0 packets/sec, 0.00% utilization
  5 minute output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  1238 packets input, 79232 bytes, 0 no buffer
  Received 686 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 ignored
  529 multicast
  918 packets output, 63766 bytes, 0 underruns
  0 output errors, 0 collisions
```

The STP information is shown in bold type in this example.

Syntax: `show interfaces [ethernet <port>] | [loopback <num>] | [slot <slot-num>] | [ve <num>] | [brief]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

You also can display the STP states of all ports by entering the **show interface brief** command such as the following, which uses the **brief** parameter.

```
Brocade#show interface brief
```

Port	Link	State	Dupl	Speed	Trunk	Tag	Priori	MAC	Name
1/1	Down	None	None	None	None	No	level0	00e0.52a9.bb00	
1/2	Down	None	None	None	None	No	level0	00e0.52a9.bb01	
1/3	Down	None	None	None	None	No	level0	00e0.52a9.bb02	
1/4	Down	None	None	None	None	No	level0	00e0.52a9.bb03	
1/5	Down	None	None	None	None	No	level0	00e0.52a9.bb04	
1/6	Down	None	None	None	None	No	level0	00e0.52a9.bb05	
1/7	Down	None	None	None	None	No	level0	00e0.52a9.bb06	
1/8	Down	None	None	None	None	No	level0	00e0.52a9.bb07	
. some rows omitted for brevity									
3/10	Down	None	None	None	None	No	level0	00e0.52a9.bb4a	
3/11	Up	Forward	Full	100M	None	No	level0	00e0.52a9.bb49	

In the example above, only one port, 3/11, is forwarding traffic toward the root bridge.

STP feature configuration

Spanning Tree Protocol (STP) features extend the operation of standard STP, enabling you to fine-tune standard STP and avoid some of its limitations.

This section describes how to configure these parameters on Brocade Layer 3 Switches using the CLI.

Fast port span

When STP is running on a device, message forwarding is delayed during the spanning tree recalculation period following a topology change. The STP forward delay parameter specifies the period of time a bridge waits before forwarding data packets. The forward delay controls the listening and learning periods of STP reconvergence. You can configure the forward delay to a value from 4 – 30 seconds. The default is 15 seconds. Thus, using the standard forward delay, convergence requires 30 seconds (15 seconds for listening and an additional 15 seconds for learning) when the default value is used.

This slow convergence is undesirable and unnecessary in some circumstances. The Fast Port Span feature allows certain ports to enter the forwarding state in four seconds. Specifically, Fast Port Span allows faster convergence on ports that are attached to end stations and thus do not present the potential to cause Layer 2 forwarding loops. Because the end stations cannot cause forwarding loops, they can safely go through the STP state changes (blocking to listening to learning to forwarding) more quickly than is allowed by the standard STP convergence time. Fast Port Span performs the convergence on these ports in four seconds (two seconds for listening and two seconds for learning).

In addition, Fast Port Span enhances overall network performance in the following ways:

- Fast Port Span reduces the number of STP topology change notifications on the network. When an end station attached to a Fast Span port comes up or down, the Brocade device does not generate a topology change notification for the port. In this situation, the notification is unnecessary since a change in the state of the host does not affect the network topology.
- Fast Port Span eliminates unnecessary MAC cache aging that can be caused by topology change notifications. Bridging devices age out the learned MAC addresses in their MAC caches if the addresses are unrefreshed for a given period of time, sometimes called the MAC aging interval. When STP sends a topology change notification, devices that receive the notification use the value of the STP forward delay to quickly age out their MAC caches. For example, if a device normal MAC aging interval is 5 minutes, the aging interval changes temporarily to the value of the forward delay (for example, 15 seconds) in response to an STP topology change.

In normal STP, the accelerated cache aging occurs even when a single host goes up or down. Because Fast Port Span does not send a topology change notification when a host on a Fast Port Span port goes up or down, the unnecessary cache aging that can occur in these circumstances under normal STP is eliminated.

Fast Port Span is a system-wide parameter and is enabled by default. Thus, when you boot a device, all the ports that are attached only to end stations run Fast Port Span. For ports that are not eligible for Fast Port Span, such as ports connected to other networking devices, the device automatically uses the normal STP settings. If a port matches any of the following criteria, the port is ineligible for Fast Port Span and uses normal STP instead:

- The port is 802.1Q tagged
- The port is a member of a trunk group
- The port has learned more than one active MAC address
- An STP Configuration BPDU has been received on the port, thus indicating the presence of another bridge on the port.

You also can explicitly exclude individual ports from Fast Port Span if needed. For example, if the only uplink ports for a wiring closet switch are Gbps ports, you can exclude the ports from Fast Port Span.

Disabling and re-enabling fast port span

Fast Port Span is a system-wide parameter and is enabled by default. Therefore, all ports that are eligible for Fast Port Span use it.

To disable or re-enable Fast Port Span, enter the following commands.

```
Brocade(config)#no fast port-span
Brocade(config)#write memory
```

Syntax: [no] fast port-span

NOTE

The **fast port-span** command has additional parameters that let you exclude specific ports. These parameters are shown in the following section.

To re-enable Fast Port Span, enter the following commands.

```
Brocade(config)#fast port-span
Brocade(config)#write memory
```

Excluding specific ports from fast port span

To exclude a port from Fast Port Span while leaving Fast Port Span enabled globally, enter commands such as the following.

```
Brocade(config)#fast port-span exclude ethernet 1
Brocade(config)#write memory
```

To exclude a set of ports from Fast Port Span, enter commands such as the following.

```
Brocade(config)#fast port-span exclude ethernet 1 ethernet 2 ethernet 3
Brocade(config)#write memory
```

To exclude a contiguous (unbroken) range of ports from Fast Span, enter commands such as the following.

```
Brocade(config)#fast port-span exclude ethernet 1 to 24
Brocade(config)#write memory
```

Syntax: **[no] fast port-span [exclude ethernet <port> [ethernet <port>] | to [<port>]]**

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To re-enable Fast Port Span on a port, enter a command such as the following.

```
Brocade(config)#no fast port-span exclude ethernet 1
Brocade(config)#write memory
```

This command re-enables Fast Port Span on port 1 only and does not re-enable Fast Port Span on other excluded ports. You also can re-enable Fast Port Span on a list or range of ports using the syntax shown above this example.

To re-enable Fast Port Span on all excluded ports, disable and then re-enable Fast Port Span by entering the following commands.

```
Brocade(config)#no fast port-span
Brocade(config)#fast port-span
Brocade(config)#write memory
```

Disabling and then re-enabling Fast Port Span clears the exclude settings and thus enables Fast Port Span on all eligible ports. To make sure Fast Port Span remains enabled on the ports following a system reset, save the configuration changes to the startup-config file after you re-enable Fast Port Span. Otherwise, when the system resets, those ports will again be excluded from Fast Port Span.

Fast Uplink Span

The Fast Port Span feature described in the previous section enhances STP performance for end stations. The Fast Uplink Span feature enhances STP performance for wiring closet switches with redundant uplinks. Using the default value for the standard STP forward delay, convergence following a transition from an active link to a redundant link can take 30 seconds (15 seconds for listening and an additional 15 seconds for learning).

You can use the Fast Uplink Span feature on a Brocade device deployed as a wiring closet switch to decrease the convergence time for the uplink ports to another device to just one second. The new Uplink port directly goes to forward mode (bypassing listening and learning modes). The wiring closet switch must be a Brocade device but the device at the other end of the link can be a Brocade device or another vendor's switch.

Configuration of the Fast Uplink Span feature takes place entirely on the Brocade device. To configure the Fast Uplink Span feature, specify a group of ports that have redundant uplinks on the wiring closet switch (Brocade device). If the active link becomes unavailable, the Fast Uplink Span feature transitions the forwarding to one of the other redundant uplink ports in just one second. All Fast Uplink Span-enabled ports are members of a single Fast Uplink Span group.

NOTE

To avoid the potential for temporary bridging loops, Brocade recommends that you use the Fast Uplink feature only for wiring closet switches (switches at the edge of the network cloud). In addition, enable the feature only on a group of ports intended for redundancy, so that at any given time only one of the ports is expected to be in the forwarding state.

NOTE

When the wiring closet switch (Brocade device) first comes up or when STP is first enabled, the uplink ports still must go through the standard STP state transition without any acceleration. This behavior guards against temporary routing loops as the switch tries to determine the states for all the ports. Fast Uplink Span acceleration applies only when a working uplink becomes unavailable.

Active uplink port failure

The active uplink port is the port elected as the root port using the standard STP rules. All other ports in the group are redundant uplink ports. If an active uplink port becomes unavailable, Fast Uplink Span transitions the forwarding of traffic to one of the redundant ports in the Fast Uplink Span group in one second bypassing listening and learning port states.

Switchover to the active uplink port

When a failed active uplink port becomes available again, switchover from the redundant port to the active uplink port is delayed by 30 seconds. The delay allows the remote port to transition to forwarding mode using the standard STP rules. After 30 seconds, the blocked active uplink port begins forwarding in just one second and the redundant port is blocked.

NOTE

Use caution when changing the spanning tree priority. If the switch becomes the root bridge, Fast Uplink Span will be disabled automatically.

Fast Uplink Span Rules for Trunk Groups

If you add a port to a Fast Uplink Span group that is a member of a trunk group, the following rules apply:

- If you add the primary port of a trunk group to the Fast Uplink Span group, all other ports in the trunk group are automatically included in the group. Similarly, if you remove the primary port in a trunk group from the Fast Uplink Span group, the other ports in the trunk group are automatically removed from the Fast Uplink Span group.
- You cannot add a subset of the ports in a trunk group to the Fast Uplink Span group. All ports in a trunk group have the same Fast Uplink Span property, as they do for other port properties.
- If the working trunk group is partially down but not completely down, no switch-over to the backup occurs. This behavior is the same as in the standard STP feature.
- If the working trunk group is completely down, a backup trunk group can go through an accelerated transition only if the following are true:
 - The trunk group is included in the fast uplink group.
 - All other ports except those in this trunk group are either disabled or blocked. The accelerated transition applies to all ports in this trunk group.

When the original working trunk group comes back (partially or fully), the transition back to the original topology is accelerated if the conditions listed above are met.

Configuring a Fast Uplink Port Group

To configure a group of ports for Fast Uplink Span, enter the following commands:

```
Brocade(config)# fast uplink-span ethernet 4/1 to 4/4
Brocade(config)# write memory
```

Syntax: [no] **fast uplink-span** [ethernet <port> [ethernet <port>... | to <port>]]

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

This example configures four ports, 4/1 – 4/4, as a Fast Uplink Span group. In this example, all four ports are connected to a wiring closet switch. Only one of the links is expected to be active at any time. The other links are redundant. For example, if the link on port 4/1 is the active link on the wiring closet switch but becomes unavailable, one of the other links takes over. Because the ports are configured in a Fast Uplink Span group, the STP convergence takes one second instead of taking at least 30 seconds using the standard STP forward delay.

You can add ports to a Fast Uplink Span group by entering the fast uplink-span command additional times with additional ports. The device can have only one Fast Uplink Span group, so all the ports you identify as Fast Uplink Span ports are members of the same group.

STP feature configuration

To remove a Fast Uplink Span group or to remove individual ports from a group, use “no” in front of the appropriate fast uplink-span command. For example, to remove ports 4/3 and 4/4 from the Fast Uplink Span group configured above, enter the following commands:

```
Brocade(config)# no fast uplink-span ethernet 4/3 to 4/4
Brocade(config)# write memory
```

To check the status of ports with Fast Uplink Span enabled.

```
Brocade(config)# show span fast-uplink-span
```

STP instance owned by VLAN 1

Global STP (IEEE 802.1D) Parameters:

VLAN ID	Root ID	Root Cost	Root Port	Prio rity Hex	Max Age sec	He- llo sec	Ho- ld sec	Fwd dly sec	Last Chang sec	Chg cnt	Bridge Address
1	000000c100000001	2	1/3/1	8000	20	2	1	15	65	15	000011111111

Port STP Parameters:

Port Num	Prio rity Hex	Path Cost	State	Fwd Trans	Design Cost	Designated Root	Designated Bridge
1/1/2	80	0	DISABLED	0	0	0000000000000000	0000000000000000
1/1/3	80	0	DISABLED	0	0	0000000000000000	0000000000000000
1/1/4	80	4	FORWARDING	1	2	000000c100000001	8000000011111111
1/1/5	80	0	DISABLED	0	0	0000000000000000	0000000000000000
1/1/6	80	0	DISABLED	0	0	0000000000000000	0000000000000000
1/1/7	80	0	DISABLED	0	0	0000000000000000	0000000000000000
1/1/8	80	0	DISABLED	0	0	0000000000000000	0000000000000000
1/1/9	80	0	DISABLED	0	0	0000000000000000	0000000000000000

Syntax: show span fast-uplink-span

Configuring Fast Uplink Span within a VLAN

You can also configure Fast Uplink Span on the interfaces within a VLAN.

To configure Fast Uplink Span for a VLAN, enter command such as the following.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#untag ethernet 8/1 to 8/2
Brocade(config-vlan-10)#fast uplink-span ethernet 8/1 to 8/2
```

Syntax: [no] fast uplink-span ethernet <port-no>

To check the status of Fast Uplink Span for a specified VLAN.

```
Brocade(config-vlan-2)#show span vlan 2 fast-uplink-span
```

STP instance owned by VLAN 2

Global STP (IEEE 802.1D) Parameters:

VLAN ID	Root ID	Root Cost	Root Port	Prio rity Hex	Max Age sec	He- llo sec	Ho- ld sec	Fwd dly sec	Last Chang sec	Chg cnt	Bridge Address
2	800000000111111111	0	Root	8000	20	2	1	15	29596	0	000011111111

Port STP Parameters:

Port Num	Prio rity	Path Cost	State	Fwd Trans	Design Cost	Designated Root	Designated Bridge
	Hex						
1/1/1	80	4	LISTENING	0	0	8000000011111111	8000000011111111

Syntax: `show span vlan <vlan-id> fast-uplink-span`

The VLAN <vlan-id> parameter displays Fast Uplink Span information for the specified VLAN.

802.1W Rapid Spanning Tree (RSTP)

Earlier implementation by Brocade of Rapid Spanning Tree Protocol (RSTP), which was 802.1W Draft 3, provided only a subset of the IEEE 802.1W standard; whereas the 802.1W RSTP feature provides the full standard. The implementation of the 802.1W Draft 3 is referred to as RSTP Draft 3.

RSTP Draft3 will continue to be supported on Brocade devices for backward compatibility. However, customers who are currently using RSTP Draft 3 should migrate to 802.1W.

The 802.1W feature provides rapid traffic reconvergence for point-to-point links within a few milliseconds (0 – 500 milliseconds), following the failure of a bridge or bridge port. This reconvergence occurs more rapidly than the reconvergence provided by the 802.1D Spanning Tree Protocol (STP) or by RSTP Draft 3.

NOTE

This rapid convergence will not occur on ports connected to shared media devices, such as hubs. To take advantage of the rapid convergence provided by 802.1W, make sure to explicitly configure all point-to-point links in a topology.

The convergence provided by the standard 802.1W protocol occurs more rapidly than the convergence provided by previous spanning tree protocols because of the following:

- Classic or legacy 802.1D STP protocol requires a newly selected Root port to go through listening and learning stages before traffic convergence can be achieved. The 802.1D traffic convergence time is calculated using the following formula.

2 x FORWARD_DELAY + BRIDGE_MAX_AGE.

If default values are used in the parameter configuration, convergence can take up to 50 seconds. (In this document STP will be referred to as 802.1D.)

- RSTP Draft 3 works only on bridges that have Alternate ports, which are the precalculated “next best root port”. (Alternate ports provide back up paths to the root bridge.) Although convergence occurs from 0 – 500 milliseconds in RSTP Draft 3, the spanning tree topology reverts to the 802.1D convergence if an Alternate port is not found.
- Convergence in 802.1w bridge is not based on any timer values. Rather, it is based on the explicit handshakes between Designated ports and their connected Root ports to achieve convergence in less than 500 milliseconds.

Bridges and bridge port roles

A bridge in an 802.1W rapid spanning tree topology is assigned as the root bridge if it has the highest priority (lowest bridge identifier) in the topology. Other bridges are referred to as non-root bridges.

Unique roles are assigned to ports on the root and non-root bridges. Role assignments are based on the following information contained in the Rapid Spanning Tree Bridge Packet Data Unit (RST BPDUs):

- Root bridge ID
- Path cost value
- Transmitting bridge ID
- Designated port ID

The 802.1W algorithm uses this information to determine if the RST BPDUs received by a port is superior to the RST BPDUs that the port transmits. The two values are compared in the order as given above, starting with the Root bridge ID. The RST BPDUs with a lower value is considered superior. The superiority and inferiority of the RST BPDUs is used to assign a role to a port.

If the value of the received RST BPDUs is the same as that of the transmitted RST BPDUs, then the port ID in the RST BPDUs are compared. The RST BPDUs with the lower port ID is superior. Port roles are then calculated appropriately.

The port role is included in the BPDUs that it transmits. The BPDUs transmitted by an 802.1W port is referred to as an RST BPDUs, while it is operating in 802.1W mode.

Ports can have one of the following roles:

- **Root** – Provides the lowest cost path to the root bridge from a specific bridge
- **Designated** – Provides the lowest cost path to the root bridge from a LAN to which it is connected
- **Alternate** – Provides an alternate path to the root bridge when the root port goes down
- **Backup** – Provides a backup to the LAN when the Designated port goes down
- **Disabled** – Has no role in the topology

Assignment of port roles

At system start-up, all 802.1W-enabled bridge ports assume a Designated role. Once start-up is complete, the 802.1W algorithm calculates the superiority or inferiority of the RST BPDUs that is received and transmitted on a port.

On a root bridge, each port is assigned a **Designated port** role, except for ports on the same bridge that are physically connected together. In these type of ports, the port that receives the superior RST BPDUs becomes the **Backup port**, while the other port becomes the **Designated port**.

On non-root bridges, ports are assigned as follows:

- The port that receives the RST BPDUs with the lowest path cost from the root bridge becomes the **Root port**.
- If two ports on the same bridge are physically connected, the port that receives the superior RST BPDUs becomes the **Backup port**, while the other port becomes the **Designated port**.
- If a non-root bridge already has a Root port, then the port that receives an RST BPDUs that is superior to those it can transmit becomes the **Alternate port**.
- If the RST BPDUs that a port receives is inferior to the RST BPDUs it transmits, then the port becomes a **Designated port**.
- If the port is down or if 802.1W is disabled on the port, that port is given the role of **Disabled port**. Disabled ports have no role in the topology. However, if 802.1W is enabled on a port with a link down and the link of that port comes up, then that port assumes one of the following port roles: Root, Designated, Alternate, or Backup.

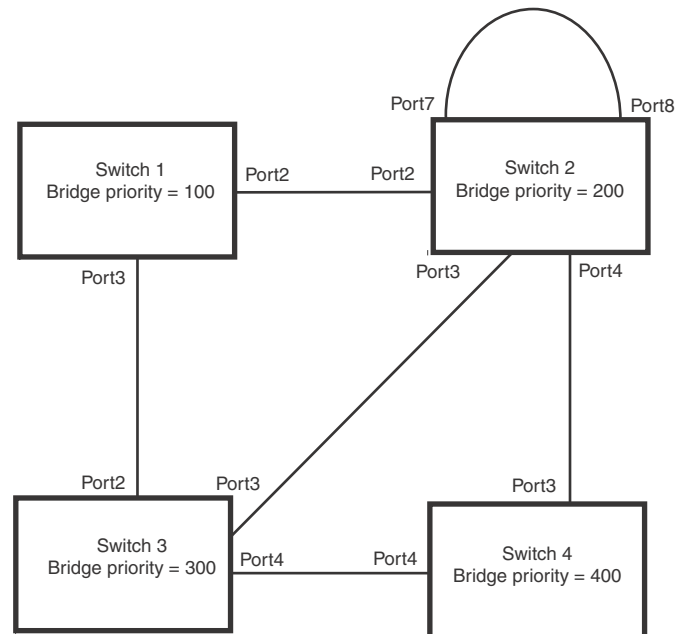
The following example (Figure 126) explains role assignments in a simple RSTP topology.

NOTE

All examples in this document assume that all ports in the illustrated topologies are point-to-point links and are homogeneous (they have the same path cost value) unless otherwise specified.

The topology in Figure 126 contains four bridges. Switch 1 is the root bridge since it has the lowest bridge priority. Switch 2 through Switch 4 are non-root bridges.

FIGURE 126 Simple 802.1W topology



Assignment of ports on Switch 1

All ports on Switch 1, the root bridge, are assigned Designated port roles.

Assignment of ports on Switch 2

Port2 on Switch 2 directly connects to the root bridge; therefore, Port2 is the Root port.

The bridge priority value on Switch 2 is superior to that of Switch 3 and Switch 4; therefore, the ports on Switch 2 that connect to Switch 3 and Switch 4 are given the Designated port role.

Furthermore, Port7 and Port8 on Switch 2 are physically connected. The RST BPDUs transmitted by Port7 are superior to those Port8 transmits. Therefore, Port8 is the Backup port and Port7 is the Designated port.

Assignment of ports on Switch 3

Port2 on Switch 3 directly connects to the Designated port on the root bridge; therefore, it assumes the Root port role.

The root path cost of the RST BPDUs received on Port4/Switch 3 is inferior to the RST BPDUs transmitted by the port; therefore, Port4/Switch 3 becomes the Designated port.

Similarly Switch 3 has a bridge priority value inferior to Switch 2. Port3 on Switch 3 connects to Port 3 on Switch 2. This port will be given the Alternate port role, since a Root port is already established on this bridge.

Assignment of ports on Switch 4

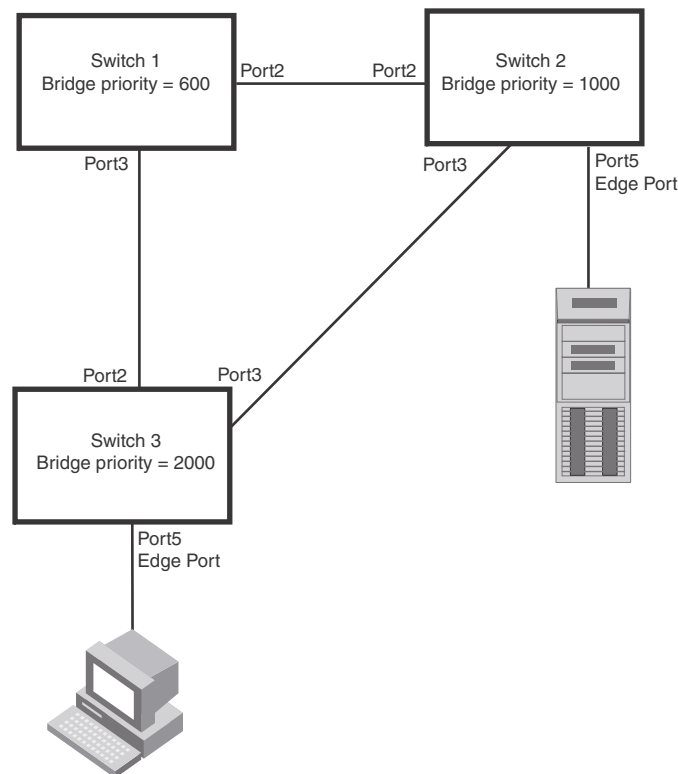
Switch 4 is not directly connected to the root bridge. It has two ports with superior incoming RST BPDUs from two separate LANs: Port3 and Port4. The RST BPDUs received on Port3 are superior to the RST BPDUs received on port 4; therefore, Port3 becomes the Root port and Port4 becomes the Alternate port.

Edge ports and edge port roles

The Brocade implementation of 802.1W allows ports that are configured as Edge ports to be present in an 802.1W topology. (Figure 127). Edge ports are ports of a bridge that connect to workstations or computers. Edge ports do not register any incoming BPDU activities.

Edge ports assume Designated port roles. Port flapping does not cause any topology change events on Edge ports since 802.1W does not consider Edge ports in the spanning tree calculations.

FIGURE 127 Topology with edge ports



However, if any incoming RST BPDUs are received from a previously configured Edge port, 802.1W automatically makes the port as a non-edge port. This is extremely important to ensure a loop free Layer 2 operation since a non-edge port is part of the active RSTP topology.

The 802.1W protocol can auto-detect an Edge port and a non-edge port. An administrator can also configure a port to be an Edge port using the CLI. It is recommended that Edge ports are configured explicitly to take advantage of the Edge port feature, instead of allowing the protocol to auto-detect them.

Point-to-point ports

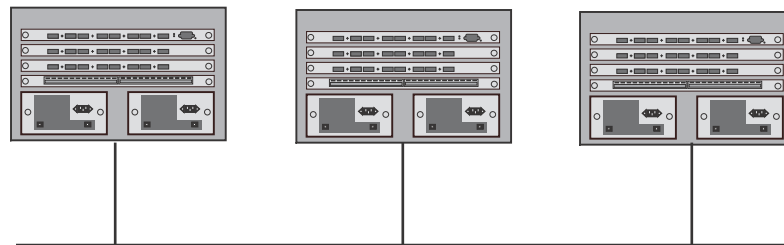
To take advantage of the 802.1W features, ports on an 802.1W topology should be explicitly configured as point-to-point links using the CLI. Shared media should not be configured as point-to-point links.

NOTE

Configuring shared media or non-point-to-point links as point-to-point links could lead to Layer 2 loops.

The topology in [Figure 128](#) is an example of shared media that should not be configured as point-to-point links. In [Figure 128](#), a port on a bridge communicates or is connected to at least two ports.

FIGURE 128 Example of shared media



Bridge port states

Ports roles can have one of the following states:

- **Forwarding** – 802.1W is allowing the port to send and receive all packets.
- **Discarding** – 802.1W has blocked data traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is forwarding. When a port is in this state, the port does not transmit or receive data frames, but the port does continue to receive RST BPDUs. This state corresponds to the listening and blocking states of 802.1D.
- **Learning** – 802.1W is allowing MAC entries to be added to the filtering database but does not permit forwarding of data frames. The device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
- **Disabled** – The port is not participating in 802.1W. This can occur when the port is disconnected or 802.1W is administratively disabled on the port.

A port on a non-root bridge with the role of Root port is always in a forwarding state. If another port on that bridge assumes the Root port role, then the old Root port moves into a discarding state as it assumes another port role.

A port on a non-root bridge with a Designated role starts in the discarding state. When that port becomes elected to the Root port role, 802.1W quickly places it into a forwarding state. However, if the Designated port is an Edge port, then the port starts and stays in a forwarding state and it cannot be elected as a Root port.

A port with an Alternate or Backup role is always in a discarding state. If the port role changes to Designated, then the port changes into a forwarding state.

If a port on one bridge has a Designated role and that port is connected to a port on another bridge that has an Alternate or Backup role, the port with a Designated role cannot be given a Root port role until two instances of the forward delay timer expires on that port.

Edge port and non-edge port states

As soon as a port is configured as an Edge port using the CLI, it goes into a forwarding state instantly (in less than 100 msec).

When the link to a port comes up and 802.1W detects that the port is an Edge port, that port instantly goes into a forwarding state.

If 802.1W detects that port as a non-edge port, the port state is changed as determined by the result of processing the received RST BPDU. The port state change occurs within four seconds of link up or after two hello timer expires on the port.

Changes to port roles and states

To achieve convergence in a topology, a port role and state changes as it receives and transmits new RST BPDUs. Changes in a port role and state constitute a topology change. Besides the superiority and inferiority of the RST BPDU, bridge-wide and per-port state machines are used to determine a port role as well as a port state. Port state machines also determine when port role and state changes occur.

Port Role Selection state machines

The bridge uses the Port Role Selection state machine to determine if port role changes are required on the bridge. This state machine performs a computation when one of the following events occur:

- New information is received on any port on the bridge
- The timer expires for the current information on a port on the bridge

Each port uses the following state machines:

- **Port Information** – This state machine keeps track of spanning-tree information currently used by the port. It records the origin of the information and ages out any information that was derived from an incoming BPDU.
- **Port Role Transition** – This state machine keeps track of the current port role and transitions the port to the appropriate role when required. It moves the Root port and the Designated port into forwarding states and moves the Alternate and Backup ports into discarding states.
- **Port Transmit** – This state machine is responsible for BPDU transmission. It checks to ensure only the maximum number of BPDUs per hello interval are sent every second. Based on what mode it is operating in, it sends out either legacy BPDUs or RST BPDUs. In this document legacy BPDUs are also referred to as STP BPDUs.

- **Port Protocol Migration** – This state machine deals with compatibility with 802.1D bridges. When a legacy BPDU is detected on a port, this state machine configures the port to transmit and receive legacy BPDUs and operate in the legacy mode.
- **Topology Change** – This state machine detects, generates, and propagates topology change notifications. It acknowledges Topology Change Notice (TCN) messages when operating in 802.1D mode. It also flushes the MAC table when a topology change event takes place.
- **Port State Transition** – This state machine transitions the port to a discarding, learning, or forwarding state and performs any necessary processing associated with the state changes.
- **Port Timers** – This state machine is responsible for triggering any of the state machines described above, based on expiration of specific port timers.

In contrast to the 802.1D standard, the 802.1W standard does not have any bridge specific timers. All timers in the CLI are applied on a per-port basis, even though they are configured under bridge parameters.

802.1W state machines attempt to quickly place the ports into either a forwarding or discarding state. Root ports are quickly placed in forwarding state when both of the following events occur:

- It is assigned to be the Root port.
- It receives an RST BPDU with a proposal flag from a Designated port. The proposal flag is sent by ports with a Designated role when they are ready to move into a forwarding state.

When a the role of Root port is given to another port, the old Root port is instructed to reroot. The old Root port goes into a discarding state and negotiates with its peer port for a new role and a new state. A peer port is the port on the other bridge to which the port is connected. For example, in [Figure 129](#), Port1 of Switch 200 is the peer port of Port2 of Switch 100.

A port with a Designated role is quickly placed into a forwarding state if one of the following occurs:

- The Designated port receives an RST BPDU that contains an agreement flag from a Root port
- The Designated port is an Edge port

However, a Designated port that is attached to an Alternate port or a Backup port must wait until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state.

Backup ports are quickly placed into discarding states.

Alternate ports are quickly placed into discarding states.

A port operating in 802.1W mode may enter a learning state to allow MAC entries to be added to the filtering database; however, this state is transient and lasts only a few milliseconds, if the port is operating in 802.1W mode and if the port meets the conditions for rapid transition.

Handshake mechanisms

To rapidly transition a Designated or Root port into a forwarding state, the Port Role Transition state machine uses handshake mechanisms to ensure loop free operations. It uses one type of handshake if no Root port has been assigned on a bridge, and another type if a Root port has already been assigned.

Handshake when no root port is elected

If a Root port has not been assigned on a bridge, 802.1W uses the *Proposing -> Proposed -> Sync -> Synced -> Agreed* handshake:

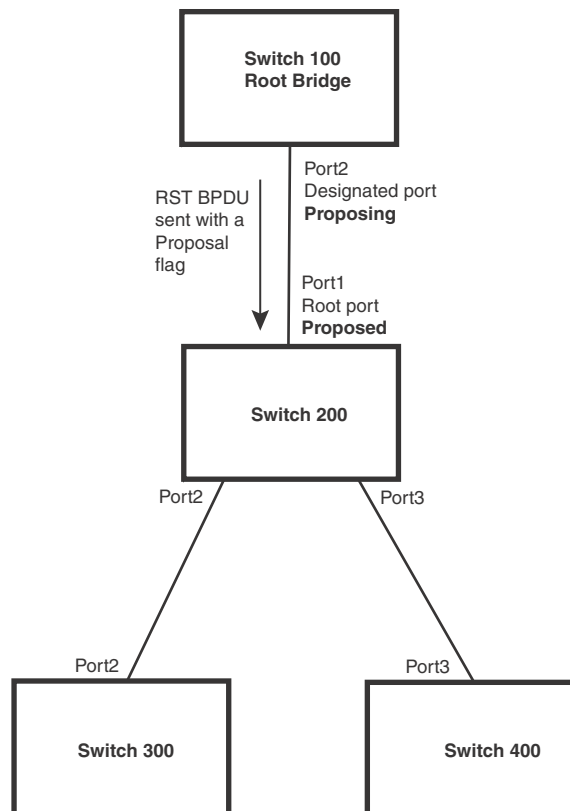
- **Proposing** – The Designated port on the root bridge sends an RST BPDUs packet to its peer port that contains a proposal flag. The proposal flag is a signal that indicates that the Designated port is ready to put itself in a forwarding state (Figure 129). The Designated port continues to send this flag in its RST BPDUs until it is placed in a forwarding state (Figure 132) or is forced to operate in 802.1D mode. (Refer to “Compatibility of 802.1W with 802.1D” on page 1136).
- **Proposed** – When a port receives an RST BPDUs with a proposal flag from the Designated port on its point-to-point link, it asserts the Proposed signal and one of the following occurs (Figure 129):
 - If the RST BPDUs that the port receives is superior to what it can transmit, the port assumes the role of a Root port. (Refer to the section on “Bridges and bridge port roles” on page 1109.)
 - If the RST BPDUs that the port receives is inferior to what it can transmit, then the port is given the role of Designated port.

NOTE

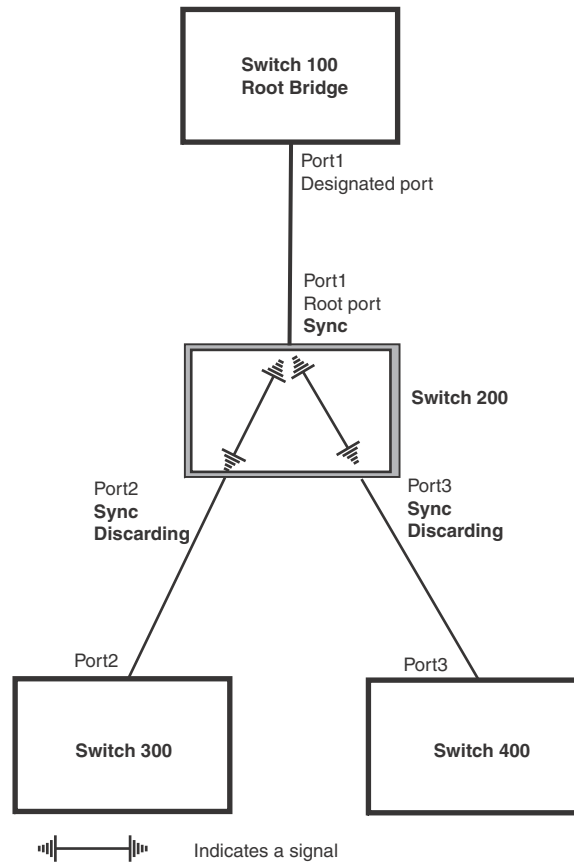
Proposed will never be asserted if the port is connected on a shared media link.

In Figure 129, Port3/Switch 200 is elected as the Root port

FIGURE 129 Proposing and proposed stage

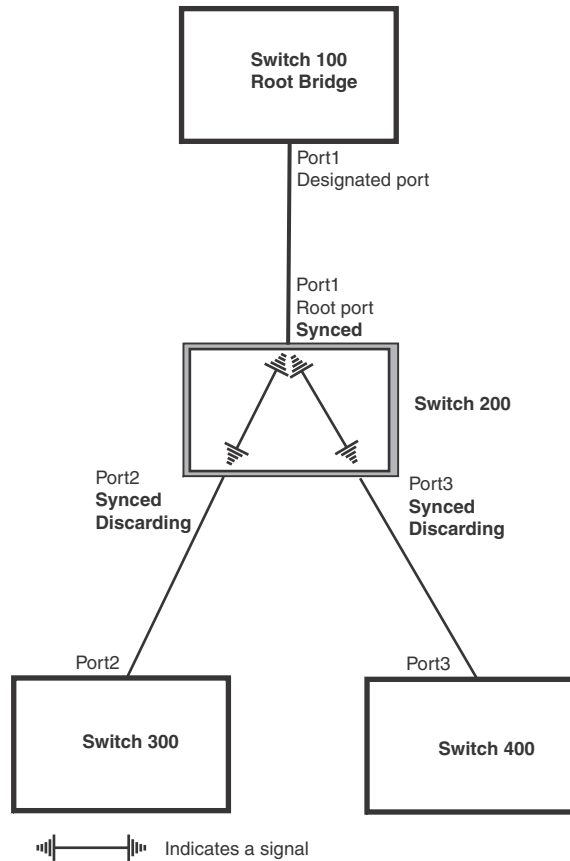


- **Sync** – Once the Root port is elected, it sets a sync signal on all the ports on the bridge. The signal tells the ports to synchronize their roles and states ([Figure 130](#)). Ports that are non-edge ports with a role of Designated port change into a discarding state. These ports have to negotiate with their peer ports to establish their new roles and states.

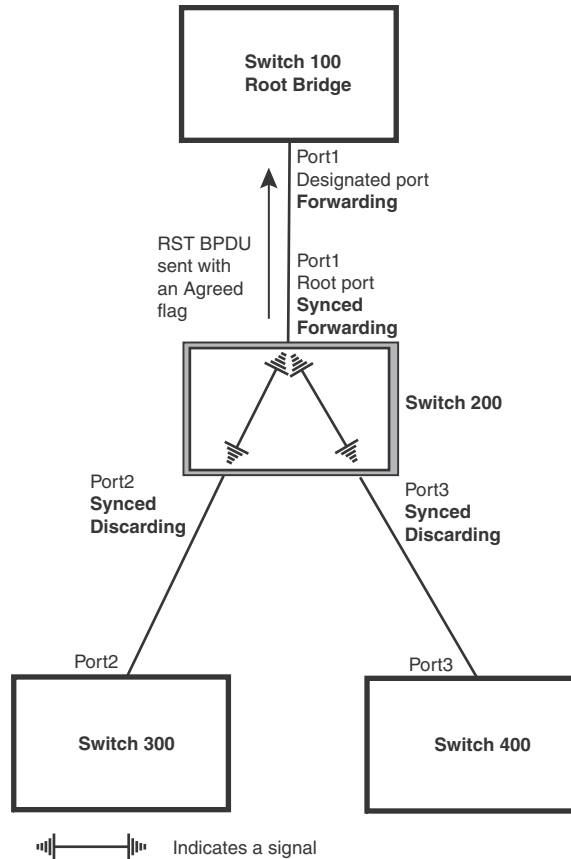
FIGURE 130 Sync stage

- **Synced** – Once the Designated port changes into a discarding state, it asserts a synced signal. Immediately, Alternate ports and Backup ports are synced. The Root port monitors the synced signals from all the bridge ports. Once all bridge ports asserts a synced signal, the Root port asserts its own synced signal ([Figure 131](#)).

FIGURE 131 Synced stage



- **Agreed** – The Root port sends back an RST BPDU containing an agreed flag to its peer Designated port and moves into the forwarding state. When the peer Designated port receives the RST BPDU, it rapidly transitions into a forwarding state.

FIGURE 132 Agree stage

At this point, the handshake mechanism is complete between Switch 100, the root bridge, and Switch 200.

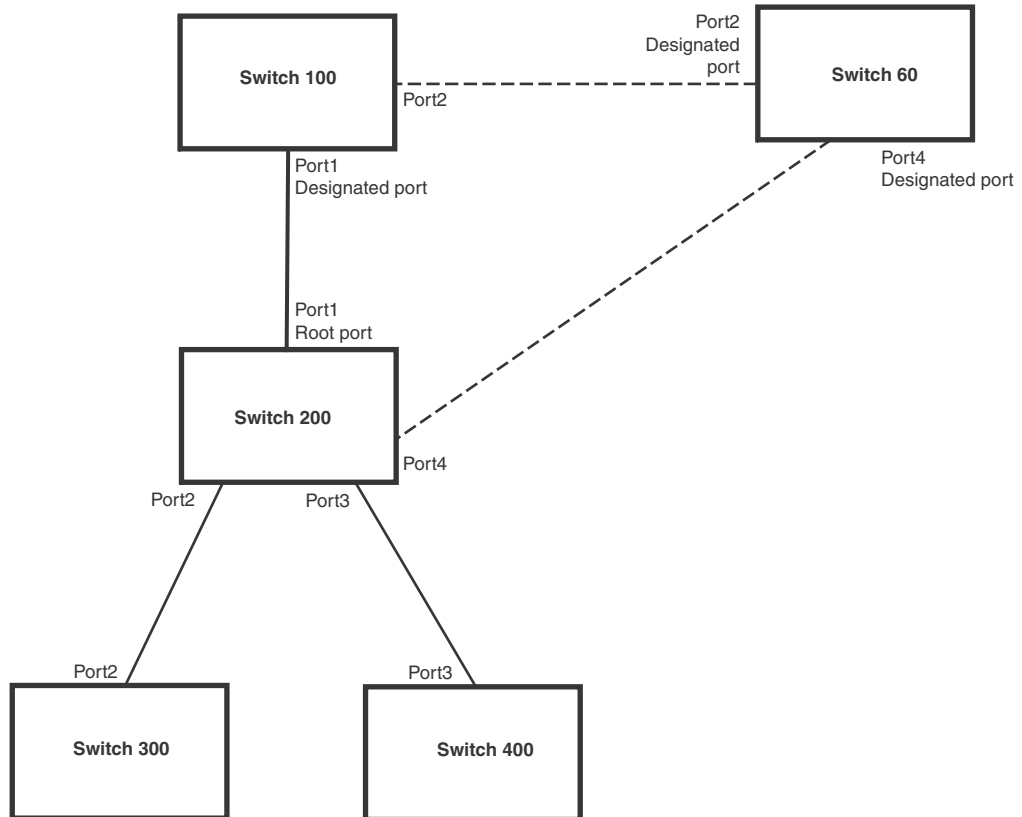
Switch 200 updates the information on the Switch 200 Designated ports (Port2 and Port3) and identifies the new root bridge. The Designated ports send RST BPDUs, containing proposal flags, to their downstream bridges, without waiting for the hello timers to expire on them. This process starts the handshake with the downstream bridges.

For example, Port2/Switch 200 sends an RST BPDU to Port2/Switch 300 that contains a proposal flag. Port2/Switch 300 asserts a proposed signal. Ports in Switch 300 then set sync signals on the ports to synchronize and negotiate their roles and states. Then the ports assert a synced signal and when the Root port in Switch 300 asserts its synced signal, it sends an RST BPDU to Switch 200 with an agreed flag.

This handshake is repeated between Switch 200 and Switch 400 until all Designated and Root ports are in forwarding states.

Handshake when a root port has been elected

If a non-root bridge already has a Root port, 802.1W uses a different type of handshake. For example, in [Figure 133](#), a new root bridge is added to the topology.

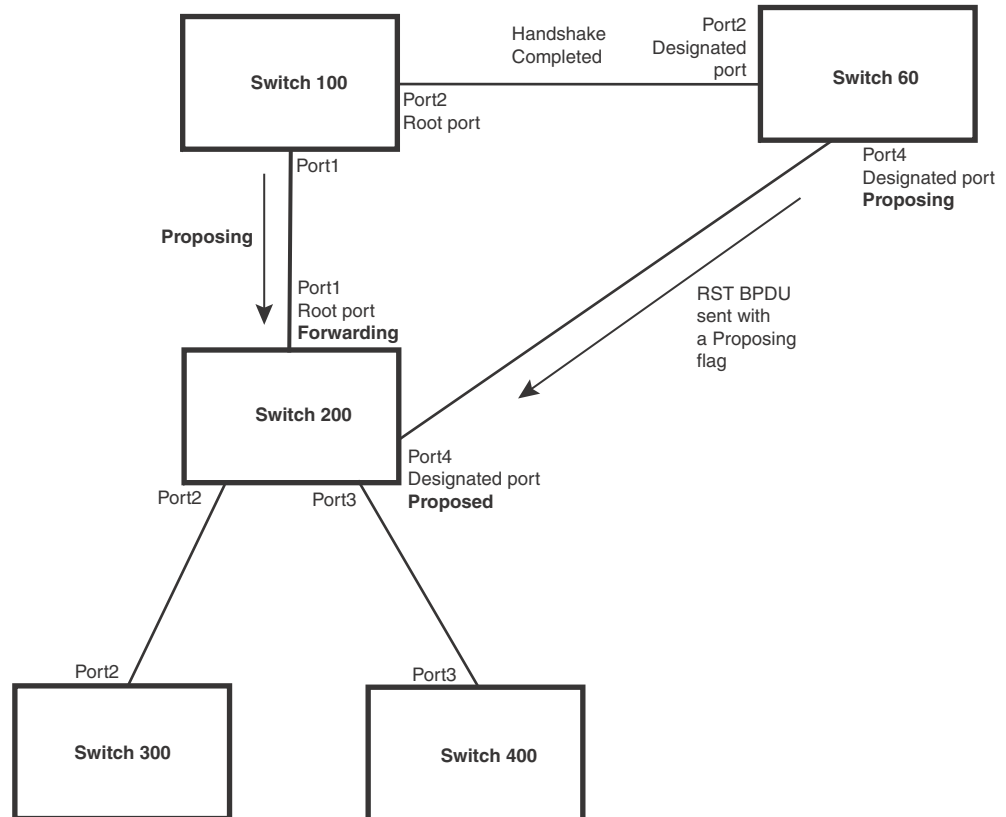
FIGURE 133 Addition of a new root bridge

The handshake that occurs between Switch 60 and Switch 100 follows the one described in the previous section ([“Handshake when no root port is elected”](#) on page 1115). The former root bridge becomes a non-root bridge and establishes a Root port ([Figure 134](#)).

However, since Switch 200 already had a Root port in a forwarding state, 802.1W uses the *Proposing -> Proposed -> Sync* and *Reroot -> Sync* and *Rerooted -> Rerooted* and *Synced -> Agreed handshake*:

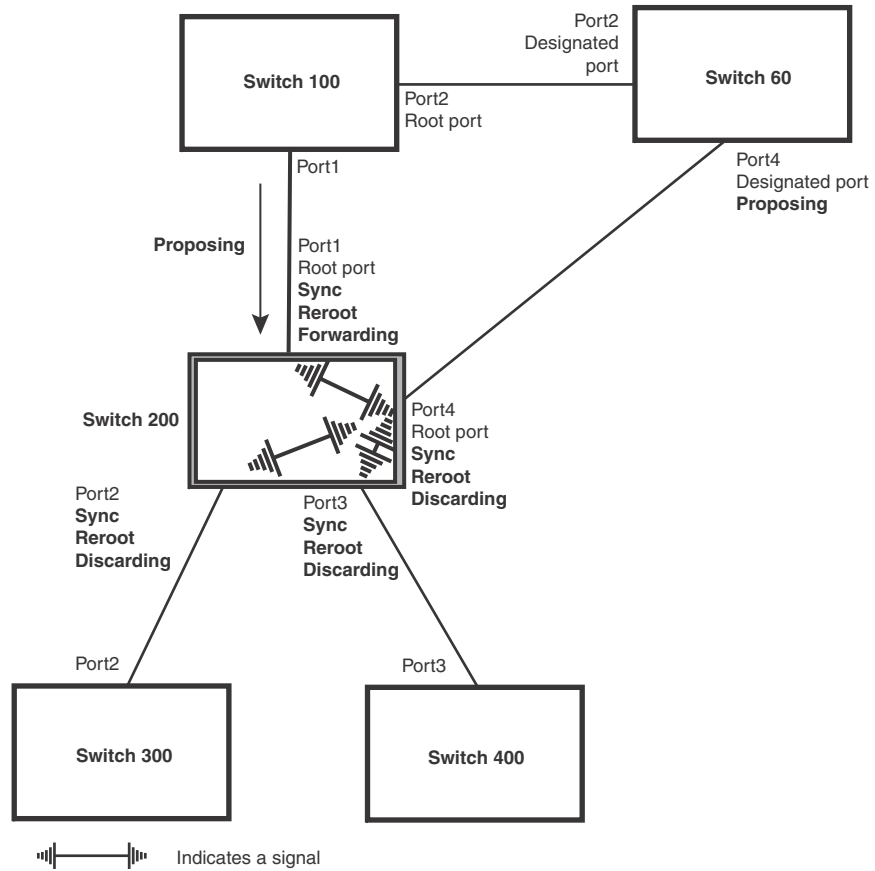
- Proposing and Proposed** – The Designated port on the new root bridge (Port4/Switch 60) sends an RST BPDU that contains a proposing signal to Port4/Switch 200 to inform the port that it is ready to put itself in a forwarding state (Figure 134). 802.1W algorithm determines that the RST BPDU that Port4/Switch 200 received is superior to what it can generate, so Port4/Switch 200 assumes a Root port role.

FIGURE 134 New root bridge sending a proposal flag

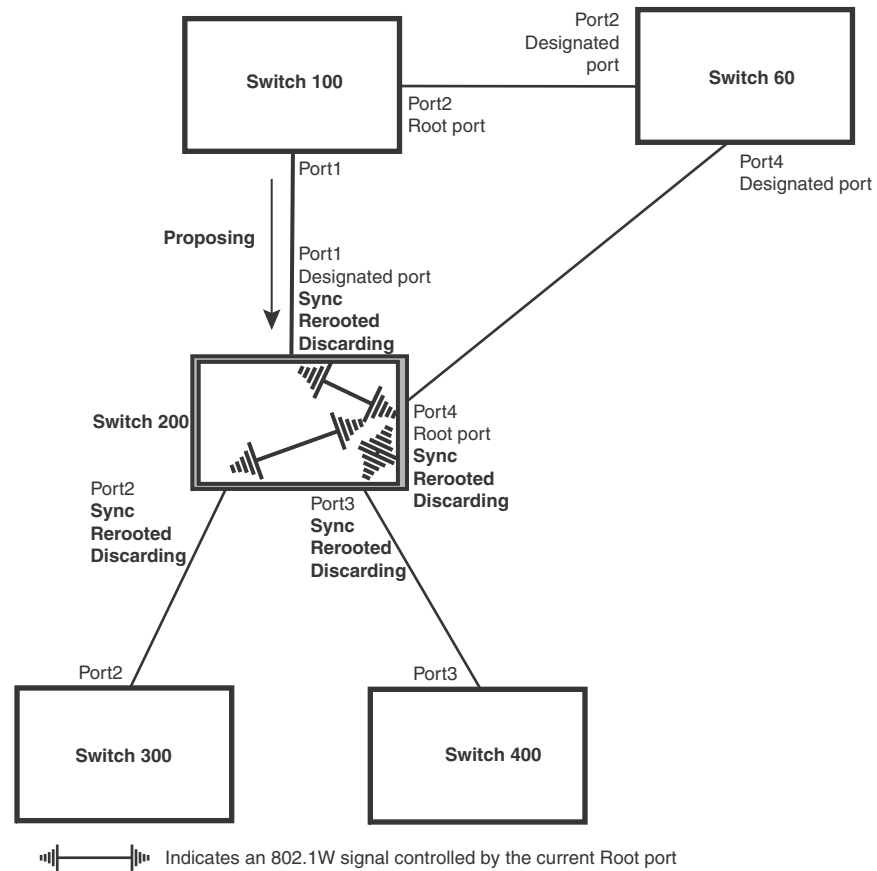


- **Sync and Reroot** – The Root port then asserts a sync and a reroot signal on all the ports on the bridge. The signal tells the ports that a new Root port has been assigned and they are to renegotiate their new roles and states. The other ports on the bridge assert their sync and reroot signals. Information about the old Root port is discarded from all ports. Designated ports change into discarding states ([Figure 135](#)).

FIGURE 135 Sync and reroot

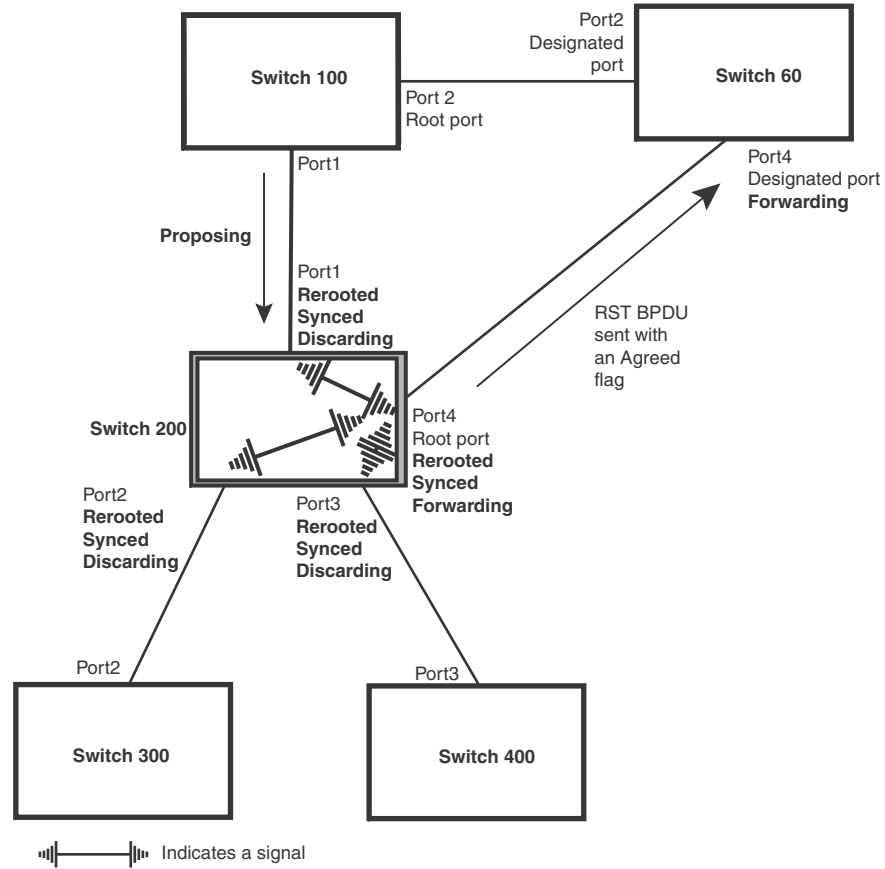


- **Sync and Rerooted** – When the ports on Switch 200 have completed the reroot phase, they assert their sync signals and continue to assert their sync signals as they continue in their discarding states. They also continue to negotiate their roles and states with their peer ports (Figure 136).

FIGURE 136 Sync and rerooted

- **Synced and Agree** – When all the ports on the bridge assert their synced signals, the new Root port asserts its own synced signal and sends an RST BPDU to Port4/Switch 60 that contains an agreed flag (Figure 136). The Root port also moves into a forwarding state.

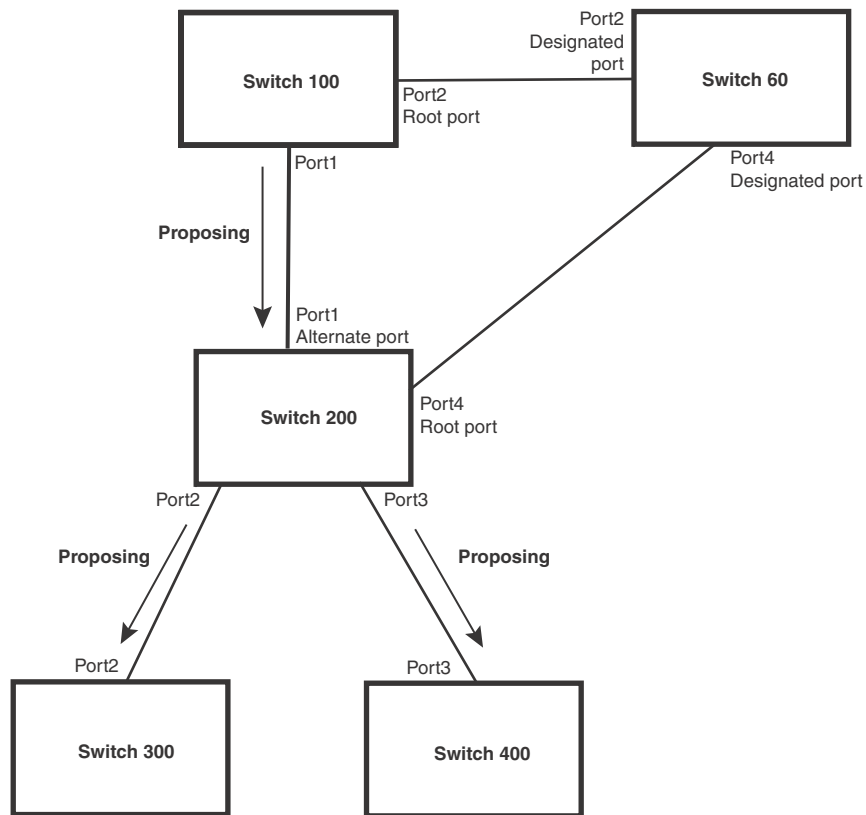
FIGURE 137 Rerooted, synced, and agreed



The old Root port on Switch 200 becomes an Alternate Port (Figure 138). Other ports on that bridge are elected to appropriate roles.

The Designated port on Switch 60 goes into a forwarding state once it receives the RST BPDU with the agreed flag.

FIGURE 138 Handshake completed after election of new root port



Recall that Switch 200 sent the agreed flag to Port4/Switch 60 and not to Port1/Switch 100 (the port that connects Switch 100 to Switch 200). Therefore, Port1/Switch 100 does not go into forwarding state instantly. It waits until two instances of the forward delay timer expires on the port before it goes into forwarding state.

At this point the handshake between the Switch 60 and Switch 200 is complete.

The remaining bridges (Switch 300 and Switch 400) may have to go through the reroot handshake if a new Root port needs to be assigned.

802.1W convergence in a simple topology

The examples in this section illustrate how 802.1W convergence occurs in a simple Layer 2 topology at start-up.

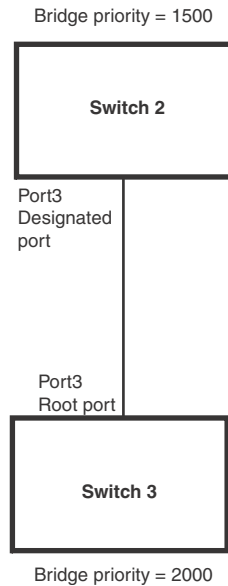
NOTE

The remaining examples assume that the appropriate handshake mechanisms occur as port roles and states change.

Convergence at start up

In [Figure 139](#), two bridges Switch 2 and Switch 3 are powered up. There are point-to-point connections between Port3/Switch 2 and Port3/Switch 3.

FIGURE 139 Convergence between two bridges



At power up, all ports on Switch 2 and Switch 3 assume Designated port roles and are at discarding states before they receive any RST BPDU.

Port3/Switch 2, with a Designated role, transmits an RST BPDU with a proposal flag to Port3/Switch 3. A ports with a Designated role sends the proposal flag in its RST BPDU when they are ready to move to a forwarding state.

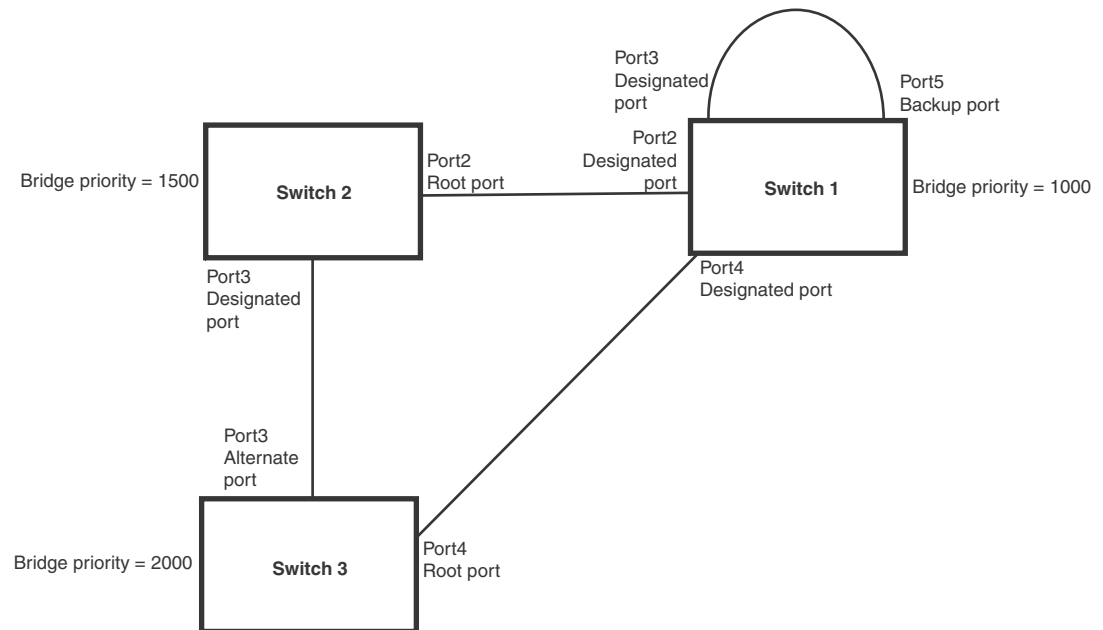
Port3/Switch 3, which starts with a role of Designated port, receives the RST BPDU and finds that it is superior to what it can transmit; therefore, Port3/Switch 3 assumes a new port role, that of a Root port. Port3/Switch 3 transmits an RST BPDU with an agreed flag back to Switch 2 and immediately goes into a forwarding state.

Port3/Switch 2 receives the RST BPDU from Port3/Switch 3 and immediately goes into a forwarding state.

Now 802.1W has fully converged between the two bridges, with Port3/Switch 3 as an operational root port in forwarding state and Port3/Switch 2 as an operational Designated port in forwarding state.

Next, Switch 1 is powered up ([Figure 140](#)).

FIGURE 140 Simple Layer 2 topology



The point-to-point connections between the three bridges are as follows:

- Port2/Switch 1 and Port2/Switch 2
- Port4/Switch 1 and Port4/Switch 3
- Port3/Switch 2 and Port3/Switch 3

Ports 3 and 5 on Switch 1 are physically connected together.

At start up, the ports on Switch 1 assume Designated port roles, which are in discarding state. They begin sending RST BPDUs with proposal flags to move into a forwarding state.

When Port4/Switch 3 receives these RST BPDUs 802.1W algorithm determines that they are better than the RST BPDUs that were previously received on Port3/Switch 3. Port4/Switch 3 is now selected as Root port. This new assignment signals Port3/Switch 3 to begin entering the discarding state and to assume an Alternate port role. As it goes through the transition, Port3/Switch 3 negotiates a new role and state with its peer port, Port3/Switch 2.

Port4/Switch 3 sends an RST BPDU with an agreed flag to Port4/Switch 1. Both ports go into forwarding states.

Port2/Switch 2 receives an RST BPDU. The 802.1W algorithm determines that these RST BPDUs that are superior to any that any port on Switch 2 can transmit; therefore, Port2/Switch 2 assumes the role of a Root port.

The new Root port then signals all ports on the bridge to start synchronization. Since none of the ports are Edge ports, they all enter the discarding state and assume the role of Designated ports. Port3/Switch 2, which previously had a Designated role with a forwarding state, starts the discarding state. They also negotiate port roles and states with their peer ports. Port3/Switch 2 also sends an RST BPU to Port3/Switch 3 with a proposal flag to request permission go into a forwarding state.

STP feature configuration

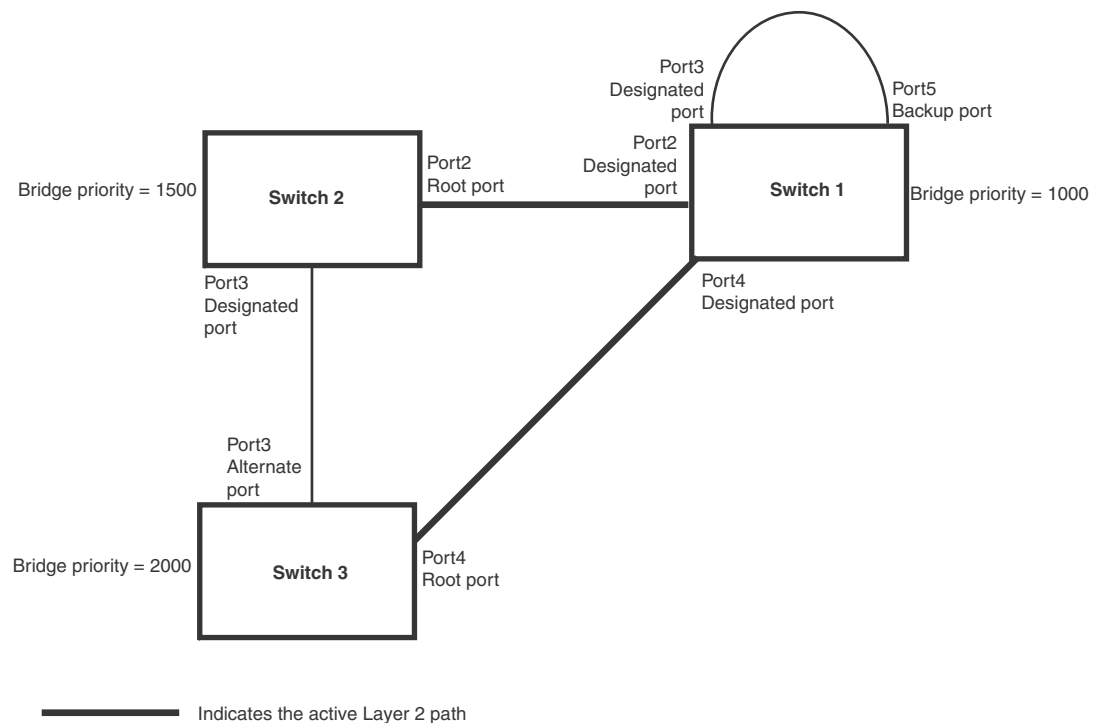
The Port2/Switch 2 bridge also sends an RST BPDU with an agreed flag Port2/Switch 1 that Port2 is the new Root port. Both ports go into forwarding states.

Now, Port3/Switch 3 is currently in a discarding state and is negotiating a port role. It received RST BPDUs from Port3/Switch 2. The 802.1W algorithm determines that the RST BPDUs Port3/Switch 3 received are superior to those it can transmit; however, they are not superior to those that are currently being received by the current Root port (Port4). Therefore, Port3 retains the role of Alternate port.

Ports 3/Switch 1 and Port5/Switch 1 are physically connected. Port5/Switch 1 received RST BPDUs that are superior to those received on Port3/Switch 1; therefore, Port5/Switch 1 is given the Backup port role while Port3 is given the Designated port role. Port3/Switch 1, does not go directly into a forwarding state. It waits until the forward delay time expires twice on that port before it can proceed to the forwarding state.

Once convergence is achieved, the active Layer 2 forwarding path converges as shown in [Figure 141](#).

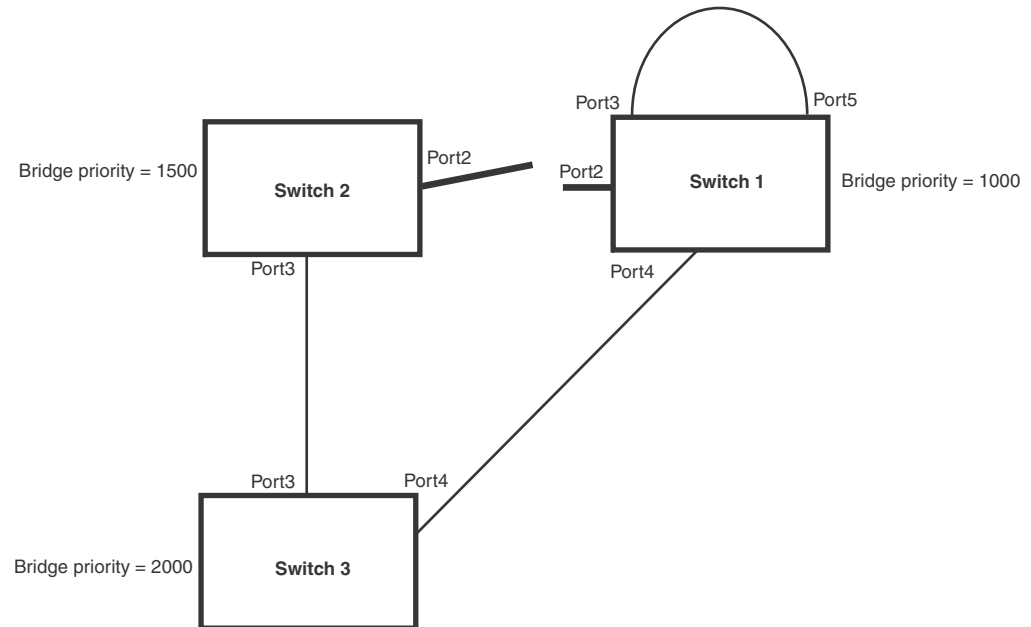
FIGURE 141 Active Layer 2 path



Convergence after a link failure

Figure 142 illustrates a link failure in the 802.1W topology. In this example, Port2/Switch, which is the port that connects Switch 2 to the root bridge (Switch 1), failed and both Switch 2 and Switch 1 are affected by the topology change.

FIGURE 142 Link failure in the topology



Switch 1 sets its Port2 into a discarding state.

At the same time, Switch 2 assumes the role of a root bridge since its root port failed and it has no operational Alternate port. Port3/Switch 2, which currently has a Designated port role, sends an RST BPDU to Switch 3. The RST BPDU contains a proposal flag and a bridge ID of Switch 2 as its root bridge ID.

When Port3/Switch 3 receives the RST BPDUs, 802.1W algorithm determines that they are inferior to those that the port can transmit. Therefore, Port3/Switch 3 is given a new role, that of a Designated port. Port3/Switch 3 then sends an RST BPDU with a proposal flag to Switch 2, along with the new role information. However, the root bridge ID transmitted in the RST BPDU is still Switch 1.

When Port3/Switch 2 receives the RST BPDU, 802.1W algorithm determines that it is superior to the RST BPDU that it can transmit; therefore, Port3/Switch 2 receives a new role; that of a Root port. Port3/Switch 2 then sends an RST BPDU with an agreed flag to Port3/Switch 3. Port3/Switch 2 goes into a forwarding state.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, Port3/Switch 3 changes into a forwarding state, which then completes the full convergence of the topology.

Convergence at link restoration

When Port2/Switch 2 is restored, both Switch 2 and Switch 1 recognize the change. Port2/Switch 1 starts assuming the role of a Designated port and sends an RST BPDU containing a proposal flag to Port2/Switch 2.

When Port2/Switch 2 receives the RST BPDUs, 802.1W algorithm determines that the RST BPDUs the port received are better than those received on Port3/Switch 3; therefore, Port2/Switch 2 is given the role of a Root port. All the ports on Switch 2 are informed that a new Root port has been assigned which then signals all the ports to synchronize their roles and states. Port3/Switch 2, which was the previous Root port, enters a discarding state and negotiates with other ports on the bridge to establish its new role and state, until it finally assumes the role of a Designated port.

Next, the following happens:

- Port3/Switch 2, the Designated port, sends an RST BPDU, with a proposal flag to Port3/Switch 3.
- Port2/Switch 2 also sends an RST BPDU with an agreed flag to Port2/Switch 1 and then places itself into a forwarding state.

When Port2/Switch 1 receives the RST BPDU with an agreed flag sent by Port2/Switch 2, it puts that port into a forwarding state. The topology is now fully converged.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, 802.1W algorithm determines that these RST BPDUs are superior to those that Port3/Switch 3 can transmit. Therefore, Port3/Switch 3 is given a new role, that of an Alternate port. Port3/Switch 3 immediately enters a discarding state.

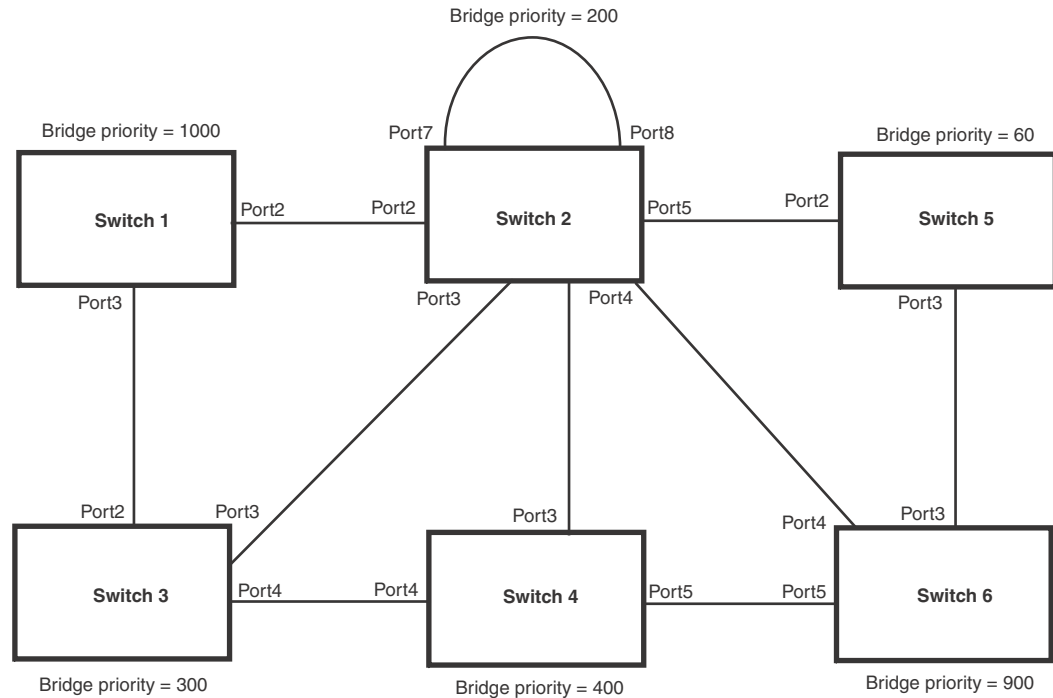
Now Port3/Switch 2 does not go into a forwarding state instantly like the Root port. It waits until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state. The wait, however, does not cause a denial of service, since the essential connectivity in the topology has already been established.

When fully restored, the topology is the same as that shown on [Figure 140](#).

Convergence in a complex 802.1W topology

Figure 143 illustrates a complex 802.1W topology.

FIGURE 143 Complex 802.1W topology



In Figure 143, Switch 5 is selected as the root bridge since it is the bridge with the highest priority. Lines in the figure show the point-to-point connection to the bridges in the topology.

Switch 5 sends an RST BPDU that contains a proposal flag to Port5/Switch 2. When handshakes are completed in Switch 5, Port5/Switch 2 is selected as the Root port on Switch 2. All other ports on Switch 2 are given Designated port role with discarding states.

Port5/Switch 2 then sends an RST BPDU with an agreed flag to Switch 5 to confirm that it is the new Root port and the port enters a forwarding state. Port7 and Port8 are informed of the identity of the new Root port. 802.1W algorithm selects Port7 as the Designated port while Port8 becomes the Backup port.

Port3/Switch 5 sends an RST BPDU to Port3/Switch 6 with a proposal flag. When Port3/Switch 5 receives the RST BPDU, handshake mechanisms select Port3 as the Root port of Switch 6. All other ports are given a Designated port role with discarding states. Port3/Switch 6 then sends an RST BPDU with an agreed flag to Port3/Switch 5 to confirm that it is the Root port. The Root port then goes into a forwarding state.

Now, Port4/Switch 6 receives RST BPDUs that are superior to what it can transmit; therefore, it is given the Alternate port role. The port remains in discarding state.

Port5/Switch 6 receives RST BPDUs that are inferior to what it can transmit. The port is then given a Designated port role.

Next Switch 2 sends RST BPDUs with a proposal flag to Port3/Switch 4. Port3 becomes the Root port for the bridge; all other ports are given a Designated port role with discarding states. Port3/Switch 4 sends an RST BPDU with an agreed flag to Switch 2 to confirm that it is the new Root port. The port then goes into a forwarding state.

Now Port4/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is then given an Alternate port role, and remains in discarding state.

Likewise, Port5/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is also given an Alternate port role, and remains in discarding state.

Port2/Switch 2 transmits an RST BPDU with a proposal flag to Port2/Switch 1. Port2/Switch 1 becomes the Root port. All other ports on Switch 1 are given Designated port roles with discarding states.

Port2/Switch 1 sends an RST BPDU with an agreed flag to Port2/Switch 2 and Port2/Switch 1 goes into a forwarding state.

Port3/Switch 1 receives an RST BPDUs that is inferior to what it can transmit; therefore, the port retains its Designated port role and goes into forwarding state only after the forward delay timer expires twice on that port while it is still in a Designated role.

Port3/Switch 2 sends an RST BPDU to Port3/Switch 3 that contains a proposal flag. Port3/Switch 3 becomes the Root port, while all other ports on Switch 3 are given Designated port roles and go into discarding states. Port3/Switch 3 sends an RST BPDU with an agreed flag to Port3/Switch 2 and Port3/Switch 3 goes into a forwarding state.

Now, Port2/Switch 3 receives an RST BPDUs that is superior to what it can transmit so that port is given an Alternate port state.

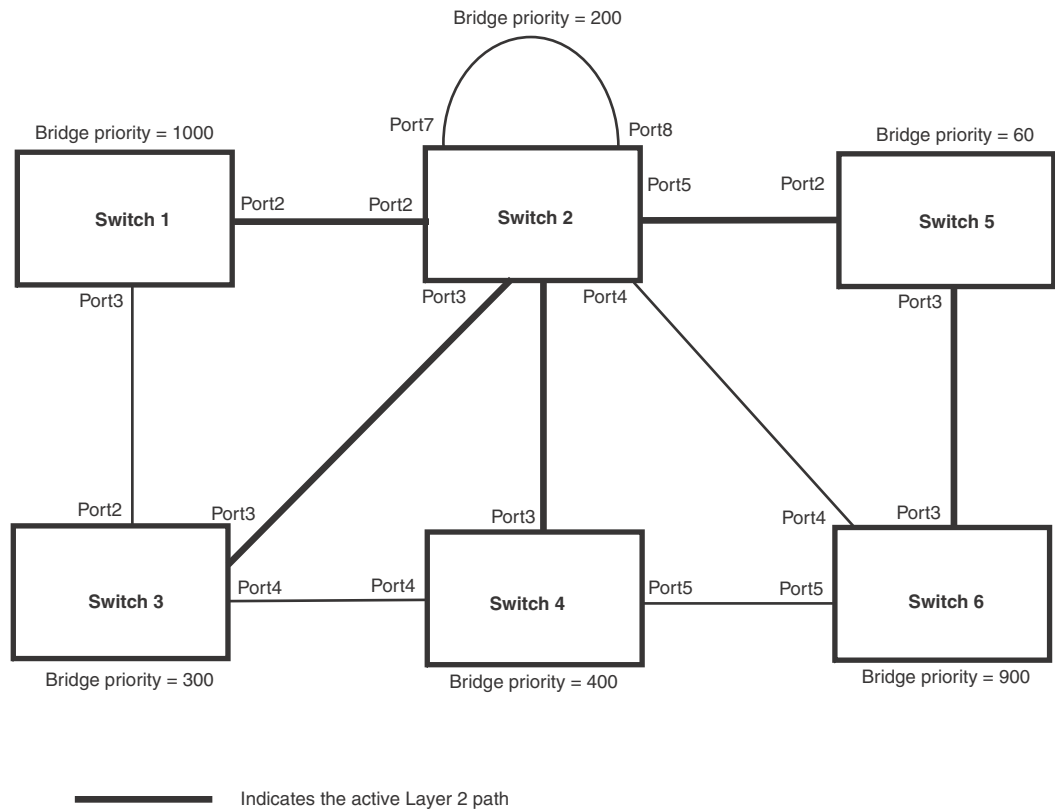
Port4/Switch 3 receives an RST BPDU that is inferior to what it can transmit; therefore, the port retains its Designated port role.

Ports on all the bridges in the topology with Designated port roles that received RST BPDUs with agreed flags go into forwarding states instantly. However, Designated ports that did not receive RST BPDUs with agreed flags must wait until the forward delay timer expires twice on those port. Only then will these port move into forwarding states.

The entire 802.1W topology converges in less than 300 msec and the essential connectivity is established between the designated ports and their connected root ports.

After convergence is complete, [Figure 144](#) shows the active Layer 2 path of the topology in [Figure 143](#).

FIGURE 144 Active Layer 2 path in complex topology



Propagation of topology change

The Topology Change state machine generates and propagates the topology change notification messages on each port. When a Root port or a Designated port goes into a forwarding state, the Topology Change state machine on those ports send a topology change notice (TCN) to all the bridges in the topology to propagate the topology change.

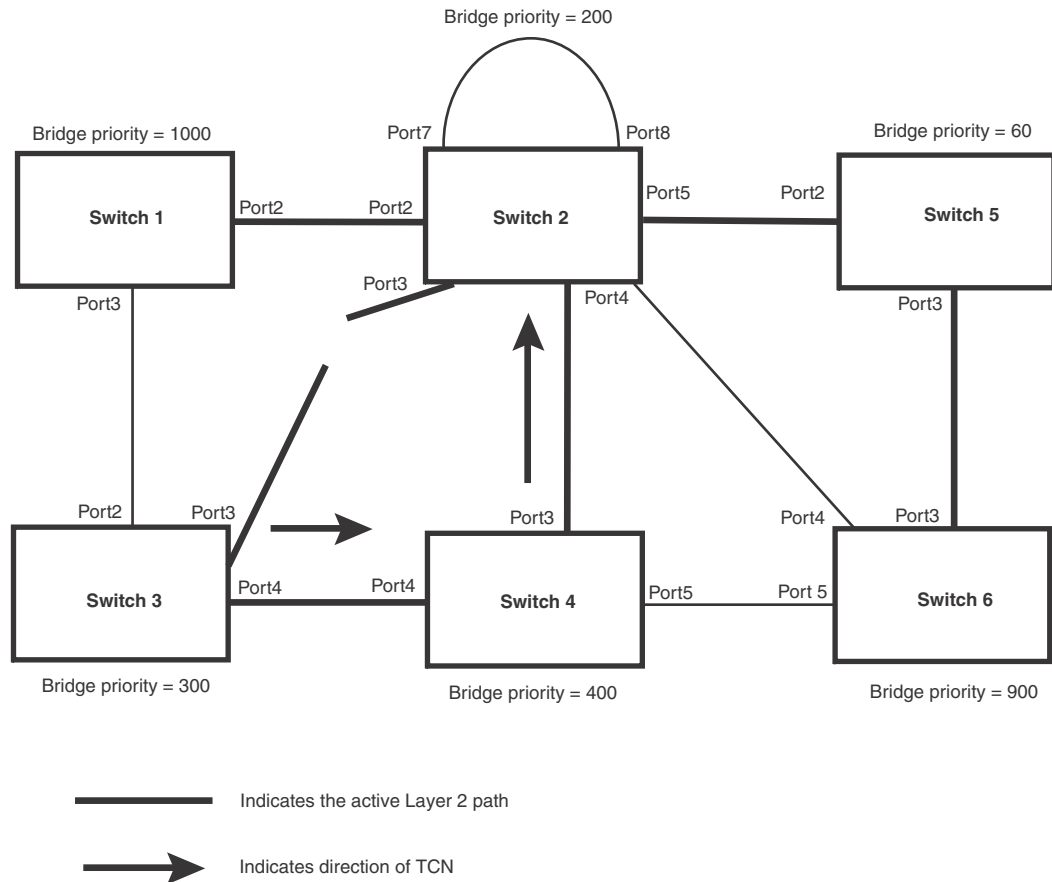
NOTE

Edge ports, Alternate ports, or Backup ports do not need to propagate a topology change.

The TCN is sent in the RST BPDU that a port sends. Ports on other bridges in the topology then acknowledge the topology change once they receive the RST BPDU, and send the TCN to other bridges until all the bridges are informed of the topology change.

For example, Port3/Switch 2 in [Figure 145](#), fails. Port4/Switch 3 becomes the new Root port. Port4/Switch 3 sends an RST BPDU with a TCN to Port4/Switch 4. To propagate the topology change, Port4/Switch 4 then starts a TCN timer on itself, on the bridge Root port, and on other ports on that bridge with a Designated role. Then Port3/Switch 4 sends RST BPDU with the TCN to Port4/Switch 2. (Note the new active Layer 2 path in [Figure 145](#).)

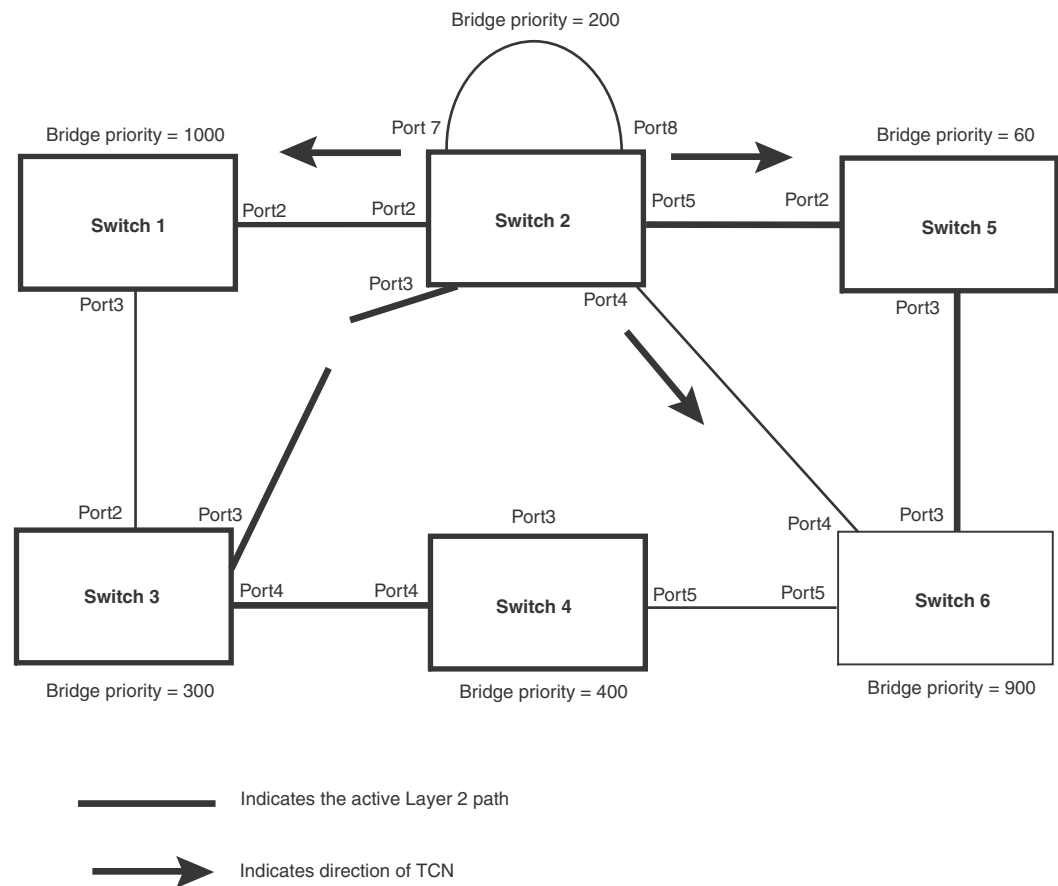
FIGURE 145 Beginning of topology change notice



Switch 2 then starts the TCN timer on the Designated ports and sends RST BPDUs that contain the TCN as follows ([Figure 146](#)):

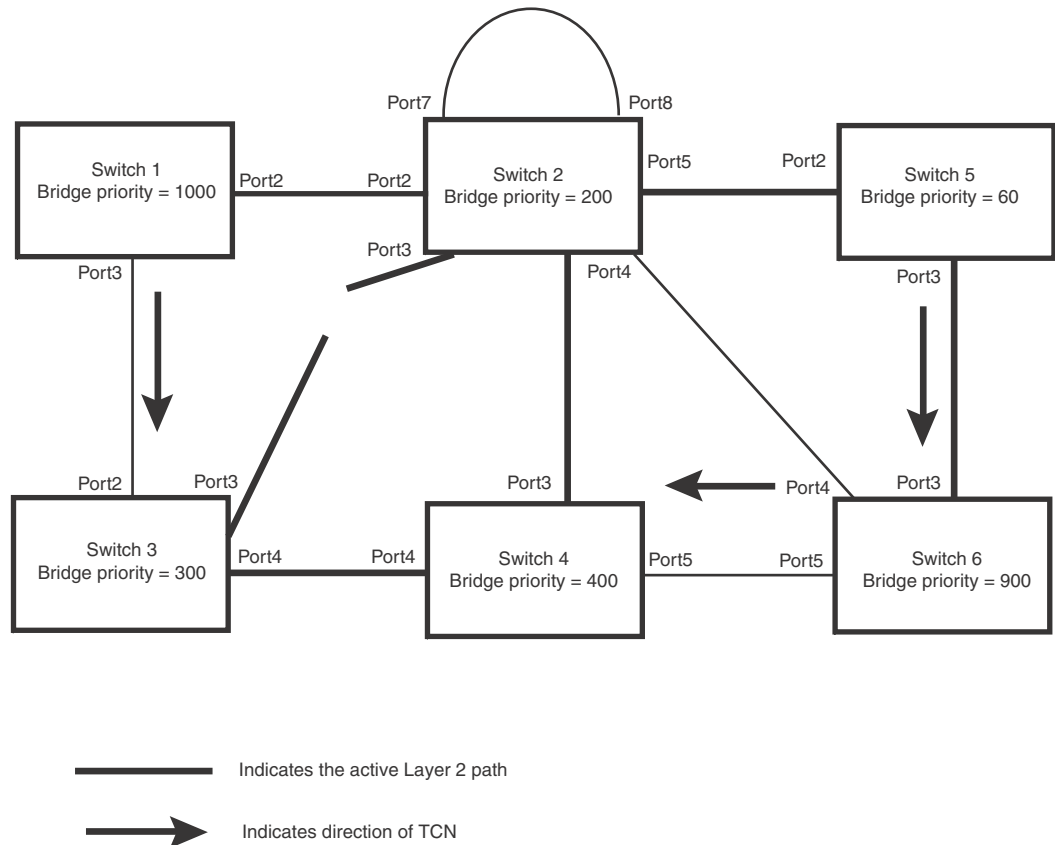
- Port5/Switch 2 sends the TCN to Port2/Switch 5
- Port4/Switch 2 sends the TCN to Port4/Switch 6
- Port2/Switch 2 sends the TCN to Port2/Switch 1

FIGURE 146 Sending TCN to bridges connected to Switch 2



Then Switch 1, Switch 5, and Switch 6 send RST BPDUs that contain the TCN to Switch 3 and Switch 4 to complete the TCN propagation (Figure 147).

FIGURE 147 Completing the TCN propagation



Compatibility of 802.1W with 802.1D

802.1W-enabled bridges are backward compatible with IEEE 802.1D bridges. This compatibility is managed on a per-port basis by the Port Migration state machine. **However, intermixing the two types of bridges in the network topology is not advisable if you want to take advantage of the rapid convergence feature.**

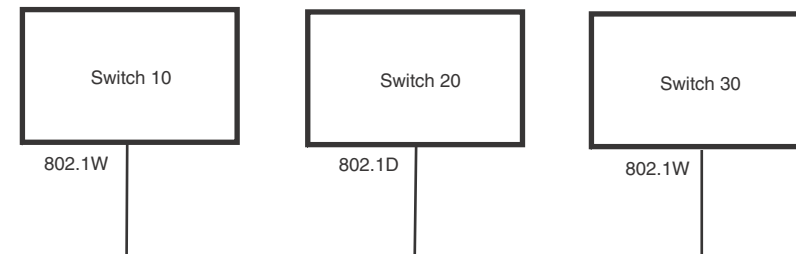
Compatibility with 802.1D means that an 802.1W-enabled port can send BPDUs in the STP or 802.1D format when one of the following events occur:

- The port receives a legacy BPDU. A legacy BPDU is an STP BPDU or a BPDU in an 802.1D format. The port that receives the legacy BPDU automatically configures itself to behave like a legacy port. It sends and receives legacy BPDUs only.
- The entire bridge is configured to operate in an 802.1D mode when an administrator sets the bridge parameter to zero at the CLI, forcing all ports on the bridge to send legacy BPDUs only.

Once a port operates in the 802.1D mode, 802.1D convergence times are used and rapid convergence is not realized.

For example, in [Figure 148](#), Switch 10 and Switch 30 receive legacy BPDUs from Switch 20. Ports on Switch 10 and Switch 30 begin sending BPDUs in STP format to allow them to operate transparently with Switch 20.

FIGURE 148 802.1W bridges with an 802.1D bridge



Once Switch 20 is removed from the LAN, Switch 10 and Switch 30 receive and transmit BPDUs in the STP format to and from each other. This state will continue until the administrator enables the **force-migration-check** command to force the bridge to send RSTP BPDU during a migrate time period. If ports on the bridges continue to hear only STP BPDUs after this migrate time period, those ports will return to sending STP BPDUs. However, when the ports receive RST BPDUs during the migrate time period, the ports begin sending RST BPDUs. The migrate time period is non-configurable. It has a value of three seconds.

NOTE

The IEEE standards state that 802.1W bridges need to interoperate with 802.1D bridges. IEEE standards set the path cost of 802.1W bridges to be between 1 and 200,000,000; whereas path cost of 802.1D bridges are set between 1 and 65,535. In order for the two bridge types to be able to interoperate in the same topology, the administrator needs to configure the bridge path cost appropriately. Path costs for either 802.1W bridges or 802.1D bridges need to be changed; in most cases, path costs for 802.1W bridges need to be changed.

Configuring 802.1W parameters on a Brocade device

The remaining 802.1W sections explain how to configure the 802.1W protocol in a Brocade device.

NOTE

With RSTP running, enabling static trunk on ports that are members of VLAN 4000 will keep the system busy for 20 to 25 seconds.

Brocade devices are shipped from the factory with 802.1W disabled. Use the following methods to enable or disable 802.1W. You can enable or disable 802.1W at the following levels:

- **Port-based VLAN** – Affects all ports within the specified port-based VLAN. When you enable or disable 802.1W within a port-based VLAN, the setting overrides the global setting. Thus, you can enable 802.1W for the ports within a port-based VLAN even when 802.1W is globally disabled, or disable the ports within a port-based VLAN when 802.1W is globally enabled.
- **Individual port** – Affects only the individual port. However, if you change the 802.1W state of the primary port in a trunk group, the change affects all ports in the trunk group.

Enabling or disabling 802.1W in a port-based VLAN

Use the following procedure to disable or enable 802.1W on a device on which you have configured a port-based VLAN. Changing the 802.1W state in a VLAN affects only that VLAN.

To enable 802.1W for all ports in a port-based VLAN, enter commands such as the following.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#spanning-tree 802-1w
```

Syntax: [no] spanning-tree 802-1w

Note regarding pasting 802.1W settings into the running configuration

If you paste 802.1W settings into the running configuration, and the pasted configuration includes ports that are already up, the ports will initially operate in STP legacy mode before operating in 802.1W RSTP mode. For example, the following pasted configuration will cause ports e 1 and e 2 to temporarily operate in STP legacy mode, because these ports are already up and running.

```
conf t
vlan 120
tag e 1 to e 2
spanning-tree 802-1w
spanning-tree 802-1w priority 1001
end
```

To avoid this issue, 802.1W commands/settings that are pasted into the configuration should be in the following order.

1. Ports that are not yet connected
2. 802.1W RSTP settings
3. Ports that are already up

Example

```
conf t
vlan 120
untag e 3
spanning-tree 802-1w
spanning-tree 802-1w priority 1001
tag e 1 to 2
end
```

In the above configuration, untagged port e3 is added to VLAN 120 *before* the 802.1W RSTP settings, and ports e1 and e2 are added *after* the 802.1W RSTP settings. When these commands are pasted into the running configuration, the ports will properly operate in 802.1W RSTP mode.

Enabling or disabling 802.1W on a single spanning tree

To enable 802.1W for all ports of a single spanning tree, enter a command such as the following.

```
Brocade(config-vlan-10)#spanning-tree single 802-1w
```

Syntax: [no] spanning-tree single 802-1w

Disabling or enabling 802.1W on an individual port

The **spanning-tree 802-1w** or **spanning-tree single 802-1w** command must be used to initially enable 802.1W on ports. Both commands enable 802.1W on all ports that belong to the VLAN or to the single spanning tree.

Once 802.1W is enabled on a port, it can be disabled on individual ports. 802.1W that have been disabled on individual ports can then be enabled as required.

NOTE

If you change the 802.1W state of the primary port in a trunk group, the change affects all ports in that trunk group.

To disable or enable 802.1W on an individual port, enter commands such as the following.

```
Brocade(config)#interface e 1
Brocade(config-if-e1000-1)#no spanning-tree
```

Syntax: [no] spanning-tree

Changing 802.1W bridge parameters

When you make changes to 802.1W bridge parameters, the changes are applied to individual ports on the bridge. To change 802.1W bridge parameters, use the following methods.

To designate a priority for a bridge, enter a command such as the following.

```
Brocade(config)#spanning-tree 802-1w priority 10
```

The command in this example changes the priority on a device on which you have not configured port-based VLANs. The change applies to the default VLAN. If you have configured a port-based VLAN on the device, you can configure the parameters only at the configuration level for individual VLANs. Enter commands such as the following.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#spanning-tree 802-1w priority 0
```

To make this change in the default VLAN, enter the following commands.

```
Brocade(config)#vlan 1
Brocade(config-vlan-1)#spanning-tree 802-1w priority 0
```

Syntax: spanning-tree 802-1w [forward-delay <value>] | [hello-time <value>] | [max-age <time>] | [force-version <value>] | [priority <value>]

The **forward-delay** <value> parameter specifies how long a port waits before it forwards an RST BPDUs after a topology change. This can be a value from 4 – 30 seconds. The default is 15 seconds.

The **hello-time** <value> parameter specifies the interval between two hello packets. This parameter can have a value from 1 – 10 seconds. The default is 2 seconds.

The **max-age** <value> parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 – 40 seconds. The default is 20 seconds.

The value of **max-age** must be greater than the value of **forward-delay** to ensure that the downstream bridges do not age out faster than the upstream bridges (those bridges that are closer to the root bridge).

The **force-version** <value> parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following values:

- **0** – The STP compatibility mode. Only STP (or legacy) BPDUs will be sent.
- **2** – The default. RST BPDUs will be sent unless a legacy bridge is detected. If a legacy bridge is detected, STP BPDUs will be sent instead.

The default is 2.

The **priority** *<value>* parameter specifies the priority of the bridge. You can enter a value from 0 – 65535. A lower numerical value means the bridge has a higher priority. Thus, the highest priority is 0. The default is 32768.

You can specify some or all of these parameters on the same command line. If you specify more than one parameter, you must specify them in the order shown above, from left to right.

Changing port parameters

The 802.1W port commands can be enabled on individual ports or on multiple ports, such as all ports that belong to a VLAN.

The 802.1W port parameters are preconfigured with default values. If the default parameters meet your network requirements, no other action is required.

You can change the following 802.1W port parameters using the following method.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#spanning-tree 802-1w ethernet 5 path-cost 15 priority 64
```

Syntax: **spanning-tree 802-1w ethernet** *<port>* **path-cost** *<value>* | **priority** *<value>* | **[admin-edge-port]** | **[admin-pt2pt-mac]** | **[force-migration-check]**

The **ethernet** *<port>* parameter specifies the interface used. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **path-cost** *<value>* parameter specifies the cost of the port path to the root bridge. 802.1W prefers the path with the lowest cost. You can specify a value from 1 – 20,000,000. [Table 198](#) shows the recommended path cost values from the IEEE standards.

TABLE 198 Recommended path cost values of 802.1W

Link speed	Recommended (Default) 802.1W path cost values	Recommended 802.1W patch cost range
Less than 100 kilobits per second	200,000,000	20,000,000 – 200,000,000
1 Megabit per second	20,000,000	2,000,000 – 200,000,000
10 Megabits per second	2,000,000	200,000 – 200,000,000
100 Megabits per second	200,000	20,000 – 200,000,000
1 Gbps per second	20,000	2,000 – 200,000,000
10 Gbps per second	2,000	200 – 20,000
100 Gbps per second	200	20 – 2,000
1 Terabits per second	20	2 – 200
10 Terabits per second	2	1 – 20

- The **priority** *<value>* parameter specifies the preference that 802.1W gives to this port relative to other ports for forwarding traffic out of the topology. You can specify a value from 0 – 240, in increments of 16. If you enter a value that is not divisible by 16, the software returns an error message. The default value is 128. A higher numerical value means a lower priority; thus, the highest priority is 0.
- Set the **admin-edge-port** to enabled or disabled. If set to enabled, then the port becomes an edge port in the domain.

Set the **admin-pt2pt-mac** to enabled or disabled. If set to enabled, then a port is connected to another port through a point-to-point link. The point-to-point link increases the speed of convergence. This parameter, however, does not auto-detect whether or not the link is a physical point-to-point link.

The **force-migration-check** parameter forces the specified port to send one RST BPDU. If only STP BPDUs are received in response to the sent RST BPDU, then the port will go return to sending STP BPDUs.

Example

Suppose you want to enable 802.1W on a system with no active port-based VLANs and change the hello-time from the default value of 2 to 8 seconds. Additionally, suppose you want to change the path and priority costs for port 5 only. To do so, enter the following commands.

```
Brocade(config)#spanning-tree 802-1w hello-time 8
Brocade(config)#spanning-tree 802-1w ethernet 5 path-cost 15 priority 64
```

Displaying information about 802-1W

To display a summary of 802-1W, use the following command.

```
Brocade#show 802-1w
--- VLAN 1 [ STP Instance owned by VLAN 1 ] -----
VLAN 1 BPDU cam_index is 2 and the IGC and DMA master Are(HEX) 0 1 2 3
Bridge IEEE 802.1W Parameters:
Bridge          Bridge      Bridge      Bridge      Force      tx
Identifier      MaxAge    Hello      FwdDly    Version    Hold
hex            sec       sec       sec       cnt
800000e080541700 20        2         15        Default 3

RootBridge      RootPath    DesignatedBri-      Root      Max      Fwd      Hel
Identifier      Cost        dge Identifier      Port      Age      Dly      lo
hex            hex         hex
800000e0804c9c00 200000      800000e0804c9c00 1         20      15      2

Port IEEE 802.1W Parameters:
      <--- Config Params -->|<----- Current state ----->
Port  Pri PortPath P2P Edge Role      State      Designa- Designated
Num    Cost  Mac  Port  Role      State      ted cost  bridge
1      128 200000 F  F  ROOT      FORWARDING 0      800000e0804c9c00
2      128 200000 F  F  DESIGNATED FORWARDING 200000 800000e080541700
3      128 200000 F  F  DESIGNATED FORWARDING 200000 800000e080541700
4      128 200000 F  F  BACKUP     DISCARDING 200000 800000e080541700
```

Syntax: `show 802-1w [vlan <vlan-id>]`

The **vlan** *<vlan-id>* parameter displays 802.1W information for the specified port-based VLAN.

The **show 802.1w** command shows the information listed in [Table 199](#).

TABLE 199 CLI display of 802.1W summary

Field	Description
VLAN ID	The port-based VLAN that owns the STP instance. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all 802.1W information is for VLAN 1.
Bridge IEEE 802.1W parameters	
Bridge Identifier	The ID of the bridge.
Bridge Max Age	The configured max age for this bridge. The default is 20.
Bridge Hello	The configured hello time for this bridge. The default is 2.
Bridge FwdDly	The configured forward delay time for this bridge. The default is 15.
Force-Version	The configured force version value. One of the following value is displayed: <ul style="list-style-type: none"> • 0 – The bridge has been forced to operate in an STP compatibility mode. • 2 – The bridge has been forced to operate in an 802.1W mode. (This is the default.)
txHoldCnt	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Root Bridge Identifier	ID of the Root bridge that is associated with this bridge
Root Path Cost	The cost to reach the root bridge from this bridge. If the bridge is the root bridge, then this parameter shows a value of zero.
Designated Bridge Identifier	The bridge from where the root information was received. It can be from the root bridge itself, but it could also be from another bridge.
Root Port	The port on which the root information was received. This is the port that is connected to the Designated Bridge.
Max Age	<p>The max age is derived from the Root port. An 802.1W-enabled bridge uses this value, along with the hello and message age parameters to compute the effective age of an RST BPDU.</p> <p>The message age parameter is generated by the Designated port and transmitted in the RST BPDU. RST BPDUs transmitted by a Designated port of the root bridge contains a message value of zero.</p> <p>Effective age is the amount of time the Root port, Alternate port, or Backup port retains the information it received from its peer Designated port. Effective age is reset every time a port receives an RST BPDU from its Designated port. If a Root port does not receive an RST BPDU from its peer Designated port for a duration more than the effective age, the Root port ages out the existing information and recomputes the topology.</p> <p>If the port is operating in 802.1D compatible mode, then max age functionality is the same as in 802.1D (STP).</p>

TABLE 199 CLI display of 802.1W summary (Continued)

Field	Description
Fwd Dly	<p>The number of seconds a non-edge Designated port waits until it can apply any of the following transitions, if the RST BPDU it receives does not have an agreed flag:</p> <ul style="list-style-type: none"> Discarding state to learning state Learning state to forwarding state <p>When a non-edge port receives the RST BPDU it goes into forwarding state within 4 seconds or after two hello timers expire on the port. Fwd Dly is also the number of seconds that a Root port waits for an RST BPDU with a proposal flag before it applies the state transitions listed above.</p> <p>If the port is operating in 802.1D compatible mode, then forward delay functionality is the same as in 802.1D (STP).</p>
Hello	The hello value derived from the Root port. It is the number of seconds between two Hello packets.
Port IEEE 802.1W parameters	
Port Num	The port number shown in a slot#/port# format.
Pri	The configured priority of the port. The default is 128 or 0x80.
Port Path Cost	The configured path cost on a link connected to this port.
P2P Mac	<p>Indicates if the point-to-point-mac parameter is configured to be a point-to-point link:</p> <ul style="list-style-type: none"> T – The link is configured as a point-to-point link. F – The link is not configured as a point-to-point link. This is the default.
Edge port	<p>Indicates if the port is configured as an operational Edge port:</p> <ul style="list-style-type: none"> T – The port is configured as an Edge port. F – The port is not configured as an Edge port. This is the default.
Role	<p>The current role of the port:</p> <ul style="list-style-type: none"> Root Designated Alternate Backup Disabled <p>Refer to “Bridges and bridge port roles” on page 1109 for definitions of the roles.</p>
State	<p>The port current 802.1W state. A port can have one of the following states:</p> <ul style="list-style-type: none"> Forwarding Discarding Learning Disabled <p>Refer to “Bridge port states” on page 1113 and “Edge port and non-edge port states” on page 1114.</p>
Designated Cost	The best root path cost that this port received, including the best root path cost that it can transmit.
Designated Bridge	The ID of the bridge that sent the best RST BPDU that was received on this port.

To display detailed information about 802-1W, enter the **802-1w** command.

```
Brocade#show 802-1w detail
=====
VLAN 1 - MULTIPLE SPANNING TREE (MSTP - IEEE 802.1W) ACTIVE
=====
BridgeId 800000e080541700, forceVersion 2, txHoldCount 3
Port 1 - Role: ROOT - State: FORWARDING
  PathCost 200000, Priority 128, AdminOperEdge F, AdminPt2PtMac F
  DesignatedPriority - Root: 0x800000e0804c9c00, Bridge: 0x800000e080541700
  ActiveTimers - rrWhile 4 rcvdInfoWhile 4
  MachineStates - PIM: CURRENT, PRT: ROOT_PORT, PST: FORWARDING
  TCM: ACTIVE, PPM: SENDING_STP, PTX: TRANSMIT_IDLE
  Received - RST BPDUs 0, Config BPDUs 1017, TCN BPDUs 0

Port 2 - Role: DESIGNATED - State: FORWARDING
  PathCost 200000, Priority 128, AdminOperEdge F, AdminPt2PtMac F
  DesignatedPriority - Root: 0x800000e0804c9c00, Bridge: 0x800000e080541700
  ActiveTimers - helloWhen 0
  MachineStates - PIM: CURRENT, PRT: DESIGNATED_PORT, PST: FORWARDING
  TCM: ACTIVE, PPM: SENDING_RSTP, PTX: TRANSMIT_IDLE
  Received - RST BPDUs 0, Config BPDUs 0, TCN BPDUs 0
```

Syntax: `show 802-1w detail [vlan <vlan-id>]`

The **vlan <vlan-id>** parameter displays 802.1W information for the specified port-based VLAN.

The **show spanning-tree 802.1W** command shows the following information.

TABLE 200 CLI display of show spanning-tree 802.1W

Field	Description
VLAN ID	ID of the VLAN that owns the instance of 802.1W and whether or not it is active.
Bridge ID	ID of the bridge.
forceVersion	the configured version of the bridge: <ul style="list-style-type: none"> • 0 – The bridge has been forced to operate in an STP compatible mode. • 2 – The bridge has been forced to operate in an 802.1W mode.
txHoldCount	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Port	ID of the port in slot#/port#format.
Role	The current role of the port: <ul style="list-style-type: none"> • Root • Designated • Alternate • Backup • Disabled Refer to “Bridges and bridge port roles” on page 1109 for definitions of the roles.

TABLE 200 CLI display of show spanning-tree 802.1W (Continued)

Field	Description
State	<p>The port current 802.1W state. A port can have one of the following states:</p> <ul style="list-style-type: none"> • Forwarding • Discarding • Learning • Disabled <p>Refer to “Bridge port states” on page 1113 and “Edge port and non-edge port states” on page 1114.</p>
Path Cost	The configured path cost on a link connected to this port.
Priority	The configured priority of the port. The default is 128 or 0x80.
AdminOperEdge	<p>Indicates if the port is an operational Edge port. Edge ports may either be auto-detected or configured (forced) to be Edge ports using the CLI:</p> <ul style="list-style-type: none"> • T – The port is and Edge port. • F – The port is not an Edge port. This is the default.
AdminP2PMac	<p>Indicates if the point-to-point-mac parameter is configured to be a point-to-point link:</p> <ul style="list-style-type: none"> • T – The link is a point-to-point link • F – The link is not a point-to-point link. This is the default.
DesignatedPriority	<p>Shows the following:</p> <ul style="list-style-type: none"> • Root – Shows the ID of the root bridge for this bridge. • Bridge – Shows the ID of the Designated bridge that is associated with this port.
ActiveTimers	<p>Shows what timers are currently active on this port and the number of seconds they have before they expire:</p> <ul style="list-style-type: none"> • rrWhile – Recent root timer. A non-zero value means that the port has recently been a Root port. • rcvdInfoWhile – Received information timer. Shows the time remaining before the information held by this port expires (ages out). This timer is initialized with the effective age parameter. (Refer to “Max Age” on page 1142.) • rbWhile – Recent backup timer. A non-zero value means that the port has recently been a Backup port. • helloWhen – Hello period timer. The value shown is the amount of time between hello messages. • tcWhile – Topology change timer. The value shown is the interval when topology change notices can be propagated on this port. • fdWhile – Forward delay timer. • mdelayWhile – Migration delay timer. The amount of time that a bridge on the same LAN has to synchronize its migration state with this port before another BPDU type can cause this port to change the BPDU that it transmits.

TABLE 200 CLI display of show spanning-tree 802.1W (Continued)

Field	Description
Machine States	<p>The current states of the various state machines on the port:</p> <ul style="list-style-type: none"> • PIM – State of the Port Information state machine. • PRT – State of the Port Role Transition state machine. • PST – State of the Port State Transition state machine. • TCM – State of the Topology Change state machine. • PPM – State of the Port Protocol Migration. • PTX – State of the Port Transmit state machine. <p>Refer to the section “Port Role Selection state machines” on page 1114 for details on state machines.</p>
Received	<p>Shows the number of BPDU types the port has received:</p> <ul style="list-style-type: none"> • RST BPDU – BPDU in 802.1W format. • Config BPDU – Legacy configuration BPDU (802.1D format). • TCN BPDU – Legacy topology change BPDU (802.1D format).

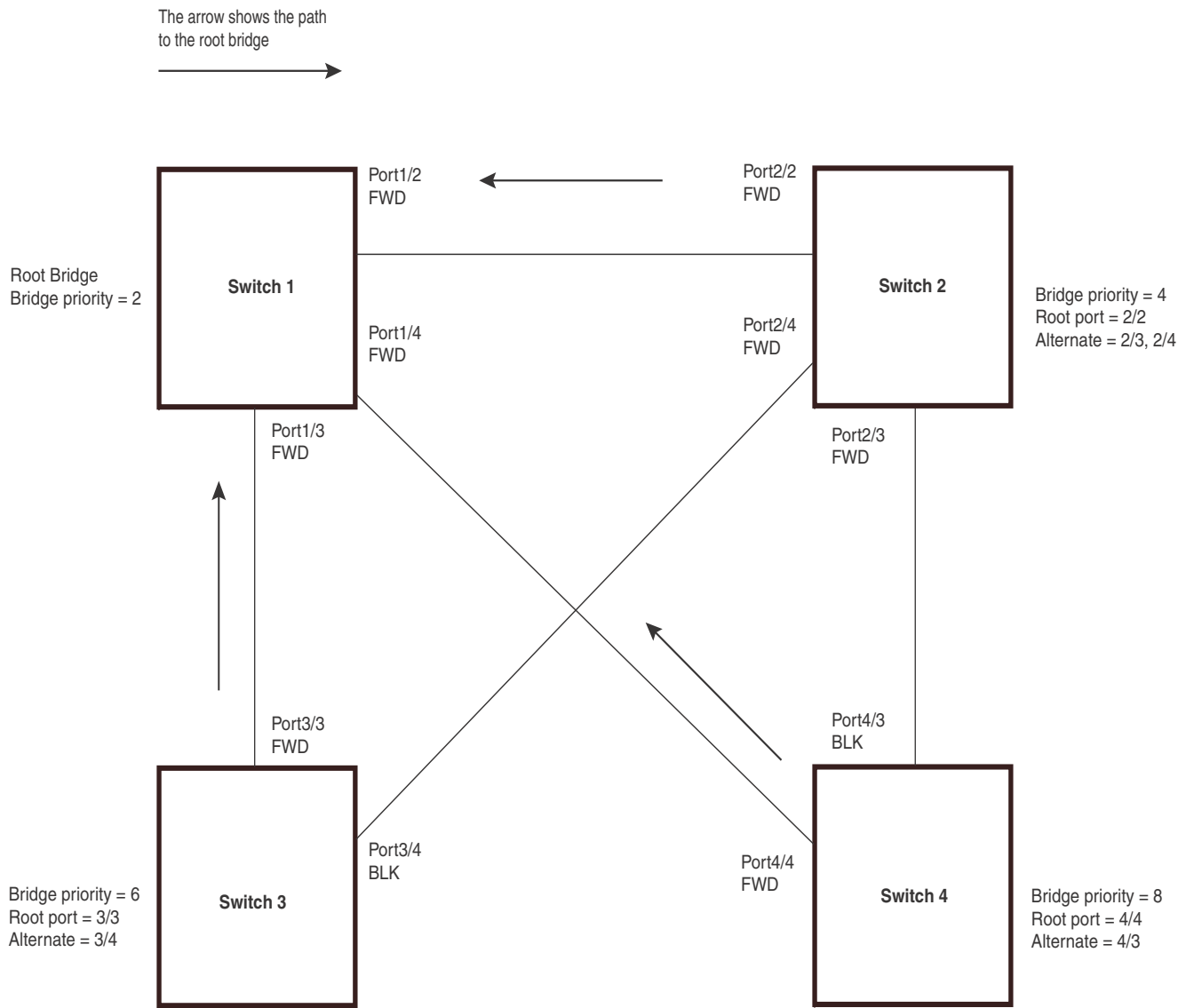
802.1W Draft 3

As an alternative to full 802.1W, you can configure 802.1W Draft 3. 802.1W Draft 3 provides a subset of the RSTP capabilities described in the 802.1W STP specification.

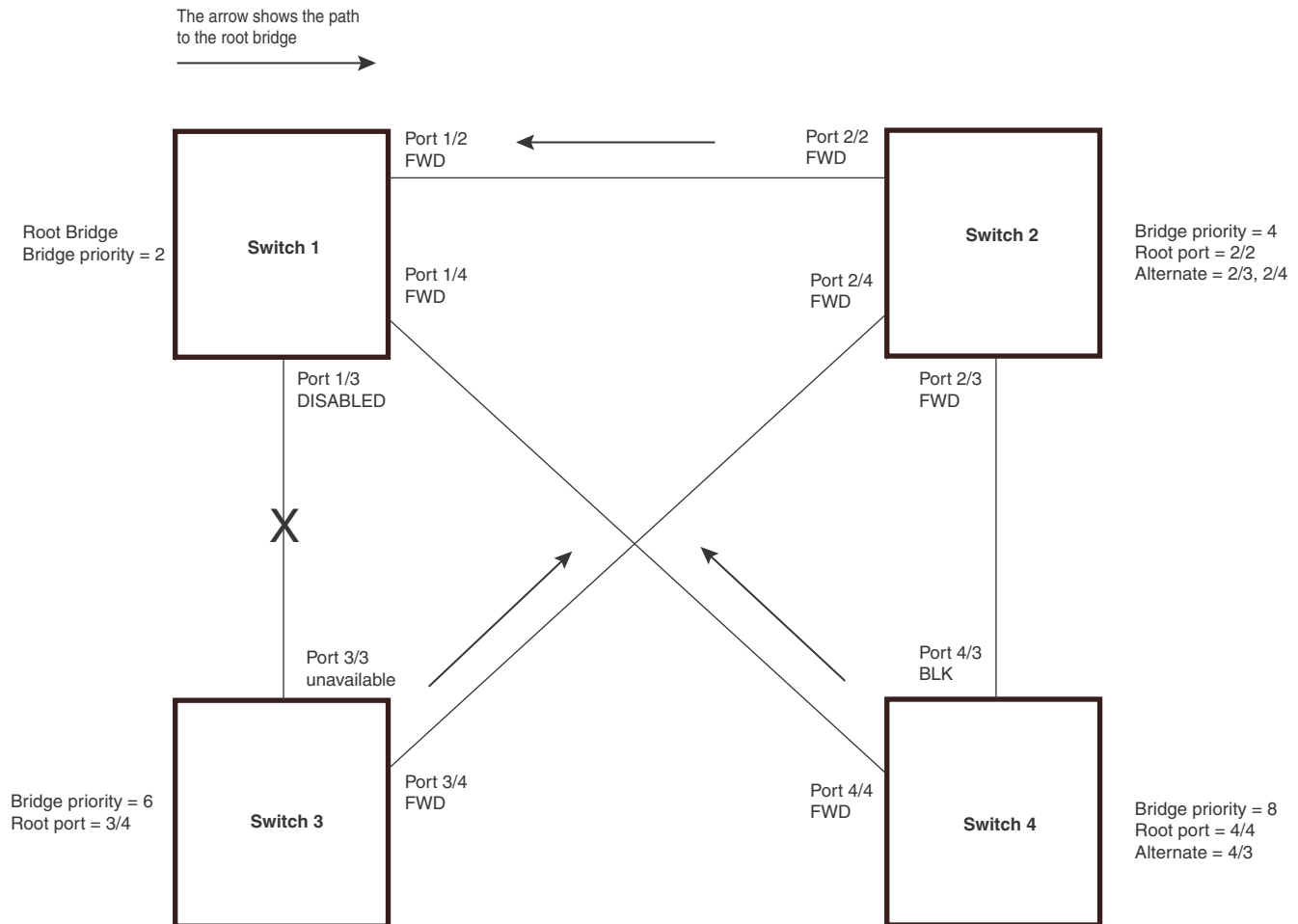
802.1W Draft 3 support is disabled by default. When the feature is enabled, if a root port on a Brocade device that is not the root bridge becomes unavailable, the device can automatically Switch over to an alternate root port, without reconvergence delays. 802.1W Draft 3 does not apply to the root bridge, since all the root bridge ports are always in the forwarding state.

[Figure 149](#) shows an example of an optimal STP topology. In this topology, all the non-root bridges have at least two paths to the root bridge (Switch 1 in this example). One of the paths is through the root port. The other path is a backup and is through the alternate port. While the root port is in the forwarding state, the alternate port is in the blocking state.

FIGURE 149 802.1W Draft 3 RSTP ready for failover



If the root port on a Switch becomes unavailable, 802.1W Draft 3 immediately fails over to the alternate port, as shown in [Figure 150](#).

FIGURE 150 802.1W Draft 3 RSTP failover to alternate root port

In this example, port 3/3 on Switch 3 has become unavailable. In standard STP (802.1D), if the root port becomes unavailable, the Switch must go through the listening and learning stages on the alternate port to reconverge with the spanning tree. Thus, port 3/4 must go through the listening and learning states before entering the forwarding state and thus reconverging with the spanning tree.

802.1W Draft 3 avoids the reconvergence delay by calculating an alternate root port, and immediately failing over to the alternate port if the root port becomes unavailable. The alternate port is in the blocking state as long as the root port is in the forwarding state, but moves immediately to the active state if the root port becomes unavailable. Thus, using 802.1W Draft 3, Switch 3 immediately fails over to port 3/4, without the delays caused by the listening and learning states.

802.1W Draft 3 selects the port with the next-best cost to the root bridge. For example, on Switch 3, port 3/3 has the best cost to the root bridge and thus is selected by STP as the root port. Port 3/4 has the next-best cost to the root bridge, and thus is selected by 802.1W Draft 3 as the alternate path to the root bridge.

Once a failover occurs, the Switch no longer has an alternate root port. If the port that was an alternate port but became the root port fails, standard STP is used to reconverge with the network. You can minimize the reconvergence delay in this case by setting the forwarding delay on the root bridge to a lower value. For example, if the forwarding delay is set to 15 seconds (the default), change the forwarding delay to a value from 3 – 10 seconds.

During failover, 802.1W Draft 3 flushes the MAC addresses learned on the unavailable root port, selects the alternate port as the new root port, and places that port in the forwarding state. If traffic is flowing in both directions on the new root port, addresses are flushed (moved) in the rest of the spanning tree automatically.

Spanning tree reconvergence time

Spanning tree reconvergence using 802.1W Draft 3 can occur within one second.

After the spanning tree reconverges following the topology change, traffic also must reconverge on all the bridges attached to the spanning tree. This is true regardless of whether 802.1W Draft 3 or standard STP is used to reconverge the spanning tree.

Traffic reconvergence happens after the spanning tree reconvergence, and is achieved by flushing the Layer 2 information on the bridges:

- Following 802.1W Draft 3 reconvergence of the spanning tree, traffic reconvergence occurs in the time it takes for the bridge to detect the link changes plus the STP maximum age set on the bridge.
- If standard STP reconvergence occurs instead, traffic reconvergence takes two times the forward delay plus the maximum age.

NOTE

802.1W Draft 3 does not apply when a failed root port comes back up. When this happens, standard STP is used.

802.1W configuration considerations

802.1W Draft 3 is disabled by default. To ensure optimal performance of the feature before you enable it, do the following:

- Configure the bridge priorities so that the root bridge is one that supports 802.1W Draft 3. (Use a Brocade device or third-party device that supports 802.1W Draft 3.)
- Change the forwarding delay on the root bridge to a value lower than the default 15 seconds. Brocade recommends a value from 3 – 10 seconds. The lower forwarding delay helps reduce reconvergence delays in cases where 802.1W Draft 3 is not applicable, such as when a failed root port comes back up.
- Configure the bridge priorities and root port costs so that each device has an active path to the root bridge if its root port becomes unavailable. For example, port 3/4 is connected to port 2/4 on Switch 2, which has the second most favorable bridge priority in the spanning tree.

NOTE

If reconvergence involves changing the state of a root port on a bridge that supports 802.1D STP but not 802.1W Draft 3, then reconvergence still requires the amount of time it takes for the ports on the 802.1D bridge to change state to forwarding (as needed), and receive BPDUs from the root bridge for the new topology.

Enabling 802.1W Draft 3

802.1W Draft 3 is disabled by default. The procedure for enabling the feature differs depending on whether single STP is enabled on the device.

NOTE

STP must be enabled before you can enable 802.1W Draft 3.

Enabling 802.1W Draft 3 when single STP is not enabled

By default, each port-based VLAN on the device has its own spanning tree. To enable 802.1W Draft 3 in a port-based VLAN, enter commands such as the following.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#spanning-tree rstp
```

Syntax: [no] spanning-tree rstp

This command enables 802.1W Draft 3. You must enter the command separately in each port-based VLAN in which you want to run 802.1W Draft 3.

NOTE

This command does not also enable STP. To enable STP, first enter the **spanning-tree** command without the **rstp** parameter. After you enable STP, enter the **spanning-tree rstp** command to enable 802.1W Draft 3.

To disable 802.1W Draft 3, enter the following command.

```
Brocade(config-vlan-10)#no spanning-tree rstp
```

Enabling 802.1W Draft 3 when single STP is enabled

To enable 802.1W Draft 3 on a device that is running single STP, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)#spanning-tree single rstp
```

Syntax: [no] spanning-tree single rstp

This command enables 802.1W Draft 3 on the whole device.

NOTE

This command does not also enable single STP. To enable single STP, first enter the **spanning-tree single** command without the **rstp** parameter. After you enable single STP, enter the **spanning-tree single rstp** command to enable 802.1W Draft 3.

To disable 802.1W Draft 3 on a device that is running single STP, enter the following command.

```
Brocade(config)#no spanning-tree single rstp
```

Single Spanning Tree (SSTP)

By default, each port-based VLAN on a Brocade device runs a separate spanning tree, which you can enable or disable on an individual VLAN basis.

Alternatively, you can configure a Brocade device to run a single spanning tree across all ports and VLANs on the device. The Single STP feature (SSTP) is especially useful for connecting a Brocade device to third-party devices that run a single spanning tree in accordance with the 802.1Q specification.

SSTP uses the same parameters, with the same value ranges and defaults, as the default STP support on Brocade devices. Refer to [“STP parameters and defaults”](#) on page 1088.

SSTP defaults

SSTP is disabled by default. When you enable the feature, all VLANs on which STP is enabled become members of a single spanning tree. All VLANs on which STP is disabled are excluded from the single spanning tree.

To add a VLAN to the single spanning tree, enable STP on that VLAN. To remove a VLAN from the single spanning tree, disable STP on that VLAN.

When you enable SSTP, all the ports that are in port-based VLANs with STP enabled become members of a single spanning tree domain. Thus, the ports share a single BPDU broadcast domain. The Brocade device places all the ports in a non-configurable VLAN, 4094, to implement the SSTP domain. However, this VLAN does not affect port membership in the port-based VLANs you have configured. Other broadcast traffic is still contained within the individual port-based VLANs. Therefore, you can use SSTP while still using your existing VLAN configurations without changing your network. In addition, SSTP does not affect 802.1Q tagging. Tagged and untagged ports alike can be members of the single spanning tree domain.

NOTE

When SSTP is enabled, the BPDUs on tagged ports go out untagged.

If you disable SSTP, all VLANs that were members of the single spanning tree run MSTP instead. In MSTP, each VLAN has its own spanning tree. VLANs that were not members of the single spanning tree were not enabled for STP. Therefore, STP remains disabled on those VLANs.

Enabling SSTP

To enable SSTP, use one of the following methods.

NOTE

If the device has only one port-based VLAN (the default VLAN), then the device is already running a single instance of STP. In this case, you do not need to enable SSTP. You need to enable SSTP only if the device contains more than one port-based VLAN and you want all the ports to be in the same STP broadcast domain.

To configure the Brocade device to run a single spanning tree, enter the following command at the global CONFIG level.

```
Brocade(config)#spanning-tree single
```

NOTE

If the device has only one port-based VLAN, the CLI command for enabling SSTP is not listed in the CLI. The command is listed only if you have configured a port-based VLAN.

To change a global STP parameter, enter a command such as the following at the global CONFIG level.

```
Brocade(config)# spanning-tree single priority 2
```

This command changes the STP priority for all ports to 2.

To change an STP parameter for a specific port, enter commands such as the following.

```
Brocade(config) spanning-tree single ethernet 1 priority 10
```

The commands shown above override the global setting for STP priority and set the priority to 10 for port 1/1.

Here is the syntax for the global STP parameters.

Syntax: [no] spanning-tree single [forward-delay <value>] [hello-time <value>] | [maximum-age <time>] | [priority <value>]

Here is the syntax for the STP port parameters.

Syntax: [no] spanning-tree single [ethernet <port> path-cost <value> | priority <value>]

NOTE

Both commands listed above are entered at the global CONFIG level.

Displaying SSTP information

To verify that SSTP is in effect, enter the following commands at any level of the CLI.

```
Brocade#show span
```

Syntax: show span [vlan <vlan-id>] | [pvst-mode] | [<num>] | [detail [vlan <vlan-id> [ethernet <port>] | <num>]]

The **vlan** <vlan-id> parameter displays STP information for the specified port-based VLAN.

The **pvst-mode** parameter displays STP information for the device Per VLAN Spanning Tree (PVST+) compatibility configuration. Refer to “[PVST/PVST+ compatibility](#)” on page 1157.

Specify the **port** variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The <num> parameter displays only the entries after the number you specify. For example, on a device with three port-based VLANs, if you enter 1, then information for the second and third VLANs is displayed, but information for the first VLAN is not displayed. Information is displayed according to VLAN number, in ascending order. The entry number is not the same as the VLAN number. For example, if you have port-based VLANs 1, 10, and 2024, then the command output has three STP entries. To display information for VLANs 10 and 2024 only, enter **show span 1**.

The **detail** parameter and its additional optional parameters display detailed information for individual ports. Refer to [“Displaying detailed STP information for each interface”](#) on page 1099.

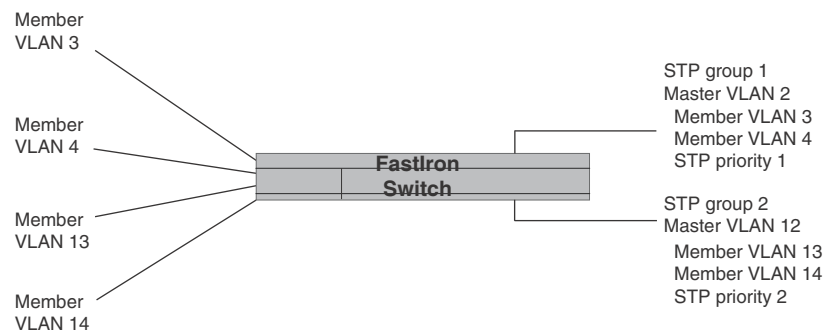
STP per VLAN group

STP per VLAN group is an STP enhancement that provides scalability while overcoming the limitations of the following scalability alternatives:

- **Standard STP** – You can configure up to 254 instances of standard STP on a Brocade device. It is possible to need more instances of STP than this in large configurations. Using STP per VLAN group, you can aggregate STP instances.
- **Single STP** – Single STP allows all the VLANs to run STP, but each VLAN runs the same instance of STP, resulting in numerous blocked ports that do not pass any Layer 2 traffic. STP per VLAN group uses all available links by load balancing traffic for different instances of STP on different ports. A port that blocks traffic for one spanning tree forwards traffic for another spanning tree.

STP per VLAN group allows you to group VLANs and apply the same STP parameter settings to all the VLANs in the group. [Figure 151](#) shows an example of a STP per VLAN group implementation.

FIGURE 151 STP per VLAN group example



A master VLAN contains one or more member VLANs. Each of the member VLANs in the STP Group runs the same instance of STP and uses the STP parameters configured for the master VLAN. In this example, the FastIron switch is configured with VLANs 3, 4, 13, and 14. VLANs 3 and 4 are grouped in master VLAN 2, which is in STP group 1. VLANs 13 and 14 are grouped in master VLAN 12, which is in STP group 2. The VLANs in STP group 1 all share the same spanning tree. The VLANs in STP group 2 share a different spanning tree.

All the ports are tagged. The ports must be tagged so that they can be in both a member VLAN and the member's master VLAN. For example, ports 1/1 – 1/4 are in member VLAN 3 and also in master VLAN 2 (since master VLAN 2 contains member VLAN 3).

STP load balancing

Notice that the STP groups each have different STP priorities. In configurations that use the STP groups on multiple devices, you can use the STP priorities to load balance the STP traffic. By setting the STP priorities for the same STP group to different values on each device, you can cause each of the devices to be the root bridge for a different STP group. This type of configuration distributes the traffic evenly across the devices and also ensures that ports that are blocked in one STP group spanning tree are used by another STP group spanning tree for forwarding. Refer to [“Configuration example for STP load sharing”](#) on page 1155 for an example using STP load sharing.

Configuring STP per VLAN group

To configure STP per VLAN group, perform the following tasks:

1. Configure the member VLANs.
2. Optionally, configure master VLANs to contain the member VLANs. This is useful when you have a lot of member VLANs and you do not want to individually configure STP on each one. Each of the member VLANs in the STP group uses the STP settings of the master VLAN.
3. Configure the STP groups. Each STP group runs a separate instance of STP.

The following CLI commands implement the STP per VLAN group configuration shown in [Figure 151](#). The following commands configure the member VLANs (3, 4, 13, and 14) and the master VLANs (2 and 12). Notice that changes to STP parameters are made in the master VLANs only, not in the member VLANs.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#spanning-tree priority 1
Brocade(config-vlan-2)#tagged ethernet 1/1 to 1/4
Brocade(config-vlan-2)#vlan 3
Brocade(config-vlan-3)#tagged ethernet 1/1 to 1/4
Brocade(config-vlan-3)#vlan 4
Brocade(config-vlan-4)#tagged ethernet 1/1 to 1/4
Brocade(config-vlan-4)#vlan 12
Brocade(config-vlan-12)#spanning-tree priority 2
Brocade(config-vlan-12)#tagged ethernet 1/1 to 1/4
Brocade(config-vlan-12)#vlan 13
Brocade(config-vlan-13)#tagged ethernet 1/1 to 1/4
Brocade(config-vlan-13)#vlan 14
Brocade(config-vlan-14)#tagged ethernet 1/1 to 1/4
Brocade(config-vlan-14)#exit
```

The following commands configure the STP groups.

```
Brocade(config)#stp-group 1
Brocade(config-stp-group-1)#master-vlan 2
Brocade(config-stp-group-1)#member-vlan 3 to 4
Brocade(config-stp-group-1)#exit
Brocade(config)#stp-group 2
Brocade(config-stp-group-2)#master-vlan 12
Brocade(config-stp-group-2)#member-vlan 13 to 14
```

Syntax: [no] stp-group <num>

This command changes the CLI to the STP group configuration level. The following commands are valid at this level. The <num> parameter specifies the STP group ID and can be from 1 – 32.

Syntax: `[no] master-vlan <num>`

This command adds a master VLAN to the STP group. The master VLAN contains the STP settings for all the VLANs in the STP per VLAN group. The `<num>` parameter specifies the VLAN ID. An STP group can contain one master VLAN.

If you delete the master VLAN from an STP group, the software automatically assigns the first member VLAN in the group to be the new master VLAN for the group.

Syntax: `[no] member-vlan <num> [to <num>]`

This command adds additional VLANs to the STP group. These VLANs also inherit the STP settings of the master VLAN in the group.

Syntax: `[no] member-group <num>`

This command adds a member group (a VLAN group) to the STP group. All the VLANs in the member group inherit the STP settings of the master VLAN in the group. The `<num>` parameter specifies the VLAN group ID.

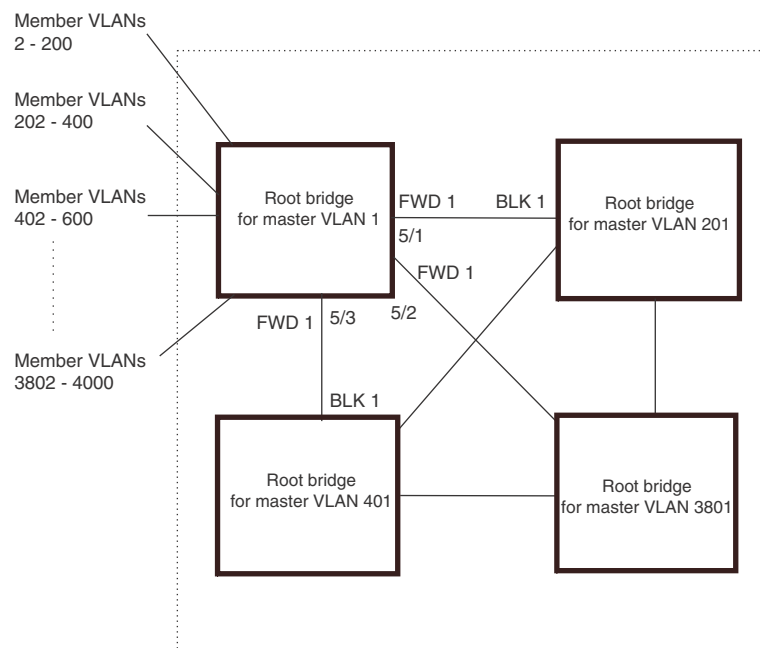
NOTE

This command is optional and is not used in the example above. For an example of this command, refer to [“Configuration example for STP load sharing”](#).

Configuration example for STP load sharing

Figure 152 shows another example of a STP per VLAN group implementation.

FIGURE 152 More complex STP per VLAN group example



In this example, each of the devices in the core is configured with a common set of master VLANs, each of which contains one or more member VLANs. Each of the member VLANs in an STP group runs the same instance of STP and uses the STP parameters configured for the master VLAN.

The STP group ID identifies the STP instance. All VLANs within an STP group run the same instance of STP. The master VLAN specifies the bridge STP parameters for the STP group, including the bridge priority. In this example, each of the devices in the core is configured to be the default root bridge for a different master VLAN. This configuration ensures that each link can be used for forwarding some traffic. For example, all the ports on the root bridge for master VLAN 1 are configured to forward BPDUs for master VLAN spanning tree. Ports on the other devices block or forward VLAN 1 traffic based on STP convergence. All the ports on the root bridge for VLAN 2 forward VLAN 2 traffic, and so on.

All the ports are tagged. The ports must be tagged so that they can be in both a member VLAN and the member's master VLAN. For example, port 1/1 – and ports 5/1, 5/2, and 5/3 are in member VLAN 2 and master VLAN 1 (since master VLAN 1 contains member VLAN 2).

Here are the commands for configuring the root bridge for master VLAN 1 in figure [Figure 151](#) for STP per VLAN group. The first group of commands configures the master VLANs. Notice that the STP priority is set to a different value for each VLAN. In addition, the same VLAN has a different STP priority on each device. This provides load balancing by making each of the devices a root bridge for a different spanning tree.

```
Brocade(config)#vlan 1
Brocade(config-vlan-1)#spanning-tree priority 1
Brocade(config-vlan-1)#tag ethernet 1/1 ethernet 5/1 to 5/3
Brocade(config-vlan-1)#vlan 201
Brocade(config-vlan-201)#spanning-tree priority 2
Brocade(config-vlan-201)#tag ethernet 1/2 ethernet 5/1 to 5/3
Brocade(config-vlan-201)#vlan 401
Brocade(config-vlan-401)#spanning-tree priority 3
Brocade(config-vlan-401)#tag ethernet 1/3 ethernet 5/1 to 5/3
...
Brocade(config-vlan-3601)#vlan 3801
Brocade(config-vlan-3801)#spanning-tree priority 20
Brocade(config-vlan-3801)#tag ethernet 1/20 ethernet 5/1 to 5/3
Brocade(config-vlan-3801)#exit
```

The next group of commands configures VLAN groups for the member VLANs. Notice that the VLAN groups do not contain the VLAN numbers assigned to the master VLANs. Also notice that no STP parameters are configured for the groups of member VLANs. Each group of member VLANs will inherit its STP settings from its master VLAN.

Set the bridge priority for each master VLAN to the highest priority (1) on one of the devices in the STP per VLAN group configuration. By setting the bridge priority to the highest priority, you make the device the default root bridge for the spanning tree. To ensure STP load balancing, make each of the devices the default root bridge for a different master VLAN.

```
Brocade(config)#vlan-group 1 vlan 2 to 200
Brocade(config-vlan-group-1)#tag ethernet 1/1 ethernet 5/1 to 5/3
Brocade(config-vlan-group-1)#vlan-group 2 vlan 202 to 400
Brocade(config-vlan-group-2)#tag ethernet 1/2 ethernet 5/1 to 5/3
Brocade(config-vlan-group-2)#vlan-group 3 vlan 402 to 600
Brocade(config-vlan-group-2)#tag ethernet 1/3 ethernet 5/1 to 5/3
...
Brocade(config-vlan-group-19)#vlan-group 20 vlan 3082 to 3282
Brocade(config-vlan-group-20)#tag ethernet 1/20 ethernet 5/1 to 5/3
Brocade(config-vlan-group-20)#exit
```

The following group of commands configures the STP groups. Each STP group in this configuration contains one master VLAN, which contains a VLAN group. This example shows that an STP group also can contain additional VLANs (VLANs not configured in a VLAN group).

```

Brocade(config)#stp-group 1
Brocade(config-stp-group-1)#master-vlan 1
Brocade(config-stp-group-1)#member-group 1
Brocade(config-stp-group-1)#member-vlan 4001 4004 to 4010
Brocade(config-stp-group-1)#stp-group 2
Brocade(config-stp-group-2)#master-vlan 201
Brocade(config-stp-group-2)#member-group 2
Brocade(config-stp-group-2)#member-vlan 4002 4003 4011 to 4015
Brocade(config-stp-group-2)#stp-group 3
Brocade(config-stp-group-3)#master-vlan 401
Brocade(config-stp-group-3)#member-group 3
...
Brocade(config-stp-group-19)#stp-group 20
Brocade(config-stp-group-20)#master-vlan 3081
Brocade(config-stp-group-20)#member-group 20

```

PVST/PVST+ compatibility

The FastIron family of switches support Cisco's Per VLAN Spanning Tree plus (PVST+), by allowing the device to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices¹.

NOTE

Brocade ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected. You do not need to perform any configuration steps to enable PVST+ support. However, to support the IEEE 802.1Q BPDUs, you might need to enable dual-mode support.

Support for Cisco's Per VLAN Spanning Tree plus (PVST+), allows a Brocade device to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices. Brocade ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected. The enhancement allows a port that is in PVST+ compatibility mode due to auto-detection to revert to the default MSTP mode when one of the following events occurs:

- The link is disconnected or broken
- The link is administratively disabled
- The link is disabled by interaction with the link-keepalive protocol

This enhancement allows a port that was originally interoperating with PVST+ to revert to MSTP when connected to a Brocade device.

Overview of PVST and PVST+

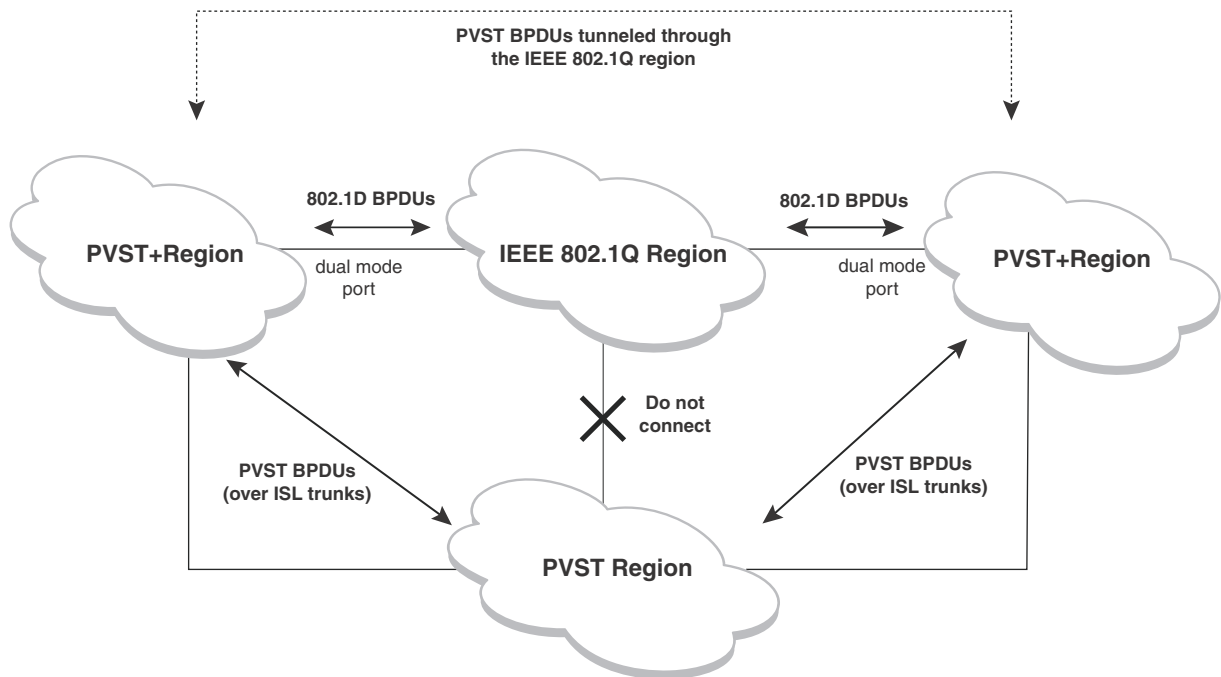
Per VLAN Spanning Tree (PVST) is a Cisco proprietary protocol that allows a Cisco device to have multiple spanning trees. The Cisco device can interoperate with spanning trees on other PVST devices but cannot interoperate with IEEE 802.1Q devices. An IEEE 802.1Q device has all its ports running a single spanning tree. **PVST+** is an extension of PVST that allows a Cisco device to also interoperate with devices that are running a single spanning tree (IEEE 802.1Q).

Enhanced PVST+ support allows a Brocade device to interoperate with PVST spanning trees and the IEEE 802.1Q spanning tree at the same time.

1. Cisco user documentation for PVST/PVST+ refers to the IEEE 802.1Q spanning tree as the **Common Spanning Tree (CST)**.

IEEE 802.1Q and PVST regions cannot interoperate directly but can interoperate indirectly through PVST+ regions. PVST BPDUs are tunneled through 802.1Q regions, while PVST BPDUs for VLAN 1 (the IEEE 802.1Q VLAN) are processed by PVST+ regions. [Figure 153](#) shows the interaction of IEEE 802.1Q, PVST, and PVST+ regions.

FIGURE 153 Interaction of IEEE 802.1Q, PVST, and PVST+ regions



VLAN tags and dual mode

The **dual-mode** feature enables a port to send and receive both tagged and untagged frames. When the dual-mode feature is enabled on a port, the port is an untagged member of one of its VLANs and is at the same time a tagged member of all its other VLANs. The untagged frames are supported on the port **Port Native VLAN**.

The dual-mode feature must be enabled on a Brocade port in order to interoperate with another vendor device. Some vendors use VLAN 1 by default to support the IEEE 802.1Q-based standard spanning tree protocols, such as 802.1d and 802.1w for sending untagged frames on VLAN 1. On Brocade switches, by default, the **Port Native VLAN** is the same as the **Default VLAN**, which is VLAN 1. Thus, to support IEEE 802.1Q in a typical configuration, a port must be able to send and receive untagged frames for VLAN 1 and tagged frames for the other VLANs, and interoperate with other vendor devices using VLAN 1.

If you want to use tagged frames on VLAN 1, you can change the default VLAN ID to an ID other than 1. You also can specify the VLAN on which you want the port to send and receive untagged frames (the Port Native VLAN). The Port Native VLAN ID does not need to be the same as the default VLAN. Make sure that the untagged (native) VLAN is also changed on the interoperating vendor side to match that on the Brocade side.

To support the IEEE 802.1Q with non-standard proprietary protocols such as PVST and PVST+, a port must always send and receive untagged frames on VLAN 1 on both sides. In this case, enable the dual-mode 1 feature to allow untagged BPDUs on VLAN 1 and use Native VLAN 1 on the interoperating vendor side. You should not use VLAN 1 for tagged frames in this case.

Configuring PVST+ support

PVST+ support is automatically enabled when the port receives a PVST BPDU. You can manually enable the support at any time or disable the support if desired.

If you want a tagged port to also support IEEE 802.1Q BPDUs, you need to enable the dual-mode feature on the port. The dual-mode feature is disabled by default and must be enabled manually.

A port that is in PVST+ compatibility mode due to auto-detection reverts to the default MSTP mode when one of the following events occurs:

- The link is disconnected or broken
- The link is administratively disabled
- The link is disabled by interaction with the link-keepalive protocol

This allows a port that was originally interoperating with PVST+ to revert to MSTP when connected to a Brocade device.

Enabling PVST+ support manually

To immediately enable PVST+ support on a port, enter commands such as the following.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#pvst-mode
```

Syntax: [no] pvst-mode

NOTE

If you disable PVST+ support, the software still automatically enables PVST+ support if the port receives a BPDU with PVST+ format.

NOTE

If 802.1W and pvst-mode (either by auto-detection or by explicit configuration) are enabled on a tagged VLAN port, 802.1W will treat the PVST BPDUs as legacy 802.1D BPDUs.

Enabling dual-mode support

To enable the dual-mode feature on a port, enter the following command at the interface configuration level for the port.

```
Brocade(config-if-1/1)#dual-mode
```

Syntax: [no] dual-mode [<vlan-id>]

The <vlan-id> specifies the port Port Native VLAN. This is the VLAN on which the port will support untagged frames. By default, the Port Native VLAN is the same as the default VLAN (which is VLAN 1 by default).

For more information about the dual-mode feature, refer to [“Dual-mode VLAN ports”](#) on page 818.

Displaying PVST+ support information

To display PVST+ information for ports on a Brocade device, enter the following command at any level of the CLI.

```
Brocade#show span pvst-mode
PVST+ Enabled on:
Port      Method
1/1       Set by configuration
1/2       Set by configuration
2/10      Set by auto-detect
3/12      Set by configuration
4/24      Set by auto-detect
```

Syntax: show span pvst-mode

This command displays the following information.

TABLE 201 CLI display of PVST+ information

Field	Description
Port	The Brocade port number. NOTE: The command lists information only for the ports on which PVST+ support is enabled.
Method	The method by which PVST+ support was enabled on the port. The method can be one of the following: <ul style="list-style-type: none"> Set by configuration – You enabled the support. Set by auto-detect – The support was enabled automatically when the port received a PVST+ BPDU.

PVST+ configuration examples

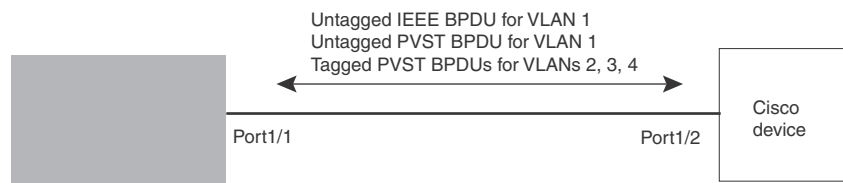
The following examples show configuration examples for two common configurations:

- Untagged IEEE 802.1Q BPDUs on VLAN 1 and tagged PVST+ BPDUs on other VLANs
- Tagged IEEE 802.1Q BPDUs on VLAN 1 and untagged BPDUs on another VLAN

Tagged port using default VLAN 1 as its port native VLAN

Figure 154 shows an example of a PVST+ configuration that uses VLAN 1 as the untagged default VLAN and VLANs 2, 3, and 4 as tagged VLANs.

FIGURE 154 Default VLAN 1 for untagged BPDU



To implement this configuration, enter the following commands.

Commands on the Brocade Device

```
Brocade(config)#vlan-group 1 vlan 2 to 4
Brocade(config-vlan-group-1)#tagged ethernet 1/1
Brocade(config-vlan-group-1)#exit
Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#dual-mode
Brocade(config-if-1/1)#pvst-mode
```

These commands configure a VLAN group containing VLANs 2, 3, and 4, add port 1/1 as a tagged port to the VLANs, and enable the dual-mode feature and PVST+ support on the port. The dual-mode feature allows the port to send and receive untagged frames for the default VLAN (VLAN 1 in this case) in addition to tagged frames for VLANs 2, 3, and 4. Enabling the PVST+ support ensures that the port is ready to send and receive PVST+ BPDUs. If you do not manually enable PVST+ support, the support is not enabled until the port receives a PVST+ BPDU.

The configuration leaves the default VLAN and the port Port Native VLAN unchanged. The default VLAN is 1 and the port Port Native VLAN also is 1. The dual-mode feature supports untagged frames on the default VLAN only. Thus, port 1/1 can send and receive untagged BPDUs for VLAN 1 and can send and receive tagged BPDUs for the other VLANs.

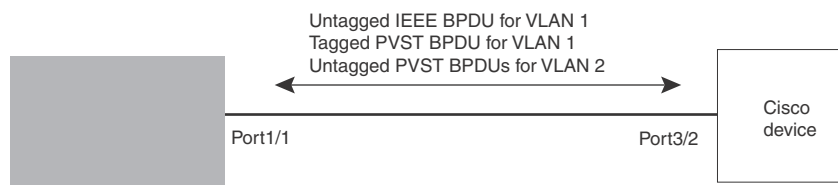
Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process tagged PVST BPDUs for VLANs 2, 3, and 4.
- Drop untagged PVST BPDUs for VLAN 1.

Untagged port using VLAN 2 as port native VLAN

Figure 155 shows an example in which a port Port Native VLAN is not VLAN 1. In this case, VLAN 1 uses tagged frames and VLAN 2 uses untagged frames.

FIGURE 155 Port Native VLAN 2 for Untagged BPDUs



To implement this configuration, enter the following commands.

Commands on the Brocade Device

```
Brocade(config)#default-vlan-id 4000
Brocade(config)#vlan 1
Brocade(config-vlan-1)#tagged ethernet 1/1
Brocade(config-vlan-1)#exit
Brocade(config)#vlan 2
Brocade(config-vlan-2)#tagged ethernet 1/1
Brocade(config-vlan-2)#exit
```

```

Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#dual-mode 2
Brocade(config-if-1/1)#pvst-mode
Brocade(config-if-1/1)#exit

```

These commands change the default VLAN ID, configure port 1/1 as a tagged member of VLANs 1 and 2, and enable the dual-mode feature and PVST+ support on port 1/1. Since VLAN 1 is tagged in this configuration, the default VLAN ID must be changed from VLAN 1 to another VLAN ID. Changing the default VLAN ID from 1 allows the port to process tagged frames for VLAN 1. VLAN 2 is specified with the **dual-mode** command, which makes VLAN 2 the port Port Native VLAN. As a result, the port processes untagged frames and untagged PVST BPDUs on VLAN 2.

NOTE

Although VLAN 2 becomes the port untagged VLAN, the CLI still requires that you add the port to the VLAN as a tagged port, since the port is a member of more than one VLAN.

Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process untagged PVST BPDUs for VLAN 2.
- Drop tagged PVST BPDUs for VLAN 1.

Note that when VLAN 1 is not the default VLAN, the ports must have the dual-mode feature enabled in order to process IEEE 802.1Q BPDUs.

For example, the following configuration is incorrect.

```

Brocade(config)#default-vlan-id 1000
Brocade(config)#vlan 1
Brocade(config-vlan-1)#tagged ethernet 1/1 to 1/2
Brocade(config-vlan-1)#exit
Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#pvst-mode
Brocade(config-if-1/1)#exit
Brocade(config)#interface ethernet 1/2
Brocade(config-if-1/2)#pvst-mode
Brocade(config-if-1/2)#exit

```

In the configuration above, all PVST BPDUs associated with VLAN 1 would be discarded. Since IEEE BPDUs associated with VLAN 1 are untagged, they are discarded because the ports in VLAN 1 are tagged. Effectively, the BPDUs are never processed by the Spanning Tree Protocol. STP assumes that there is no better bridge on the network and sets the ports to FORWARDING. This could cause a Layer 2 loop.

The following configuration is correct.

```

Brocade(config)#default-vlan-id 1000
Brocade(config)#vlan 1
Brocade(config-vlan-1)#tagged ethernet 1/1 to 1/2
Brocade(config-vlan-1)#exit
Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#pvst-mode
Brocade(config-if-1/1)#dual-mode
Brocade(config-if-1/1)#exit
Brocade(config)#interface ethernet 1/2
Brocade(config-if-1/2)#pvst-mode
Brocade(config-if-1/2)#dual-mode
Brocade(config-if-1/2)#exit

```


Setting the ports as dual-mode ensures that the untagged IEEE 802.1Q BPDUs reach the VLAN 1 instance.

PVRST compatibility

PVRST, the "rapid" version of per-VLAN spanning tree (PVST), is a Cisco proprietary protocol. PVRST corresponds to the Brocade full implementation of IEEE 802.1w (RSTP). Likewise, PVST, also a Cisco proprietary protocol, corresponds to the Brocade implementation of IEEE 802.1D (STP). When a Brocade device receives PVRST BPDUs on a port configured to run 802.1w, it recognizes and processes these BPDUs and continues to operate in 802.1w mode.

PVRST compatibility is automatically enabled when a port receives a PVRST BPDU.

BPDU guard

In an STP environment, switches, end stations, and other Layer 2 devices use Bridge Protocol Data Units (BPDUs) to exchange information that STP will use to determine the best path for data flow.

The BPDU guard, an enhancement to STP, removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Brocade port to which the end station is connected. STP BPDU guard shuts down the port and puts it into an errdisable state. This disables the connected device's ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a CLI message is displayed to warn the network administrator of a severe invalid configuration. The BPDU guard feature provides a secure response to invalid configurations because the administrator must manually put the interface back in service if errdisable recovery is not enabled.

NOTE

BPDU guard is not supported on tagged ports. It can be configured on a tagged port, but the configuration will have no effect.

Enabling BPDU protection by port

You enable STP BPDU guard on individual interfaces. The feature is disabled by default.

To enable STP BPDU guard on a specific port, enter a command such as the following.

```
Brocade(config) interface ethe 2/1
Brocade(config-if-e1000-2/1)#stp-bpdu-guard
```

Syntax: `[no] stp-bpdu-guard`

The **no** parameter disables the BPDU guard on this interface.

You can also use the multiple interface command to enable this feature on multiple ports at once.

Example

```
Brocade(config)#interface ethernet 1/1 to 1/9
Brocade(config-mif-1/1-1/9)#stp-bpdu-guard
Brocade(config-mif-1/1-1/9)#
```

This will enable stp-bpdu-guard on ports 0/1/1 to 0/1/9

Re-enabling ports disabled by BPDU guard

When a BPDU Guard-enabled port is disabled by BPDU Guard, the Brocade device will place the port in **errdisable** state and display a message on the console indicating that the port is errdisabled (refer to [“BPDU guard status example console messages”](#) on page 1165). In addition, the **show interface** command output will indicate that the port is errdisabled.

Example

```
Brocade#show int e 2
Gigabit Ethernet2 is ERR-DISABLED (bpduguard), line protocol is down
```

To re-enable a port that is in **errdisable** state, you must first disable the port then re-enable it. Enter commands such as the following.

```
Brocade(config)#int e 2
Brocade(config-if-e1000-2)#disable
Brocade(config-if-e1000-2)#enable
```

If you attempt to enable an errdisabled port without first disabling it, the following error message will appear on the console.

```
Brocade(config-if-e1000-2)#enable
Port 2 is errdisabled, do disable first and then enable to enable it
```

Displaying the BPDU guard status

To display the BPDU guard state, enter the **show running configuration** or the **show stp-bpdu-guard** command.

For FastIron X Series devices, enter the **stp-bpdu-guard** command.

```
Brocade#show stp-bpdu-guard
BPDU Guard Enabled on:
Interface Violation
Port 1 No
Port 2 No
Port 3 No
Port 4 No
Port 5 No
Port 6 No
Port 7 No
Port 8 No
Port 9 No
Port 10 No
Port 11 No
Port 12 Yes
Port 13 No
```

BPDU guard status example configurations

Example

The following example shows how to configure BPDU guard at the interface level and to verify the configuration by issuing the **show stp-bpdu-guard** and the **show interface** commands.

```
Brocade(config)#interface ethernet 1
Brocade(config-if-e1000-1)#stp-bpdu-guard
Brocade(config-if-e1000-1)#
Brocade(config-if-e1000-1)#show stp-bpdu-guard
BPDU Guard Enabled on:
Port
1
Brocade(config-if-e1000-1)#
Brocade(config-if-e1000-1)#show interfaces ethernet 1
GigabitEthernet1 is up, line protocol is up
Hardware is GigabitEthernet, address is 000c.dba0.7100 (bia 000c.dba0.7100)
Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
Configured mdi mode AUTO, actual MDI
Member of L2 VLAN ID 2, port is untagged, port state is FORWARDING
BPDU guard is Enabled, R00T protect is Disabled
STP configured to ON, priority is level0, flow control enabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 96 bits-time, IPG GMII 96 bits-time
IP MTU 1500 bytes
300 second input rate: 8 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 256 bits/sec, 0 packets/sec, 0.00% utilization
88 packets input, 15256 bytes, 0 no buffer
Received 75 broadcasts, 13 multicasts, 0 unicasts
1 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
4799 packets output, 313268 bytes, 0 underruns
Transmitted 90 broadcasts, 4709
```

BPDU guard status example console messages

A console message such as the following is generated after a BPDU guard violation occurs on a system that is running MSTP.

```
Brocade(config-if-e1000-23)#MSTP: Received BPDU on BPDU guard enabled Port
23,errdisable Port 23
```

A console message such as the following is generated after a BPDU guard violation occurs on a system that is running STP.

```
Brocade(config)#STP: Received BPDU on BPDU guard enabled Port 23 (vlan=1),
errdisable Port 23
```

A console message such as the following is generated after a BPDU guard violation occurs on a system that is running RSTP.

```
Brocade(config-vlan-1)#RSTP: Received BPDU on BPDU guard enabled Port 23
(vlan=1),errdisable Port 23
```

Root guard

The standard STP (802.1D), RSTP (802.1W) or 802.1S does not provide any way for a network administrator to securely enforce the topology of a switched layer 2 network. The forwarding topology of a switched network is calculated based on the root bridge position, along with other parameters. This means any switch can be the root bridge in a network as long as it has the lowest bridge ID. The administrator cannot enforce the position of the root bridge. A better forwarding topology comes with the requirement to place the root bridge at a specific predetermined location. Root Guard can be used to predetermine a root bridge location and prevent rogue or unwanted switches from becoming the root bridge.

When root guard is enabled on a port, it keeps the port in a designated role. If the port receives a superior STP Bridge Protocol Data Units (BPDU), it puts the port into a ROOT-INCONSISTANT state and triggers a log message and an SNMP trap. The ROOT-INCONSISTANT state is equivalent to the BLOCKING state in 802.1D and to the DISCARDING state in 802.1W. No further traffic is forwarded on this port. This allows the bridge to prevent traffic from being forwarded on ports connected to rogue or misconfigured STP bridges.

Once the port stops receiving superior BPDUs, root guard automatically sets the port back to learning, and eventually to a forwarding state through the spanning-tree algorithm.

Configure root guard on all ports where the root bridge should not appear. This establishes a protective network perimeter around the core bridged network, cutting it off from the user network.

NOTE

Root guard may prevent network connectivity if it is improperly configured. Root guard must be configured on the perimeter of the network rather than the core.

NOTE

Root guard is not supported when MSTP is enabled.

Enabling STP root guard

An STP root guard is configured on an interface by entering commands similar to the following.

```
Brocade(config)#interface ethernet 5/5
Brocade(config-if-e10000-5/5)spanning-tree root-protect
```

Syntax: [no] spanning-tree root-protect

Enter the **no** form of the command to disable STP root guard on the port.

Displaying the STP root guard

To display the STP root guard state, enter the **show running configuration** or the **show spanning-tree root-protect** command.

```
Brocade#show spanning-tree root-protect
Root Protection Enabled on:
Port 1
```

Syntax: show spanning-tree root-protect

Displaying the root guard by VLAN

You can display root guard information for all VLANs or for a specific VLAN. For example, to display root guard violation information for VLAN 7.

Syntax: `show spanning-tree [<vlan-id>]`

If you do not specify a `<vlan-id>`, information for all VLANs is displayed. For example, to display root guard violation information for VLAN 7.

```
Brocade#show spanning-tree vlan 7
STP instance owned by VLAN 7
Global STP (IEEE 802.1D) Parameters:
VLAN Root Root Root Prio Max He- Ho- Fwd Last Chg Bridge
ID ID Cost Port rity Age llo ld dly Chang cnt Address
Hex sec sec sec sec sec
7 a000000011112220 0 Root a000 20 2 1 15 4 4 000011112220
Port STP Parameters:
Port Prio Path State Fwd Design Designated Designated
Num rity Cost Trans Cost Root Bridge
Hex
1 80 19 ROOT-INCONS 2 0 a000000011112220 a000000011112220
```

Error disable recovery

In case a BPDU guard violation occurs, a port is placed into an errdisable state which is functionally equivalent to a Disable state. Once in an errdisable state, it remains in that state until one of the following methods is used to return the port to an Enabled state.

1. Manually disabling and enabling that interface
2. Automatically, through the errdisable recovery mechanism

The **errdisable recovery interval** command is used to configure a time-out for ports in errdisable state, after which the ports are re-enabled automatically.

When BPDU guard puts a port into errdisabled state, the port remains in errdisabled state unless it is enabled manually by issuing a **disable** command and then the **enable** command on the associated interface or you have errdisable recovery turned on. The **errdisable** command allows you to choose the type of error that automatically reenables the port after a specified amount of time.

Enabling error disable recovery

To enable errdisable recovery for BPDU Guard, enter a command such as the following.

```
Brocade(config)#errdisable recovery cause bpduguard
```

To enable error disable recovery for any reason, enter a command such as the following.

```
Brocade(config)#errdisable recovery cause all
```

Syntax: `errdisable recovery [cause bpduguard | all]`

The **cause** is the reason why the port is in the errdisable state. Valid values are **bpduguard** and **all**.

Use the **bpduguard** parameter to allow the port to recover from the errdisabled state, if the state was caused by a BPDU guard violation.

The **all** parameter allows ports to recover from an errdisabled state caused by any reason, for example, a BPDU Guard violation or loop detection violation.

Setting the recovery interval

The **errdisable recovery interval** command allows you to configure a timeout for ports in errdisabled state, after which the ports are reenabled automatically. To set the errdisable recovery time-out interval, enter a command such as the following.

```
Brocade(config)#errdisable recovery interval 20
```

Syntax: [no] **errdisable recovery interval** <seconds>

The **seconds** parameter allows you to set the timeout value for the recovery mechanism when the port is in an errdisabled state. Once this timeout value expires, the ports are automatically re-enabled. Valid values are from 10 to 65535 seconds (10 seconds to 24 hours).

Displaying the error disable recovery state by interface

The port status of errdisabled displays in the output of the **show interface** and the **show interface brief** commands. In this example, errdisable is enabled on interface ethernet 1 and errdisable is enabled because of a BPDU guard violation.

```
Brocade#show interfaces ethernet 1
GigabitEthernet1 is ERR-DISABLED (bpduguard), line protocol is down
BPDU guard is Enabled, ROOT protect is Disabled
Hardware is GigabitEthernet, address is 000c.dba0.7100 (bia 000c.dba0.7100)
Configured speed auto, actual unknown, configured duplex fdx, actual unknown
Configured mdi mode AUTO, actual unknown
Member of L2 VLAN ID 2, port is untagged, port state is DISABLED
STP configured to ON, priority is level0, flow control enabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 96 bits-time, IPG GMII 96 bits-time
IP MTU 1500 bytes
300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
145 packets input, 23561 bytes, 0 no buffer
Received 124 broadcasts, 21 multicasts, 0 unicasts
1 input errors, 0 CRC, 0 frame, 0 ignored
0 runs, 0 giants
5067 packets output, 330420 bytes, 0 underruns
Transmitted 90 broadcasts, 4977 multicasts, 0 unicasts
0 output errors, 0 collisions
```

Displaying the recovery state for all conditions

Use the **show errdisable recovery** command to display all the default error disable recovery state for all possible conditions. In this example, port 6 is undergoing a recovery.

```
Brocade#show errdisable recovery
ErrDisable Reason Timer Status
-----
all reason Disabled
bpduguard Enabled
```

```

Timeout Value: 300 seconds
Interface that will be enabled at the next timeout:
Interface Errdisable reason Time left (sec)
-----
Port 6 bpduguard 297

```

Syntax: show errdisable recovery

Displaying the recovery state by port number and cause

To see which ports are under an errdisabled state, use the **show errdisable summary** command. This command not only shows the port number, but also displays the reason why the port is in an errdisable state and the method used to recover the port. In this example, port 6 is errdisabled for a BPDU guard violation.

```

Brocade#show errdisable summary
Port 6 ERR_Disabled for bpduguard

```

Syntax: show errdisable summary

Errdisable Syslog messages

When the system places a port into an errdisabled state for BPDU guard, a log message is generated. When the errdisable recovery timer expires, a log message is also generated.

A Syslog message such as the following is generated after a port is placed into an errdisable state for BPDU guard.

```

STP: VLAN 50 BPDU-guard port 3 detect (Received BPDU), putting into err-disable
state

```

A Syslog message such as the following is generated after the recovery timer expires.

```

ERR_DISABLE: Interface ethernet 3, err-disable recovery timeout

```

802.1s Multiple Spanning Tree Protocol

Multiple Spanning Tree Protocol (MSTP), as defined in IEEE 802.1s, allows multiple VLANs to be managed by a single STP instance and supports per-VLAN STP. As a result, several VLANs can be mapped to a reduced number of spanning-tree instances. This ensures loop-free topology for one or more VLANs that have the similar layer-2 topology. The Brocade implementation supports up to 16 spanning tree instances in an MSTP enabled bridge which means that it can support up to 16 different Layer 2 topologies. The spanning tree algorithm used by MSTP is RSTP which provides quick convergence.

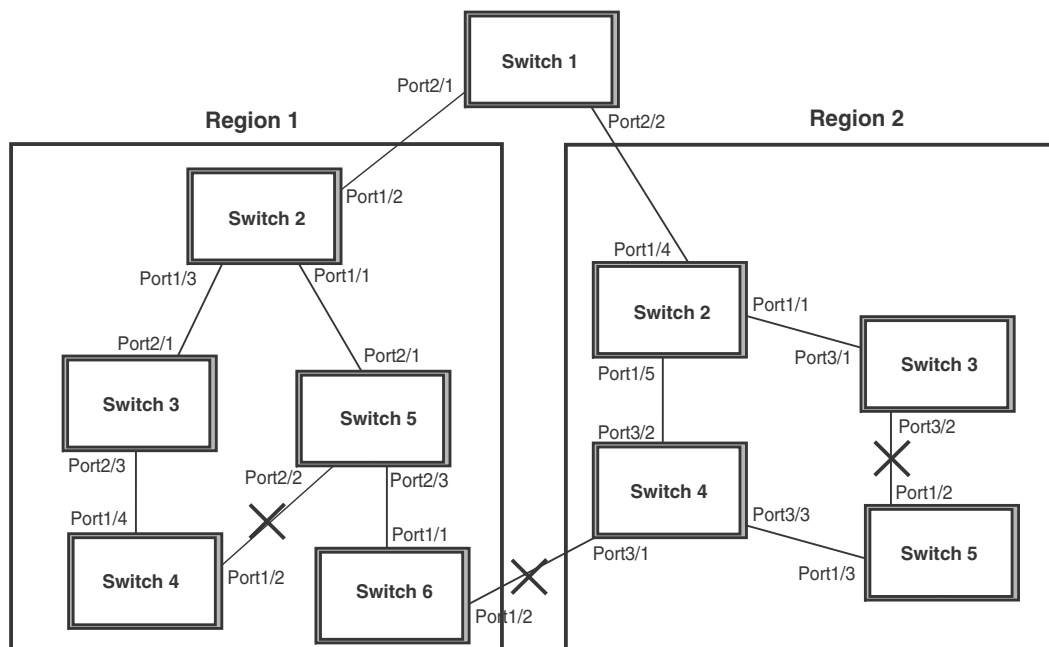
Multiple spanning-tree regions

Using MSTP, the entire network runs a common instance of RSTP. Within that common instance, one or more VLANs can be individually configured into distinct regions. The entire network runs the common spanning tree instance (CST) and the regions run a local instance. The local instance is known as Internal Spanning Tree (IST). The CST treats each instance of IST as a single bridge. Consequently, ports are blocked to prevent loops that might occur within an IST and also throughout the CST. With the exception of the provisions for multiple instances, MSTP operates exactly like RSTP.

For example, in [Figure 156](#) a network is configured with two regions: Region 1 and Region 2. The entire network is running an instance of CST. Each of the regions is running an instance of IST. In addition, this network contains Switch 1 running MSTP that is not configured in a region and consequently is running in the CIST instance. In this configuration, the regions are each regarded as a single bridge to the rest of the network, as is Switch 1. The CST prevents loops from occurring across the network. Consequently, a port is blocked at port 1/2 of switch 4.

Additionally, loops must be prevented in each of the IST instances. Within the IST Region 1, a port is blocked at port 1/2 of switch 4 to prevent a loop in that region. Within Region 2, a port is blocked at port 3/2 of switch 3 to prevent a loop in that region.

FIGURE 156 MSTP configured network



The following definitions describe the STP instances that define an MSTP configuration.

Common Spanning (CST) – CST is defined in 802.1q and assumes one spanning-tree instance for the entire bridged network regardless of the number of VLANs. In MSTP, an MSTP region appears as a virtual bridge that runs CST.

Internal Spanning Tree (IST) – IST is a new terminology introduced in 802.1s. An MSTP bridge must handle at least these two instances: one IST and one or more MSTIs (Multiple Spanning Tree Instances). Within each MST region, the MSTP maintains multiple spanning-tree instances. Instance 0 is a special instance known as IST, which extends CST inside the MST region. IST always exists if the switch runs MSTP. Besides IST, this implementation supports up to 15 MSTIs, numbered from 1 to 4094.

Common and Internal Spanning Trees (CIST) – CIST is a collection of the ISTs in each MST region and the CST that interconnects the MST regions and single spanning trees.

Multiple Spanning Tree Instance (MSTI) – The MSTI is identified by an MST identifier (MSTid) value between 1 and 4094.

MSTP Region – These are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels. Also, one or more VLANs can be mapped to one MSTP instance (IST or MSTI) but a VLAN cannot be mapped to multiple MSTP instances.

NOTE

One or more VLANs can be mapped to one MSTP instance (IST or MSTI) but a VLAN cannot be mapped to multiple MSTP instances.

Configuration notes

When configuring MSTP, note the following:

- With MSTP running, enabling static trunk on ports that are members of many VLANs (4000 or more VLANs) will keep the system busy for 20 to 25 seconds.

Configuring MSTP mode and scope

With the introduction of MSTP, a system can be either under MSTP mode or not under MSTP mode. The default state is to **not** be under MSTP mode. MSTP configuration can only be performed in a system under MSTP mode.

With a system configured under MSTP mode, there is a concept called MSTP scope. MSTP scope defines the VLANs that are under direct MSTP control. You cannot run 802.1D or 802.1w on any VLAN (even outside of MSTP scope) and you cannot create topology groups when a system is under MSTP mode. While a VLAN group will still be supported when a system is under MSTP mode, the member VLAN should either be all in the MSTP scope or all out of the MSTP scope.

When a system is configured from non-MSTP mode to MSTP mode, the following changes are made to the system configuration:

- All 802.1D and 802.1w STP instances are deleted regardless of whether the VLAN is inside the MSTP scope or not
- All topology groups are deleted
- Any GVRP configuration is deleted
- Any VSRP configuration is deleted
- Single-span (if configured) is deleted
- MRP running on a VLAN inside MSTP scope is deleted

- The CIST is created and all VLANs inside the MSTP scope are attached with the CIST

Make sure that no physical layer-2 loops exist prior to switching from non-MSTP mode to MSTP mode. If, for example, you have an L2 loop topology configured as a redundancy mechanism before you perform the switch, a Layer 2 storm should be expected.

To configure a system into MSTP mode, use the following command at the Global Configuration level.

```
Brocade(config)#mstp scope all
```

Syntax: [no] mstp scope all

NOTE

MSTP is not operational however until the **mstp start** command is issued as described in [“Activating MSTP on a switch”](#) on page 1178.

Once the system is configured into MSTP mode, CIST (sometimes referred to as “instance 0”) is created and all existing VLANs inside the MSTP scope are controlled by CIST. In addition, whenever you create a new VLAN inside MSTP scope, it is put under CIST control by default. In the Brocade MSTP implementation however, a VLAN ID can be pre-mapped to another MSTI as described in [“Configuring an MSTP instance”](#) on page 1175. A VLAN whose ID is pre-mapped, will attach to the specified MSTI instead of to the CIST when created.

NOTE

Once under MSTP mode, CIST always controls all ports in the system. If you do not want a port to run MSTP, configure the **no spanning-tree** command under the specified interface configuration.

Using the [no] option on a system that is configured for MSTP mode changes the system to non-MSTP mode. When this switch is made, all MSTP instances are deleted together with all MSTP configurations. ALL VLANs inside the original MSTP scope will not run any Layer-2 protocols after the switch.

Reduced occurrences of MSTP reconvergence

When a VLAN is deleted, the Brocade device retains the associated VLAN to MSTI mapping instead of deleting it from the configuration. This way, a VLAN can be pre-mapped to an MSTI and MSTP reconvergence may not be necessary when a VLAN is added to or deleted from the configuration. As long as the VLAN being created or deleted is pre-mapped to an MSTI, and the VLAN to MSTI mapping has not changed, MSTP reconvergence will not occur.

NOTE

MSTP reconvergence occurs when the VLAN to MSTI mapping is changed using the **mstp instance** command.

You can optionally remove VLAN to MSTI mappings from the configuration. Refer to [“Deleting a VLAN to MSTI mapping”](#) on page 1173.

The following shows an example application.

Example application of MSTP reconvergence

The following example shows the running configuration file before and after deleting a VLAN from the configuration. The VLAN to MSTI mapping is retained in the running configuration, even after the VLAN is deleted.

```

Brocade(config-vlan-20)#show run
Current configuration:
!
ver 04.2.00bT3e1
!
!
vlan 1 name DEFAULT-VLAN by port
    no spanning-tree
!
vlan 10 by port
    tagged ethe 1 to 2
    no spanning tree
!
vlan 20 by port                                <----- VLAN 20 configuration
    tagged ethe 1 to 2
    no spanning-tree
!
mstp scope all
mstp instance 0 vlan 1
mstp instance 1 vlan 20
mstp start
some lines ommitted for brevity...

Brocade(config-vlan-20)#no vlan 20            <----- VLAN 20 deleted

Brocade(config-vlan-20)#show run
Current configuration:
!
ver 04.2.00bT3e1
!
!
vlan 1 name DEFAULT-VLAN by port
    no spanning-tree
!
vlan 10 by port
    tagged ethe 1 to 2
    no spanning-tree
!
mstp scope all
mstp instance 0 vlan 1
mstp instance 1 vlan 10
mstp instance 1 vlan 20
mstp start
some lines ommitted for brevity...
<----- VLAN to MSTI mapping kept in
running configuration, even though
VLAN 20 was deleted

```

Deleting a VLAN to MSTI mapping

You can optionally remove a VLAN to MSTI mapping using the **no mstp instance** command. To do so, enter a command such as the following.

```
Brocade(config)#no mstp instance 7 vlan 4 to 7
```

This command deletes the VLAN to MSTI mapping from the running configuration and triggers an MSTP reconvergence.

Syntax: **no mstp instance** <instance-number> **vlan** <vlan-id> | **vlan-group** <group-id>]

The **instance** parameter defines the number for the instance of MSTP that you are deleting.

The **vlan** parameter identifies one or more VLANs or a range of VLANs to the instance defined in this command.

The **vlan-group** parameter identifies one or more VLAN groups to the instance defined in this command.

Viewing the MSTP configuration digest

The MSTP Configuration Digest indicates the occurrence of an MSTP reconvergence. The Configuration Digest is recalculated whenever an MSTP reconvergence occurs. To view the Configuration Digest, use the **show mstp config** command. The following shows an example output.

```
SW-FESX624F+2XG Switch(config-vlan-20)#show mstp config
MSTP CONFIGURATION
-----
Scope          : all system
Name           :
Revision       : 0
Version        : 3 (MSTP mode)
Config Digest: 0x9bbda9c70d91f633e1e145fbcbf8d321
Status        : Started

Instance VLANs
-----
0          1
1          10 20
```

Syntax: **show mstp config**

Configuring additional MSTP parameters

To configure a switch for MSTP, you could configure the name and the revision on each switch that is being configured for MSTP. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all switches that interoperate with the same VLAN assignments. Port cost, priority and global parameters can then be configured for individual ports and instances. In addition, operational edge ports and point-to-point links can be created and MSTP can be disabled on individual ports.

Each of the commands used to configure and operate MSTP are described in the following:

- [“Setting the MSTP name”](#)
- [“Setting the MSTP revision number”](#)
- [“Configuring an MSTP instance”](#)
- [“Configuring bridge priority for an MSTP instance”](#)
- [“Setting the MSTP global parameters”](#)
- [“Setting ports to be operational edge ports”](#)
- [“Setting automatic operational edge ports”](#)
- [“Setting point-to-point link”](#)
- [“Disabling MSTP on a port”](#)

- “Forcing ports to transmit an MSTP BPDU”
- “Activating MSTP on a switch”

Setting the MSTP name

Each switch that is running MSTP is configured with a name. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP name, use a command such as the following at the Global Configuration level.

```
Brocade(config)#mstp name Brocade
```

Syntax: [no] mstp name <name>

The **name** parameter defines an ASCII name for the MSTP configuration. The default name is for the name variable to be blank.

Setting the MSTP revision number

Each switch that is running MSTP is configured with a revision number. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP revision number, use a command such as the following at the Global Configuration level.

```
Brocade(config)#mstp revision 4
```

Syntax: [no] mstp revision <revision-number>

The **revision** parameter specifies the revision level for MSTP that you are configuring on the switch. It can be a number from 0 and 65535. The default revision number is 0.

Configuring an MSTP instance

An MSTP instance is configured with an MSTP ID for each region. Each region can contain one or more VLANs. The Brocade implementation of MSTP allows you to assign VLANs or ranges of VLANs to an MSTP instance before or after they have been defined. If pre-defined, a VLAN will be placed in the MSTI that it was assigned to immediately when the VLAN is created. Otherwise, the default operation is to condition of assign all new VLANs to the CIST. VLANs assigned to the CIST by default can be moved later to a specified MSTI.

To configure an MSTP instance and map one or more VLANs to that MSTI, use a command such as the following at the Global Configuration level.

```
Brocade(config)#mstp instance 7 vlan 4 to 7
```

Syntax: [no] mstp instance <instance-number> [vlan <vlan-id> | vlan-group <group-id>]

The **instance** parameter defines the number for the instance of MSTP that you are configuring. The value 0 (which identifies the CIST) cannot be used. You can have up to 15 instances, number 1 – 4094.

The **vlan** parameter assigns one or more VLANs or a range of VLANs to the instance defined in this command.

The **vlan-group** parameter assigns one or more VLAN groups to the instance defined in this command.

The **no** option moves a VLAN or VLAN group from its assigned MSTI back into the CIST.

NOTE

The system does not allow an MSTI without any VLANs mapped to it. Consequently, removing all VLANs from an MSTI, deletes the MSTI from the system. The CIST by contrast will exist regardless of whether or not any VLANs are assigned to it or not. Consequently, if all VLANs are moved out of a CIST, the CIST will still exist and functional.

Configuring bridge priority for an MSTP instance

Priority can be configured for a specified instance. To configure priority for an MSTP instance, use a command such as the following at the Global Configuration level.

```
Brocade(config)#mstp instance 1 priority 8192
```

Syntax: [no] mstp instance <instance-number> priority <priority-value>

The <instance-number> variable is the number for the instance of MSTP that you are configuring.

You can set a **priority** to the instance that gives it forwarding preference over lower priority instances within a VLAN or on the switch. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 61440 in increments of 4096. The default value is 32768.

Setting the MSTP global parameters

MSTP has many of the options available in RSTP as well as some unique options. To configure MSTP Global parameters for all instances on a switch.

```
Brocade(config)#mstp force-version 0 forward-delay 10 hello-time 4 max-age 12
max-hops 9
```

Syntax: [no] mstp force-version <mode-number> forward-delay <value> hello-time <value>
max-age <value> max-hops <value>

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following <mode-number> values:

- 0 – The STP compatibility mode. Only STP BPDUs will be sent. This is equivalent to single STP.
- 2 – The RSTP compatibility mode. Only RSTP BPDUS will be sent. This is equivalent to single STP.
- 3 – MSTP mode. In this default mode, only MSTP BPDUS will be sent.

The **forward-delay** <value> specifies how long a port waits before it forwards an RST BPDU after a topology change. This can be a value from 4 – 30 seconds. The default is 15 seconds.

The **hello-time** <value> parameter specifies the interval between two hello packets. The parameter can have a value from 1 – 10 seconds. The default is 2 seconds.

The **max-age** <value> parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 – 40 seconds, where the value adheres to the following formula.

max age equal to or greater than 2 x (hello-time + 1) AND max age equal to or greater than 2 x (forward-delay – 1)

The default max-age is 20 seconds.

The **max-hops** *<value>* parameter specifies the maximum hop count. You can specify a value from 1 – 40 hops. The default value is 20 hops.

Setting ports to be operational edge ports

You can define specific ports as edge ports for the region in which they are configured to connect to devices (such as a host) that are not running STP, RSTP, or MSTP. If a port is connected to an end device such as a PC, the port can be configured as an edge port. To configure ports as operational edge ports enter a command such as the following.

```
Brocade(config)#mstp admin-edge-port ethernet 3/1
```

Syntax: [no] **mstp admin-edge-port ethernet** *<port>*

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Setting automatic operational edge ports

You can configure a Layer 3 switch to automatically set a port as an operational edge port if the port does not receive any BPDUs since link-up. If the port receives a BPDU later, it is automatically reset to become an operational non-edge port. This feature is set globally to apply to all ports on a router where it is configured. This feature is configured as shown in the following.

```
Brocade(config)#mstp edge-port-auto-detect
```

Syntax: [no] **mstp edge-port-auto-detect**

NOTE

If this feature is enabled, it takes the port about 3 seconds longer to come to the enable state.

Setting point-to-point link

You can set a point-to-point link between ports to increase the speed of convergence. To create a point-to-point link between ports, use a command such as the following at the Global Configuration level.

```
Brocade(config)#mstp admin-pt2pt-mac ethernet 2/5 ethernet 4/5
```

Syntax: [no] **mstp admin-pt2pt-mac ethernet** *<port>*

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Disabling MSTP on a port

To disable MSTP on a specific port, use a command such as the following at the Global Configuration level.

```
Brocade(config)#mstp disable ethernet 2/1
```

Syntax: [no] mstp disable ethernet <port>

The <port> variable specifies the location of the port for which you want to disable MSTP. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

When a port is disabled for MSTP, it behaves as blocking for all the VLAN traffic that is controlled by MSTIs and the CIST.

Forcing ports to transmit an MSTP BPDU

To force a port to transmit an MSTP BPDU, use a command such as the following at the Global Configuration level.

```
Brocade(config)#mstp force-migration-check ethernet 3/1
```

Syntax: [no] mstp force-migration-check ethernet <port>

The <port> variable specifies the port or ports from which you want to transmit an MSTP BPDU. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Activating MSTP on a switch

MSTP scope must be enabled on the switch as described in [“Configuring MSTP mode and scope”](#) on page 1171 before MSTP can be enabled.

To enable MSTP on your switch, use the following at the Global Configuration level.

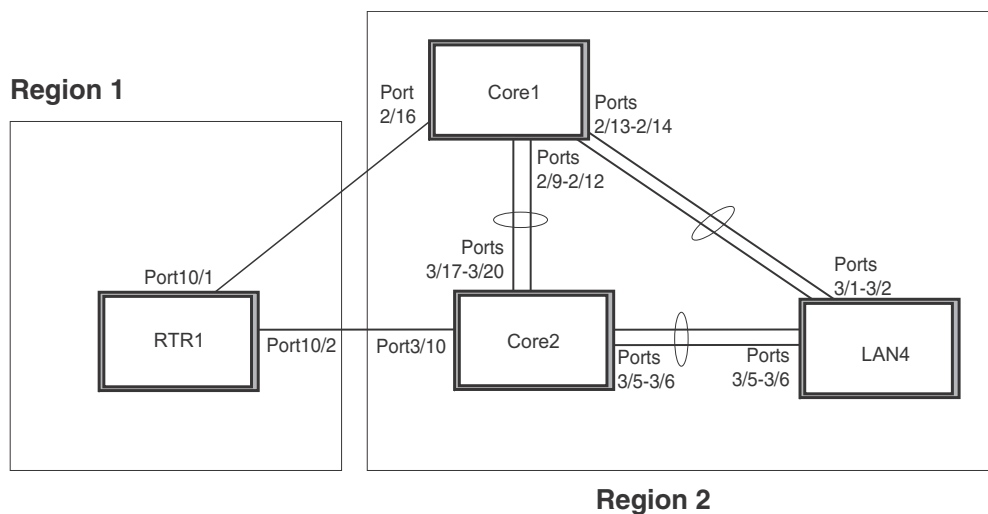
```
Brocade(config)#mstp start
```

Syntax: [no] mstp start

The [no] option disables MSTP from operating on a switch.

Example of an MSTP configuration

In [Figure 157](#) four Brocade device routers are configured in two regions. There are four VLANs in four instances in Region 2. Region 1 is in the CIST.

FIGURE 157 Sample MSTP configuration**RTR1 on MSTP configuration**

```
Brocade(config-vlan-4093)#tagged ethernet 10/1 to 10/2
Brocade(config-vlan-4093)#exit
Brocade(config)#mstp scope all
Brocade(config)#mstp name Reg1
Brocade(config)#mstp revision 1
Brocade(config)#mstp admin-pt2pt-mac ethernet 10/1 to 10/2
Brocade(config)#mstp start
Brocade(config)#hostname RTR1
```

Core 1 on MSTP configuration

```
Brocade(config)#trunk ethernet 2/9 to 2/12 ethernet 2/13 to 2/14
Brocade(config-vlan-1)#name DEFAULT-VLAN by port
Brocade(config-vlan-1)#exit
Brocade(config)#vlan 20 by port
Brocade(config-vlan-20)#tagged ethernet 2/9 to 2/14 ethernet 2/16
Brocade(config-vlan-20)#exit
Brocade(config)#vlan 21 by port
Brocade(config-vlan-21)#tagged ethernet 2/9 to 2/14 ethernet 2/16
Brocade(config-vlan-21)#exit
Brocade(config)#vlan 22 by port
Brocade(config-vlan-22)#tagged ethernet 2/9 to 2/14 ethernet 2/16
Brocade(config-vlan-22)#exit
Brocade(config)#vlan 23 by port
Brocade(config)#mstp scope all
Brocade(config)#mstp name HR
Brocade(config)#mstp revision 2
Brocade(config)#mstp instance 20 vlan 20
Brocade(config)#mstp instance 21 vlan 21
Brocade(config)#mstp instance 22 vlan 22
Brocade(config)#mstp instance 0 priority 8192
Brocade(config)#mstp admin-pt2pt-mac ethernet 2/9 to 2/14
```

802.1s Multiple Spanning Tree Protocol

```
Brocade(config)#mstp admin-pt2pt-mac ethernet 2/16
Brocade(config)#mstp disable ethernet 2/240.
Brocade(config)#mstp start
Brocade(config)#hostname CORE1
```

Core2 on MSTP configuration

```
Brocade(config)#trunk ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
Brocade(config)#vlan 1 name DEFAULT-VLAN by port
Brocade(config-vlan-1)#exit
Brocade(config)#vlan 20 by port
Brocade(config-vlan-20)#tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
Brocade(config-vlan-20)#exit
Brocade(config)#vlan 21 by port
Brocade(config-vlan-21)#tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
Brocade(config-vlan-21)#exit
Brocade(config)#vlan 22 by port
Brocade(config-vlan-22)#tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
Brocade(config-vlan-22)#exit
Brocade(config)#mstp scope all
Brocade(config)#mstp name HR
Brocade(config)#mstp revision 2
Brocade(config)#mstp instance 20 vlan 20
Brocade(config)#mstp instance 21 vlan 21
Brocade(config)#mstp instance 22 vlan 22
Brocade(config)#mstp admin-pt2pt-mac ethernet 3/17 to 3/20 ethernet 3/5 to 3/6
Brocade(config)#mstp admin-pt2pt-mac ethernet 3/10
Brocade(config)#mstp disable ethernet 3/7 ethernet 3/24
Brocade(config)#mstp start
Brocade(config)#hostname CORE2
```

LAN 4 on MSTP configuration

```
Brocade(config)#trunk ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
Brocade(config)#vlan 1 name DEFAULT-VLAN by port
Brocade(config-vlan-1)#exit
Brocade(config)#vlan 20 by port
Brocade(config-vlan-20)#tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
Brocade(config)#exit
Brocade(config)#vlan 21 by port
Brocade(config-vlan-21)#tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
Brocade(config-vlan-21)#exit
Brocade(config)#vlan 22 by port
Brocade(config-vlan-22)#tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
Brocade(config)#mstp scope all
Brocade(config)#mstp config name HR
Brocade(config)#mstp revision 2
Brocade(config)#mstp instance 20 vlan 20
Brocade(config)#mstp instance 21 vlan 21
Brocade(config)#mstp instance 22 vlan 22
Brocade(config)#mstp admin-pt2pt-mac ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
Brocade(config)#mstp start
Brocade(config)#hostname LAN4
```

Displaying MSTP statistics

MSTP statistics can be displayed using the commands shown below.

To display all general MSTP information, enter the following command.

```
Brocade#show mstp
MSTP Instance 0 (CIST) - VLANs: 1
-----
Bridge          Bridge Bridge Bridge Bridge Root   Root   Root   Root
Identifier      MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex             sec     sec    sec   cnt    sec     sec   sec   cnt
8000000cdb80af01 20      2      15    20     20      2     15    19

Root            ExtPath  RegionalRoot      IntPath  Designated      Root
Bridge          Cost      Bridge            Cost      Bridge            Port
hex             hex             hex             hex
8000000480bb9876 2000      8000000cdb80af01 0          8000000480bb9876 3/1

Port  Pri  PortPath  P2P Edge Role      State      Designa-  Designated
Num    Cost    Mac Port              ted cost   bridge
3/1   128  2000      T   F    ROOT      FORWARDING 0          8000000480bb9876

MSTP Instance 1 - VLANs: 2
-----
Bridge          Max RegionalRoot      IntPath  Designated      Root  Root
Identifier      Hop Bridge            Cost      Bridge            Port  Hop
hex             cnt hex             Cost      hex             cnt
8001000cdb80af01 20  8001000cdb80af01 0          8001000cdb80af01 Root  20

Port  Pri  PortPath  Role      State      Designa-  Designated
Num    Cost    Mac              State      ted cost   bridge
3/1   128  2000      MASTER    FORWARDING 0          8001000cdb80af01
```

Syntax: `show mstp <instance-number>`

The `<instance-number>` variable specifies the MSTP instance that you want to display information for.

TABLE 202 Output from Show MSTP

Field	Description
MSTP Instance	The ID of the MSTP instance whose statistics are being displayed. For the CIST, this number is 0.
VLANs	The number of VLANs that are included in this instance of MSTP. For the CIST this number will always be 1.
Bridge Identifier	The MAC address of the bridge.
Bridge MaxAge sec	Displays configured Max Age.
Bridge Hello sec	Displays configured Hello variable.
Bridge FwdDly sec	Displays configured FwdDly variable.
Bridge Hop cnt	Displays configured Max Hop count variable.
Root MaxAge sec	Max Age configured on the root bridge.
Root Hello sec	Hello interval configured on the root bridge.

TABLE 202 Output from Show MSTP (Continued)

Field	Description
Root FwdDly sec	FwdDly interval configured on the root bridge.
Root Hop Cnt	Current hop count from the root bridge.
Root Bridge	Bridge identifier of the root bridge.
ExtPath Cost	The configured path cost on a link connected to this port to an external MSTP region.
Regional Root Bridge	The Regional Root Bridge is the MAC address of the Root Bridge for the local region.
IntPath Cost	The configured path cost on a link connected to this port within the internal MSTP region.
Designated Bridge	The MAC address of the bridge that sent the best BPDU that was received on this port.
Root Port	Port indicating shortest path to root. Set to "Root" if this bridge is the root bridge.
Port Num	The port number of the interface.
Pri	The configured priority of the port. The default is 128.
PortPath Cost	Configured or auto detected path cost for port.
P2P Mac	Indicates if the port is configured with a point-to-point link: <ul style="list-style-type: none"> • T – The port is configured in a point-to-point link • F – The port is not configured in a point-to-point link
Edge	Indicates if the port is configured as an operational edge port: <ul style="list-style-type: none"> • T – indicates that the port is defined as an edge port. • F – indicates that the port is not defined as an edge port
Role	The current role of the port: <ul style="list-style-type: none"> • Master • Root • Designated • Alternate • Backup • Disabled
State	The port current spanning tree state. A port can have one of the following states: <ul style="list-style-type: none"> • Forwarding • Discarding • Learning • Disabled
Designated Cost	Port path cost to the root bridge.
Max Hop cnt	The maximum hop count configured for this instance.
Root Hop cnt	Hop count from the root bridge.

Displaying MSTP information for a specified instance

The following example displays MSTP information specified for an MSTP instance.

```
Brocade#show mstp 1
MSTP Instance 1 - VLANs: 2
-----
Bridge          Max RegionalRoot    IntPath   Designated    Root    Root
Identifier      Hop Bridge          Cost       Bridge        Port    Hop
hex             cnt hex              hex           hex           cnt
8001000cdb80af01 20 8001000cdb80af01 0           8001000cdb80af01 Root    20

Port  Pri PortPath  Role          State        Designa-  Designated
Num    Cost                                     ted cost  bridge
3/1   128 2000          MASTER       FORWARDING  0         8001000cdb80af01
```

Refer to [Table 202](#) for details about the display parameters.

Displaying MSTP information for CIST instance 0

Instance 0 is the Common and Internal Spanning Tree Instance (CIST). When you display information for this instance there are some differences with displaying other instances. The following example displays MSTP information for CIST Instance 0.

```
Brocade#show mstp 0
MSTP Instance 0 (CIST) - VLANs: 1
-----
Bridge          Bridge Bridge Bridge Bridge Root   Root   Root   Root
Identifier      MaxAge Hello FwdDly Hop   MaxAge Hello FwdDly Hop
hex             sec   sec   sec   cnt   sec   sec   sec   cnt
8000000cdb80af01 20    2    15   20    20    2    15   19
Root           ExtPath RegionalRoot    IntPath   Designated    Root
Bridge         Cost       Bridge          Cost       Bridge        Port
hex            hex              hex           hex
8000000480bb9876 2000      8000000cdb80af01 0           8000000480bb9876 3/1
Port  Pri PortPath  P2P Edge Role   State        Designa-  Designated
Num    Cost      Mac Port          ted cost  bridge
3/1   128 2000      T   F   ROOT    FORWARDING  0         8000000480bb9876
```

To display details about the MSTP configuration, enter the following command.

```
Brocade#show mstp conf
MSTP CONFIGURATION
-----
Name       : Reg1
Revision   : 1
Version    : 3 (MSTP mode)
Status     : Started

Instance VLANs
-----
0          4093
```

802.1s Multiple Spanning Tree Protocol

To display details about the MSTP that is configured on the device, enter the following command.

```
Brocade#show mstp detail
MSTP Instance 0 (CIST) - VLANs: 4093
-----
Bridge: 800000b000c00000 [Priority 32768, SysId 0, Mac 00b000c00000]
FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 6/54 - Role: DESIGNATED - State: FORWARDING
PathCost 20000, Priority 128, OperEdge T, OperPt2PtMac F, Boundary T
Designated - Root 800000b000c00000, RegionalRoot 800000b000c00000,
Bridge 800000b000c00000, ExtCost 0, IntCost 0
ActiveTimers - helloWhen 1
MachineState - PRX-DISCARD, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
PRT-ACTIVE_PORT, PST-FORWARDING, TCM-INACTIVE
BPDUs - Rcvd MST 0, RST 0, Config 0, TCN 0
Sent MST 6, RST 0, Config 0, TCN 0
```

Refer to [Table 202](#) for explanation about the parameters in the output.

Syntax: `show mstp [<mstp-id> | configuration | detail] [| begin <string> | exclude <string> | include <string>]`

Enter an MSTP ID for <mstp-id>.

Base Layer 3 and Routing Protocols

Table 203 lists the individual Brocade FastIron switches and the base Layer 3 features they support. These features are supported in the base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 203 Supported base Layer 3 features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
Static IP routing	Yes	Yes	Yes	Yes	Yes
Layer 3 system parameter limits	Yes	Yes	Yes	Yes	Yes
Static ARP entries	Yes (up to 6,000)	No	Yes (up to 1,000)	Yes (up to 1,000)	Yes (up to 1,024)
RIP V1 and V2 (Static RIP support only in the base Layer 3 image. The Brocade device with base Layer 3 does not learn RIP routes from other Layer 3 devices. However, the device does advertise directly connected routes.)	Yes	Yes	Yes	Yes	Yes
Redistribution of IP static routes into RIP	Yes	Yes	Yes	Yes	Yes
RIP default route learning	Yes	Yes	Yes	Yes	Yes
Route loop prevention: <ul style="list-style-type: none"> • Split horizon • Poison reverse 	Yes	Yes	Yes	Yes	Yes
Route-only support (supported with edge Layer 3 and full Layer 3 images only)	Yes	Yes	Yes	Yes	Yes

NOTE

ICX 6430 devices does not support Layer 3 features. RIP and OSPF are available as part of Premium license.

NOTE

Layer 2 with base Layer 3 images provide static RIP support. The device does not learn RIP routes from other Layer 3 devices. However, the device does advertise directly connected routes and can be configured to dynamically learn default routes. Brocade recommends that you deploy these devices only at the edge of your network, because incoming traffic can learn directly connected routes advertised by the Brocade device, but outgoing traffic to other devices must use statically configured or default routes.

TCAM entries in FWS devices

The size of the TCAM in FWS devices is limited to 1024 routing entries. When this limit is reached, the following warning message is displayed.

```
No free TCAM entry available. System will be unstable. Please reboot system.
```

At the same time, the following syslog message is generated.

```
System: No Free Tcam Entry available. System will be unstable.
```

You must reboot the device when you see these messages.

Adding a static IP route

To add a static IP route, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#ip route 209.157.2.0 255.255.255.0 192.168.2.1
```

This command adds a static IP route to the 209.157.2.x/24 subnet.

Syntax: [no] ip route <dest-ip-addr> <dest-mask> <next-hop-ip-addr> [<metric>] [tag <num>]

or

Syntax: [no] ip route <dest-ip-addr>/<mask-bits> <next-hop-ip-addr> [<metric>] [tag <num>]

On FastIron X Series devices, you can add up to 2048 static IP routes.

The <dest-ip-addr> variable is the route destination. The <dest-mask> variable is the network mask for the route destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24. To configure a default route, enter 0.0.0.0 for <dest-ip-addr> and 0.0.0.0 for <dest-mask> (or 0 for the <mask-bits> variable if you specify the address in CIDR format). Specify the IP address of the default gateway using the <next-hop-ip-addr> variable.

The <next-hop-ip-addr> variable is the IP address of the next hop router (gateway) for the route.

The <metric> variable specifies the cost of the route and can be a number from 1 through 16. The default is 1. The metric is used by RIP. If you do not enable RIP, the metric is not used.

The tag <num> parameter specifies the tag value of the route. The possible value is from 0 through 4294967295. The default value is 0.

NOTE

You cannot specify **null0** or another interface as the next hop in the base Layer 3 image.

Adding a static ARP entry

NOTE

Adding a static ARP entry is supported on FastIron X Series and Brocade FCX Series devices. It is *not* supported on FastIron WS Series devices.

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the Brocade device, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the Brocade device receives an ARP request from the device that has the entry address. The software places a static ARP entry into the ARP cache as soon as you create the entry.

To add a static ARP entry, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#arp 1 209.157.22.3 aaaa.bbbb.cccc ethernet 3
```

This command adds a static ARP entry that maps IP address 209.157.22.3 to MAC address aaaa.bbbb.cccc. The entry is for a MAC address connected to Brocade port 3.

Syntax: **[no]** **arp** *<num>* *<ip-addr>* *<mac-addr>* **ethernet** *<port>*

The *<num>* variable specifies the entry number. You can specify a number from 1 up to the maximum number of static entries allowed on the device. You can allocate more memory to increase this amount. To do so, enter the **system-max ip-static-arp** *<num>* command at the global CONFIG level of the CLI.

The *<ip-addr>* variable specifies the IP address of the device that has the MAC address of the entry.

The *<mac-addr>* variable specifies the MAC address of the entry.

The **ethernet** *<port>* parameter specifies the port number attached to the device that has the MAC address of the entry. Specify the *<port>* variable in one of the following formats:

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **clear arp** command clears learned ARP entries but does not remove any static ARP entries.

Modifying and displaying Layer 3 system parameter limits

This section shows how to view and configure some of the Layer 3 system parameter limits.

Layer 3 configuration notes

- Changing the system parameters reconfigures the device memory. Whenever you reconfigure the memory on a Brocade device, you must save the change to the startup-config file, and then reload the software to place the change into effect.
- The Layer 3 system parameter limits for FastIron IPv6 models are automatically adjusted by the system and cannot be manually modified. Refer to [“FastIron second generation modules”](#) on page 1190.

FWS with base Layer 3

NOTE

This section applies to the FastIron WS base Layer 3 code. It does not apply to the FastIron X Series and Brocade FCX Series devices.

The base Layer 3 code uses ternary content-addressable memory (TCAM) for IP routing, access lists, MAC-based VLANs, and other features. The default value for TCAM is 256 routes.

You can set the TCAM number reserved for IP routing using the **system-max** command.

```
Brocade(config)#system-max hw-ip-route-tcam 450
```

Syntax: [no] **system-max hw-ip-route-tcam** <num>

The <num> variable specifies a value from 64 through 1020. The default is 256.

The **show ip route** command displays the usage of TCAM in routing.

```
Brocade#show ip route
Total number of IP routes: 29
Route TCAM: total 64, used 64, add HW route failure=34

Start index: 1  D:Connected  R:RIP  S:Static  O:OSPF  *:Candidate default
      Destination      NetMask      Gateway      Port      Cost      Type
1      2.2.10.0          255.255.255.0  0.0.0.0      v10        1          D
2      2.2.11.0          255.255.255.0  0.0.0.0      v11        1          D
...
```

In this example, *add HW route failure* shows the number of failures in installing routing TCAM.

The base Layer 3 code installs routing TCAM for the interface subnet and static routes. A packet is routed in hardware if its destination IP address matches the interface or stack route subnet, and the next hop or destination ARP is resolved. The unknown unicast packets have destination IP addresses that do not match any static route or interface subnets.

The handling of unknown unicast packets differs depending on TCAM availability. For example, for a system that never encounters an out-of-TCAM situation, If the default route is not configured or its next hop ARP is not resolved, all unknown unicast packets are handled based on the default Layer 2 behavior. They are either dropped in hardware if **route-only** is configured or VLAN-flooded in hardware.

If the default route is configured and its next hop ARP is resolved, unknown unicast packets are hardware-routed to the next hop, and not VLAN-flooded.

Once the device runs out of TCAM, it traps the unknown unicast packets to the CPU for processing. If the default route is defined and its next hop ARP is resolved, the packets are routed by the CPU. Otherwise, they follow the default Layer 2 behavior. Because the base Layer 3 code does not allow **route-only** configurations, these packets are VLAN-flooded.

The system does not change this CPU-handling behavior back to hardware switching even when TCAM again becomes available.

FastIron first generation modules

You can configure the following Layer 3 system parameters on FastIron X Series first generation modules:

- Number of IP next hops and IP route entries
- Number of hardware logical interfaces (physical port and VLAN pairs)
- Number of multicast output interfaces (clients)

These parameters are automatically enabled with pre-defined default values. You can, however, adjust these values to conform with your network topology.

To display the current settings for the Layer 3 system parameters, use the **show default value** command. Refer to [“Displaying Layer 3 system parameter limits”](#) on page 1191.

To modify the default settings for the Layer 3 system parameters, use the **system max** command at the global CONFIG level of the CLI. Refer to [“Modifying Layer 3 system parameter limits on first generation modules”](#) on page 1189.

Modifying Layer 3 system parameter limits on first generation modules

NOTE

The commands in this section are supported on FastIron X Series devices only.

The Layer 3 system parameter limits share the same hardware memory space and, by default, consume all of the hardware memory allocated for these Layer 3 limits. Therefore, to increase the limit for one of the parameters, you must first decrease one or both of the other parameters' limits. If you enter a value that exceeds the memory limit, an error message displays and the configuration will not take effect.

For example, if the network topology has a smaller number of IP next hops and routes, but has numerous multicast output interfaces, you could decrease the number of IP next hops and routes, and then increase the number of multicast output interfaces. To do so, enter commands such as the following.

```
Brocade(config)#system-max hw-ip-next-hop 1024
Brocade(config)#system-max hw-ip-mcast-mll 2048
Brocade(config)#write memory
Brocade(config)#reload
```

Likewise, if the network topology does not have a large number of VLANs, and the VLANs configured on physical ports are not widely distributed, you could decrease the number of hardware logical interfaces, and then increase the number of IP next hops and multicast output interfaces. To do so, enter commands such as the following.

```
Brocade(config)#system-max hw-logical-interface 2048
Brocade(config)#system-max hw-ip-next-hop 3072
Brocade(config)#system-max hw-ip-mcast-mll 2048
Brocade(config)#write memory
Brocade(config)#reload
```

Syntax: **system-max hw-ip-next-hop** *<num>*

Syntax: **system-max hw-logical-interface** *<num>*

Syntax: **system-max hw-ip-mcast-mll** *<num>*

NOTE

The **system-max** commands are not supported on IPv6 devices. Refer to [“FastIron second generation modules”](#) on page 1190.

The **hw-ip-next-hop** *<num>* parameter specifies the maximum number of IP next hops and routes supported on the device. Note that the maximum number includes unicast next hops and multicast route entries. Enter a number from 100 through 6144. The default is 2048.

You can define the maximum number of hops on FastIron X Series devices using the **hw-ip-next-hop** *<num>* parameter with the following first generation modules installed:

- SX-FI424F
- SX-FI424C
- SX-FI424P
- SX-FI424HF
- SX-FI42XG
- SX-FI42XGW

If these modules are not installed, then the maximum number of hops is automatically set and is not configurable.

The **hw-logical-interface** *<num>* parameter specifies the number of hardware logical interface pairs (physical port and VLAN pairs) supported on the device. Enter a number from 0 through 4096. When this parameter is set to 4096 (the maximum), the limit is not enforced. If you enter a number less than 4096, the limit is the total number of physical port and VLAN pairs that are IP-enabled in the system. The default is 4096.

The **hw-ip-mcast-mll** *<num>* parameter specifies the maximum number of multicast output interfaces (clients) supported on the device. If a given source or group has clients in *n* tagged VLANs on the router, then *n* entries are consumed for that source or group entry. Enter a number from 0 through 4096. The default is 1024.

FastIron second generation modules

FastIron IPv6 models support the same Layer 3 system parameters that use hardware memory as do FastIron IPv4 models. However, there are some configuration differences between second generation modules and first generation modules. The differences are as follows:

- Number of IP next hops and IP route entries – 6144 maximum and default value. The system automatically calculates this value, based on the maximum number of VLANs supported systemwide.

- Number of hardware logical interfaces (physical port and VLAN pairs) – This value is the same as the maximum number of VLANs supported systemwide, so it is not configurable nor displayed in the **show default value** output in second generation modules.
- Number of multicast output interfaces (clients) – 3072 maximum. This value is fixed in second generation modules and cannot be modified. This system parameter occupies its own hardware memory space.

To display the current settings for the Layer 3 system parameters, use the **show default value** command. Refer to [“Displaying Layer 3 system parameter limits”](#) on page 1191.

FastIron third generation modules

The default value of next hop entries on FastIron X Series devices with the following third generation modules installed is 16384. This value is predefined and not editable.

- SX-FI48GPP
- SX-FI-2XG
- SX-FI-8XG
- SX-FI-24HF
- SX-FI-24GPP

If the FastIron X Series device is installed with first generation or second generation modules, the system automatically calculates the default value for these modules.

Displaying Layer 3 system parameter limits

To display the Layer 3 system parameter defaults, maximum values, and current values, enter the **show default value** command at any level of the CLI.

The following example shows output on a FastIron X Series with first generation modules.

```
Brocade#show default value
```

```
sys log buffers:50          mac age time:300 sec          telnet sessions:5
ip arp age:10 min          bootp relay max hops:4      ip ttl:64 hops
ip addr per intf:24
igmp group memb.:140 sec   igmp query:60 sec
ospf dead:40 sec          ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec
```

System Parameters	Default	Maximum	Current
ip-arp	4000	64000	4000
ip-static-arp	512	1024	512

some lines omitted for brevity....

hw-ip-next-hop	2048	6144	2048
hw-logical-interface	4096	4096	4096
hw-ip-mcast-ml1	1024	4096	1024

Modifying and displaying Layer 3 system parameter limits

The following example shows output on a FastIron X Series with second generation modules.

```
Brocade#show default value

sys log buffers:50          mac age time:300 sec          telnet sessions:5

ip arp age:10 min          bootp relay max hops:4      ip ttl:64 hops
ip addr per intf:24

igmp group memb.:140 sec   igmp query:60 sec

ospf dead:40 sec           ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec

System Parameters   Default   Maximum   Current
ip-arp              4000     64000    4000
ip-static-arp       512      1024     512

some lines omitted for brevity....

hw-ip-next-hop      6144     6144     6144
hw-ip-mcast-mll     1024     4096     1024
hw-traffic-condition 50        1024     50
```

The following example shows output on a FastIron X Series with third generation modules.

```
Brocade#show default value

sys log buffers:50          mac age time:300 sec          telnet sessions:5

ip arp age:10 min          bootp relay max hops:4      ip ttl:64 hops
ip addr per intf:24

igmp group memb.:140 sec   igmp query:60 sec

ospf dead:40 sec           ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec

System Parameters   Default   Maximum   Current
ip-arp              4000     64000    4000
ip-static-arp       512      1024     512

some lines omitted for brevity....

hw-ip-next-hop      6144     6144     6144
hw-ip-mcast-mll     1024     4096     1024
hw-traffic-condition 50        1024     50
```

Configuring RIP

If you want the Brocade device to use Routing Information Protocol (RIP), you must enable the protocol globally, and then enable RIP on individual ports. When you enable RIP on a port, you also must specify the version (version 1 only, version 2 only, or version 1 compatible with version 2).

Optionally, you also can set or change the following parameters:

- Route redistribution – You can enable the software to redistribute static routes from the IP route table into RIP. Redistribution is disabled by default.
- Learning of default routes – The default is disabled.
- Loop prevention (split horizon or poison reverse) – The default is poison reverse.

Enabling RIP

RIP is disabled by default. You must enable the protocol both globally and on the ports on which you want to use RIP.

To enable RIP globally, enter the following command.

```
Brocade(config)#router rip
```

Syntax: [no] router rip

To enable RIP on a port and specify the RIP version, enter commands such as the following.

```
Brocade(config-rip-router)#interface ethernet 1
Brocade(config-if-e1000-1)#ip rip v1-only
```

These commands change the CLI to the configuration level for port 1 and enable RIP version 1 on the interface. You must specify the version.

Syntax: interface ethernet <port>

Syntax: [no] ip rip v1-only | v1-compatible-v2 | v2-only

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Enabling redistribution of IP static routes into RIP

By default, the software does not redistribute the IP static routes in the route table into RIP. To configure redistribution, perform the following tasks.

1. Configure redistribution filters (optional).

You can configure filters to permit or deny redistribution for a route based on the route metric. You also can configure a filter to change the metric. You can configure up to 64 redistribution filters. The software uses the filters in ascending numerical order and immediately takes the action specified by the filter. Thus, if filter 1 denies redistribution of a given route, the software does not redistribute the route, regardless of whether a filter with a higher ID permits redistribution of that route.

NOTE

The default redistribution action is permit, even after you configure and apply a permit or deny filter. To deny redistribution of specific routes, you must configure a deny filter.

NOTE

The option to set the metric is not applicable to static routes.

2. Enable redistribution.

NOTE

If you plan to configure redistribution filters, do not enable redistribution until you have configured the filters.

When you enable redistribution, all types of routes are redistributed into RIP; redistribution is not limited to IP static routes. If you want to deny certain routes from being redistributed into RIP, configure deny filters for those routes before you enable redistribution. You can configure up to 64 RIP redistribution filters. They are applied in ascending numerical order.

NOTE

The default redistribution action is permit, even after you configure and apply redistribution filters to the port. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (filter ID 64), and then apply filters with lower filter IDs to allow specific routes.

Configuring a redistribution filter

To configure a redistribution filter, enter a command such as the following.

```
Brocade(config-rip-router)#deny redistribute 1 static address 207.92.0.0
255.255.0.0
```

This command denies redistribution of all 207.92.x.x IP static routes.

Syntax: [no] permit | deny redistribute <filter-num> static address <ip-addr> <ip-mask>
[match-metric <value> | set-metric <value>]

The <filter-num> variable specifies the redistribution filter ID. Specify a number from 1 through 64. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **static address** <ip-addr> <ip-mask> parameters apply redistribution to the specified network and subnet address. Use 0 to specify “any”. For example, “207.92.0.0 255.255.0.0” means “any 207.92.x.x subnet”. However, to specify any subnet (all subnets match the filter), enter **static address 255.255.255.255 255.255.255.255**.

The **match-metric** <value> parameter applies redistribution to those routes with a specific metric value. Possible values are from 1 through 15.

The **set-metric** <value> parameter sets the RIP metric value that will be applied to the routes imported into RIP.

NOTE

The **set-metric** parameter does not apply to static routes.

The following command denies redistribution of a 207.92.x.x IP static route only if the route metric is 5.

```
Brocade(config-rip-router)#deny redistribute 2 static address 207.92.0.0
255.255.0.0 match-metric 5
```

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x.

```
Brocade(config-rip-router)#deny redistribute 64 static address 255.255.255.255
255.255.255.255
Brocade(config-rip-router)#permit redistribute 1 static address 10.10.10.0
255.255.255.0
Brocade(config-rip-router)#permit redistribute 2 static address 20.20.20.0
255.255.255.0
```

Enabling redistribution

After you configure redistribution parameters, you must enable redistribution.

To enable RIP redistribution, enter the following command.

```
Brocade(config-rip-router)#redistribution
```

Syntax: [no] redistribution

Enabling learning of default routes

By default, the software does not learn RIP default routes.

To enable learning of default RIP routes, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/1
Brocade(config-if-e1000-1)#ip rip learn-default
```

Syntax: [no] ip rip learn-default

Changing the route loop prevention method

RIP can use the following methods to prevent routing loops:

- Split horizon – The Brocade device does not advertise a route on the same interface as the one on which it learned the route.
- Poison reverse – The Brocade device assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which it learned the route. This is the default.

NOTE

These methods are in addition to the RIP maximum valid route cost of 15.

To enable split horizon, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/1
Brocade(config-if-e1000-1)#no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

Other Layer 3 protocols

For information about other IP configuration commands in the Layer 2 with base Layer 3 image that are not included in this chapter, refer to [Chapter 26, “IP Configuration”](#).

For information about enabling or disabling Layer 3 routing protocols, refer to [“Enabling or disabling routing protocols”](#) on page 1196.

Enabling or disabling routing protocols

This section describes how to enable or disable routing protocols. For complete configuration information about the routing protocols, refer to [“RIP overview”](#) on page 1199.

The full Layer 3 code supports the following protocols:

- BGP4
- DVMRP
- IGMP
- IP
- IP multicast (PIM-SM, PIM-DM)
- OSPF

- PIM
- RIPV1 and V2
- VRRP
- VRRP-E
- VSRP

IP routing is enabled by default on devices running Layer 3 code. All other protocols are disabled, so you must enable them to configure and use them.

To enable a protocol on a device running full Layer 3 code, enter **router** at the global CONFIG level, followed by the protocol to be enabled. The following example shows how to enable OSPF.

```
Brocade(config)#router ospf
```

Syntax: router bgp | dvmrp | igmp | ip | ospf | pim | rip | vrrp | vrrp-e | vsrp

Enabling or disabling Layer 2 switching

By default, Brocade Layer 3 switches support Layer 2 switching. These devices modify the routing protocols that are not supported on the devices. If you want to disable Layer 2 switching, you can do so globally or on individual ports, depending on the version of software your device is running.

NOTE

Consult your reseller or Brocade to understand the risks involved before disabling all Layer 2 switching operations.

Configuration notes and feature limitations for Layer 2 switching

- Enabling or disabling Layer 2 switching is supported in the edge Layer 3 and full Layer 3 software images only.
- FWS EPREM devices support disabling Layer 2 switching at the global CONFIG level only.
- FastIron X Series and Brocade FCX Series devices support disabling Layer 2 switching at the interface configuration level as well as the global CONFIG level.
- Enabling or disabling Layer 2 switching is not supported on virtual interfaces.

Command syntax for Layer 2 switching

To globally disable Layer 2 switching on a Layer 3 switch, enter commands such as the following.

```
Brocade(config)#route-only
Brocade(config)#exit
Brocade#write memory
Brocade#reload
```

To re-enable Layer 2 switching on a Layer 3 switch, enter the following commands.

Enabling or disabling Layer 2 switching

```
Brocade(config)#no route-only
Brocade(config)#exit
Brocade#write memory
Brocade#reload
```

Syntax: [no] route-only

To disable Layer 2 switching only on a specific interface, go to the interface configuration level for that interface, and then disable the feature. The following commands show how to disable Layer 2 switching on port 2.

```
Brocade(config)#interface ethernet 2
Brocade(config-if-e1000-2)#route-only
```

RIP (IPv4)

Table 204 lists the individual Brocade FastIron switches and the Routing Information Protocol (RIP) for IPv4 features they support. These features are supported in the edge Layer 3 and full Layer 3 software images.

TABLE 204 Supported RIP features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
RIP V1 and V2	Yes	Yes	Yes	Yes	Yes
Route learning and advertising	Yes	Yes	Yes	Yes	Yes
Route redistribution into RIP	Yes	Yes	Yes	Yes	Yes
Route metrics	Yes	Yes	Yes	Yes	Yes
Route loop prevention: <ul style="list-style-type: none"> • Poison reverse • Split horizon 	Yes	Yes	Yes	Yes	Yes
RIP route advertisement suppression on a VRRP or VRRP-E backup interface	Yes	Yes	Yes	Yes	Yes
Route filters	Yes	Yes	Yes	Yes	Yes
CPU utilization statistics for RIP	Yes	Yes	Yes	Yes	Yes

RIP overview

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a *distance vector* (a number representing a distance) to measure the cost of a given route. The cost is a distance vector because the cost often is equivalent to the number of router hops between the Brocade Layer 3 Switch and the destination network.

A Brocade Layer 3 Switch can receive multiple paths to a destination. The software evaluates the paths, selects the best path, and saves the path in the IP route table as the route to the destination. Typically, the best path is the path with the fewest hops. A hop is another router through which packets must travel to reach the destination. If the Brocade Layer 3 Switch receives a RIP update from another router that contains a path with fewer hops than the path stored in the Brocade Layer 3 Switch route table, the Layer 3 Switch replaces the older route with the newer one. The Layer 3 Switch then includes the new path in the updates it sends to other RIP routers, including Brocade Layer 3 Switches.

RIP routers, including the Brocade Layer 3 Switch, also can modify a route cost, generally by adding to it, to bias the selection of a route for a given destination. In this case, the actual number of router hops may be the same, but the route has an administratively higher cost and is thus less likely to be used than other, lower-cost routes.

A RIP route can have a maximum cost of 15. Any destination with a higher cost is considered unreachable. Although limiting to larger networks, the low maximum hop count prevents endless loops in the network.

Brocade Layer 3 Switches support the following RIP versions:

- Version (V1)
- V1 compatible with V2
- Version (V2) (the default)

ICMP host unreachable message for undeliverable ARPs

If the router receives an ARP request packet that it is unable to deliver to the final destination because of the ARP timeout and no ARP response is received (the router knows of no route to the destination address), the router sends an ICMP Host Unreachable message to the source.

RIP parameters and defaults

The following tables list the RIP parameters, their default values, and where to find configuration information.

RIP global parameters

[Table 205](#) lists the global RIP parameters and their default values, and indicates where you can find configuration information.

TABLE 205 RIP global parameters

Parameter	Description	Default	Reference
RIP state	The global state of the protocol. NOTE: You also must enable the protocol on individual interfaces. Globally enabling the protocol does not allow interfaces to send and receive RIP information. Refer to Table 206 on page 1201.	Disabled	page 1202
Administrative distance	The administrative distance is a numeric value assigned to each type of route on the router. When the router is selecting from among multiple routes (sometimes of different origins) to the same destination, the router compares the administrative distances of the routes and selects the route with the lowest administrative distance. This parameter applies to routes originated by RIP. The administrative distance stays with a route when it is redistributed into other routing protocols.	120	page 1204
Redistribution	RIP can redistribute routes from other routing protocols such as OSPF and BGP4 into RIP. A redistributed route is one that a router learns through another protocol, then distributes into RIP.	Disabled	page 1205

TABLE 205 RIP global parameters (Continued)

Parameter	Description	Default	Reference
Redistribution metric	RIP assigns a RIP metric (cost) to each external route redistributed from another routing protocol into RIP. An external route is a route with at least one hop (packets must travel through at least one other router to reach the destination). This parameter applies to routes that are redistributed from other protocols into RIP.	1 (one)	page 1206
Update interval	How often the router sends route updates to its RIP neighbors.	30 seconds	page 1207
Learning default routes	The router can learn default routes from its RIP neighbors. NOTE: You also can enable or disable this parameter on an individual interface basis. Refer to Table 206 on page 1201.	Disabled	page 1207
Advertising and learning with specific neighbors	The Layer 3 Switch learns and advertises RIP routes with all its neighbors by default. You can prevent the Layer 3 Switch from advertising routes to specific neighbors or learning routes from specific neighbors.	Learning and advertising permitted for all neighbors	page 1208

RIP interface parameters

[Table 206](#) lists the interface-level RIP parameters and their default values, and indicates where you can find configuration information.

TABLE 206 RIP interface parameters

Parameter	Description	Default	Reference
RIP state and version	The state of the protocol and the version that is supported on the interface. The version can be one of the following: <ul style="list-style-type: none"> Version 1 only Version 2 only Version 1, but also compatible with version 2 NOTE: You also must enable RIP globally.	Disabled	page 1202
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	page 1203
Learning default routes	Locally overrides the global setting. Refer to Table 205 on page 1200.	Disabled	page 1207

TABLE 206 RIP interface parameters (Continued)

Parameter	Description	Default	Reference
Loop prevention	<p>The method a router uses to prevent routing loops caused by advertising a route on the same interface as the one on which the router learned the route.</p> <ul style="list-style-type: none"> Split horizon – The router does not advertise a route on the same interface as the one on which the router learned the route. Poison reverse – The router assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the router learned the route. 	<p>Poison reverse</p> <p>NOTE: Enabling split horizon disables poison reverse on the interface.</p>	page 1209
Advertising and learning specific routes	You can control the routes that a Layer 3 Switch learns or advertises.	The Layer 3 Switch learns and advertises all RIP routes on all interfaces.	page 1210

RIP parameter configuration

Use the following procedures to configure RIP parameters on a system-wide and individual interface basis.

Enabling RIP

RIP is disabled by default. To enable it, use the following procedure.

NOTE

You must enable the protocol globally and also on individual interfaces on which you want to advertise RIP. Globally enabling the protocol does not enable it on individual interfaces.

To enable RIP globally, enter the **router rip** command.

```
Brocade(config)#router rip
```

Syntax: [no] router rip

After globally enabling the protocol, you must enable it on individual interfaces. You can enable the protocol on physical interfaces as well as virtual routing interfaces. To enable RIP on an interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/1
Brocade(config-if-0/1/1)#ip rip v1-only
```

Syntax: [no] ip rip v1-only | v1-compatible-v2 | v2-only

NOTE

You must specify the RIP version.

Enabling ECMP for routes in RIP

ECMP for routes in RIP is disabled by default. Use the **ecmp-enable** command to enable the feature at the router rip level.

```
Brocade(config-rip-router)#ecmp-enable
```

Syntax: [no] **ecmp-enable**

Configuring metric parameters

By default, a Brocade Layer 3 Switch port increases the cost of a RIP route that is learned on the port by one. You can configure individual ports to add more than one to a learned route cost. In addition, you can configure a RIP offset list to increase the metric for learned or advertised routes based on network address.

Changing the cost of routes learned on a port

By default, a Brocade Layer 3 Switch port increases the cost of a RIP route that is learned on the port. The Layer 3 Switch increases the cost by adding one to the route metric before storing the route.

You can change the amount that an individual port adds to the metric of RIP routes learned on the port. To do so, use the following method.

NOTE

RIP considers a route with a metric of 16 to be unreachable. Use this metric only if you do not want the route to be used. You can prevent the Layer 3 Switch from using a specific port for routes learned through that port by setting its metric to 16.

To increase the cost a port adds to RIP routes learned in that port, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/6/1
Brocade(config-if-0/6/1)#ip metric 5
```

These commands configure port 6/1 to add 5 to the cost of each route learned on the port.

Syntax: **ip metric** <1-16>

Configuring a RIP offset list

A RIP offset list allows you to add to the metric of specific inbound or outbound routes learned or advertised by RIP. RIP offset lists provide a simple method for adding to the cost of specific routes and therefore biasing the Layer 3 Switch route selection away from those routes.

A RIP offset list consists of the following parameters:

- An access control list (ACL) that specifies the routes to which to add the metric.
- The direction:
 - In applies to routes the Layer 3 Switch learns from RIP neighbors.
 - Out applies to routes the Layer 3 Switch is advertising to its RIP neighbors.
- The type and number of a specific port to which the RIP offset list applies (optional).

The software adds the offset value to the routing metric (cost) of the routes that match the ACL. If a route matches both a global offset list and an interface-based offset list, the interface-based offset list takes precedence. The interface-based offset list metric is added to the route in this case.

You can configure up to 24 global RIP offset lists and up to 24 RIP offset lists on each interface.

To configure a global RIP offset list, enter commands such as the following.

```
Brocade(config)#access-list 21 deny 160.1.0.0 0.0.255.255
Brocade(config)#access-list 21 permit any
Brocade(config)#router rip
Brocade(config-rip-router)#offset-list 21 out 10
```

The commands in this example configure a standard ACL. The ACL matches on all IP networks except 160.1.x.x. When the Layer 3 Switch advertises a route that matches ACL 21, the offset list adds 10 to the route metric.

Syntax: `[no] offset-list <ACL-number-or-name> in | out <metric> [ethernet <port>]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

In the following example, the Layer 3 Switch uses ACL 21 to add 10 to the metric of routes received on Ethernet port 0/2/1.

```
Brocade(config-rip-router)#offset-list 21 in 10 ethernet 0/2/1
```

Changing the administrative distance

By default, the Layer 3 Switch assigns the default RIP administrative distance (120) to RIP routes. When comparing routes based on administrative distance, the Layer 3 Switch selects the route with the lower distance. You can change the administrative distance for RIP routes.

NOTE

Refer to [“Changing administrative distances”](#) on page 1374 for the default distances for all route sources.

To change the administrative distance for RIP routes, enter the **distance** command.

```
Brocade(config-rip-router)#distance 140
```

This command changes the administrative distance to 140 for all RIP routes.

Syntax: `[no] distance <num>`

The *<num>* variable specifies a range from 1 through 255.

Configuring redistribution

You can configure the Layer 3 Switch to redistribute routes learned through Open Shortest Path First (OSPF) or Border Gateway Protocol version 4 (BGP4) into RIP. When you redistribute a route from one of these other protocols into RIP, the Layer 3 Switch can use RIP to advertise the route to its RIP neighbors.

To configure redistribution, perform the following tasks:

1. Configure redistribution filters (optional). You can configure filters to permit or deny redistribution for a route based on its origin (OSPF, BGP4, and so on), the destination network address, and the route metric. You also can configure a filter to set the metric based on these criteria.
2. Change the default redistribution metric (optional). The Layer 3 Switch assigns a RIP metric of 1 to each redistributed route by default. You can change the default metric to a value up to 16.
3. Enable redistribution.

NOTE

Do not enable redistribution until you configure the other redistribution parameters.

Configuring redistribution filters

RIP redistribution filters apply to all interfaces. The software uses the filters in ascending numerical order and immediately takes the action specified by the filter. Thus, if filter 1 denies redistribution of a given route, the software does not redistribute the route, regardless of whether a filter with a higher ID would permit redistribution of that route.

NOTE

The default redistribution action is permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (the filter with the highest ID), and then apply filters with lower filter IDs to allow specific routes.

To configure a redistribution filter, enter a command such as the following.

```
Brocade(config-rip-router)#deny redistribute 2 all address 207.92.0.0 255.255.0.0
```

This command denies redistribution for all types of routes to the 207.92.x.x network.

Syntax: [no] permit | deny redistribute <filter-num> all | bgp | ospf | static address <ip-addr> <ip-mask> [match-metric <value> | set-metric <value>]

The <filter-num> variable specifies the redistribution filter ID. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **all** parameter applies redistribution to all route types.

The **bgp** parameter applies redistribution to BGP4 routes only.

The **ospf** parameter applies redistribution to OSPF routes only.

The **static** parameter applies redistribution to IP static routes only.

The **address** *<ip-addr> <ip-mask>* parameters apply redistribution to the specified network and subnet address. Use 0 to specify “any”. For example, “207.92.0.0 255.255.0.0” means “any 207.92.x.x subnet”. However, to specify any subnet (all subnets match the filter), enter “address 255.255.255.255 255.255.255.255”.

The **match-metric** *<value>* parameter applies the redistribution filter only to those routes with the specified metric value; possible values are from 1 through 15.

The **set-metric** *<value>* parameter sets the RIP metric value that will be applied to those routes imported into RIP.

The following command denies redistribution into RIP for all OSPF routes.

```
Brocade(config-rip-router)#deny redistribute 3 ospf address 207.92.0.0
255.255.0.0
```

The following command denies redistribution for all OSPF routes that have a metric of 10.

```
Brocade(config-rip-router)#deny redistribute 3 ospf address 207.92.0.0
255.255.0.0 match-metric 10
```

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x.

```
Brocade(config-rip-router)#deny redistribute 64 static address 255.255.255.255
255.255.255.255
Brocade(config-rip-router)#permit redistribute 1 static address 10.10.10.0
255.255.255.0
Brocade(config-rip-router)#permit redistribute 2 static address 20.20.20.0
255.255.255.0
```

NOTE

This example assumes that the highest RIP redistribution filter ID configured on the device is 64.

Changing the redistribution metric

When the Layer 3 Switch redistributes a route into RIP, the software assigns a RIP metric (cost) to the route. By default, the software assigns a metric of 1 to each route that is redistributed into RIP. You can increase the metric that the Layer 3 Switch assigns up to 15.

To change the RIP metric the Layer 3 Switch assigns to redistributed routes, enter a command such as the following.

```
Brocade(config-rip-router)#default-metric 10
```

This command assigns a RIP metric of 10 to each route that is redistributed into RIP.

Syntax: [no] default-metric *<1-15>*

Enabling redistribution

After you configure redistribution parameters, you need to enable redistribution.

To enable RIP redistribution, enter the **redistribution** command.

```
Brocade(config-rip-router)#redistribution
```

Syntax: [no] redistribution

The **no** form of this command disables RIP redistribution.

Removing a RIP redistribution deny filter

To remove a previously configured RIP redistribution deny filter, perform the following task:

1. Remove the RIP redistribution deny filter.
2. Disable the redistribution function.
3. Re-enable redistribution.

The following shows an example of how to remove a RIP redistribution deny filter.

```
Brocade(config-rip-router)#no deny redistribute 2 all address 207.92.0.0
255.255.0.0
Brocade(config-rip-router)#no redistribution
Brocade(config-rip-router)#redistribution
```

Route learning and advertising parameters

By default, a Brocade Layer 3 Switch learns routes from all its RIP neighbors and advertises RIP routes to those neighbors.

You can configure the following learning and advertising parameters:

- **Update interval** – The update interval specifies how often the Layer 3 Switch sends RIP route advertisements to its neighbors. You can change the interval to a value from 1 through 1000 seconds. The default is 30 seconds.
- **Learning and advertising of RIP default routes** – The Layer 3 Switch learns and advertises RIP default routes by default. You can disable learning and advertising of default routes on a global or individual interface basis.
- **Learning of standard RIP routes** – By default, the Layer 3 Switch learns RIP routes from all its RIP neighbors. You can configure RIP neighbor filters to explicitly permit or deny learning from specific neighbors.

Changing the update interval for route advertisements

The update interval specifies how often the Layer 3 Switch sends route advertisements to its RIP neighbors. You can specify an interval from 1 through 1000 seconds. The default is 30 seconds.

To change the RIP update interval, enter a command such as the following.

```
Brocade(config-rip-router)#update-time 120
```

This command configures the Layer 3 Switch to send RIP updates every 120 seconds.

Syntax: update-time <1-1000>

Enabling learning of RIP default routes

You can enable learning of RIP default routes on a global or individual interface basis.

To enable learning of default RIP routes on a global basis, enter the following command.

```
Brocade(config-rip-router)#learn-default
```

Syntax: [no] learn-default

To enable learning of default RIP routes on an individual interface basis, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/1
Brocade(config-if-0/1/1)#ip rip learn-default
```

Syntax: [no] ip rip learn-default

Configuring a RIP neighbor filter

By default, a Brocade Layer 3 Switch learns RIP routes from all its RIP neighbors. Neighbor filters allow you to specify the neighbor routers from which the Brocade device can receive RIP routes. Neighbor filters apply globally to all ports.

To configure a RIP neighbor filter, enter a command such as the following.

```
Brocade(config-rip-router)#neighbor 1 deny any
```

This command configures the Layer 3 Switch so that the device does not learn any RIP routes from any RIP neighbors.

Syntax: [no] neighbor <filter-num> permit | deny <source-ip-address> | any

The following commands configure the Layer 3 Switch to learn routes from all neighbors except 192.168.1.170. Once you define a RIP neighbor filter, the default action changes from learning all routes from all neighbors to denying all routes from all neighbors except the ones you explicitly permit. To deny learning from a specific neighbor but allow all other neighbors, you must add a filter that allows learning from all neighbors. Be sure to add the filter to permit all neighbors last (the one with the highest filter number). Otherwise, the software can match on the permit all filter instead of a filter that denies a specific neighbor, and learn routes from that neighbor.

```
Brocade(config-rip-router)#neighbor 2 deny 192.16.1.170
Brocade(config-rip-router)#neighbor 1024 permit any
```

Denying route advertisements for connected routes

By default, RIP advertises all connected routes to neighboring routers except for the management route. To configure the router to not advertise connected routes, use the **dont-advertise-connected** command. When the **dont-advertise-connected** command is configured, the router only sends RIP enabled interface routes.

To configure the **dont-advertise-connected** command under the router RIP configuration level, enter the **router rip** command.

```
FastIron(config)#router rip
FastIron(config-rip-router)#dont-advertise-connected
```

Syntax: [no] dont-advertise-connected

To disable the configuration, use the **no** form of the command.

Changing the route loop prevention method

RIP can use the following methods to prevent routing loops:

- Split horizon – The Layer 3 Switch does not advertise a route on the same interface as the one on which the router learned the route.
- Poison reverse – The Layer 3 Switch assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the router learned the route. This is the default.

These loop prevention methods are configurable on an individual interface basis. One of the methods is always in effect on an interface enabled for RIP. If you disable one method, the other method is enabled.

NOTE

These methods may be used in addition to the RIP maximum valid route cost of 15.

Disabling poison-reverse

To disable poison reverse and enable split horizon on an interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/1
Brocade(config-if-0/1/1)#no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

To disable split horizon and enable poison reverse on an interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/1
Brocade(config-if-0/1/1)#ip rip poison-reverse
```

Suppressing RIP route advertisement on a VRRP or VRRP-E backup interface

NOTE

This section applies only if you configure the Layer 3 Switch for Virtual Router Redundancy Protocol (VRRP) or VRRP Extended (VRRP-E). Refer to [Chapter 39, “VRRP and VRRP-E”](#).

Normally, a VRRP or VRRP-E backup includes route information for the virtual IP address (the backed-up interface) in RIP advertisements. As a result, other routers receive multiple paths for the backed-up interface and might sometimes unsuccessfully use the path to the backup rather than the path to the master.

You can prevent the backups from advertising route information for the backed-up interface by enabling suppression of the advertisements.

To suppress RIP advertisements for the backed-up interface, enter the following commands.

```
Brocade(config)#router rip
Brocade(config-rip-router)#use-vrrp-path
```

Syntax: [no] use-vrrp-path

The syntax is the same for VRRP and VRRP-E.

Configuring RIP route filters

You can configure RIP route filters to permit or deny learning or advertising of specific routes. Configure the filters globally, then apply them to individual interfaces. When you apply a RIP route filter to an interface, you specify whether the filter applies to learned routes (in) or advertised routes (out).

NOTE

A route is defined by the destination IP address and network mask.

NOTE

By default, routes that do not match a route filter are learned or advertised. To prevent a route from being learned or advertised, you must configure a filter to deny the route.

To configure RIP filters, enter commands such as the following.

```
Brocade(config-rip-router)#filter 1 permit 192.53.4.1 255.255.255.0
Brocade(config-rip-router)#filter 2 permit 192.53.5.1 255.255.255.0
Brocade(config-rip-router)#filter 3 permit 192.53.6.1 255.255.255.0
Brocade(config-rip-router)#filter 4 deny 192.53.7.1 255.255.255.0
```

These commands explicitly permit RIP routes to three networks, and deny the route to one network.

Because the default action is permit, all other routes (routes not explicitly permitted or denied by the filters) can be learned or advertised.

Syntax: `filter <filter-num> permit | deny <source-ip-address> | any <source-mask> | any [log]`

Applying a RIP route filter to an interface

Once you define RIP route filters, you must assign them to individual interfaces. The filters do not take effect until you apply them to interfaces. When you apply a RIP route filter, you also specify whether the filter applies to learned routes or advertised routes:

- Out filters apply to routes the Layer 3 Switch advertises to its neighbor on the interface.
- In filters apply to routes the Layer 3 Switch learns from its neighbor on the interface.

To apply RIP route filters to an interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/2
Brocade(config-if-0/1/2)#ip rip filter-group in 2 3 4
```

These commands apply RIP route filters 2, 3, and 4 to all routes learned from the RIP neighbor on port 1/2.

Syntax: `[no] ip rip filter-group in | out <filter-list>`

Displaying RIP filters

To display the RIP filters configured on the router, enter the **show ip rip** command at any CLI level.

```
Brocade#show ip rip
```

```

      RIP Route Filter Table
  Index  Action  Route IP Address  Subnet Mask
  1      deny   any               any

      RIP Neighbor Filter Table
  Index  Action  Neighbor IP Address
  1      permit any

```

Syntax: **show ip rip**

[Table 207](#) describes the information displayed by the **show ip rip** command.

TABLE 207 CLI display of RIP filter information

Field	Definition
Route filters	
The rows underneath “RIP Route Filter Table” list the RIP route filters. If no RIP route filters are configured on the device, the following message is displayed: “No Filters are configured in RIP Route Filter Table”.	
Index	The filter number. You assign this number when you configure the filter.
Action	<p>The action the router takes if a RIP route packet matches the IP address and subnet mask of the filter. The action can be one of the following:</p> <ul style="list-style-type: none"> deny – RIP route packets that match the address and network mask information in the filter are dropped. If applied to an interface outbound filter group, the filter prevents the router from advertising the route on that interface. If applied to an interface inbound filter group, the filter prevents the router from adding the route to its IP route table. permit – RIP route packets that match the address and network mask information are accepted. If applied to an interface outbound filter group, the filter allows the router to advertise the route on that interface. If applied to an interface inbound filter group, the filter allows the router to add the route to its IP route table.
Route IP Address	The IP address of the route destination network or host.
Subnet Mask	The network mask for the IP address.
Neighbor filters	
The rows underneath “RIP Neighbor Filter Table” list the RIP neighbor filters. If no RIP neighbor filters are configured on the device, the following message is displayed: “No Filters are configured in RIP Neighbor Filter Table”.	
Index	The filter number. You assign this number when you configure the filter.
Action	<p>The action the router takes for RIP route packets to or from the specified neighbor:</p> <ul style="list-style-type: none"> deny – If the filter is applied to an interface outbound filter group, the filter prevents the router from advertising RIP routes to the specified neighbor on that interface. If the filter is applied to an interface inbound filter group, the filter prevents the router from receiving RIP updates from the specified neighbor. permit – If the filter is applied to an interface outbound filter group, the filter allows the router to advertise RIP routes to the specified neighbor on that interface. If the filter is applied to an interface inbound filter group, the filter allows the router to receive RIP updates from the specified neighbor.
Neighbor IP Address	The IP address of the RIP neighbor.

Displaying CPU utilization statistics

You can display CPU utilization statistics for RIP and other IP protocols. To display CPU utilization statistics for RIP for the previous five-second, one-minute, five-minute, fifteen-minute, and runtime intervals, enter the **show process cpu** command at any level of the CLI.

```
Brocade#show process cpu
```

Process Name	5Sec(%)	1Min(%)	5Min(%)	15Min(%)	Runtime(ms)
ARP	0.01	0.03	0.09	0.22	9
BGP	0.04	0.06	0.08	0.14	13
GVRP	0.00	0.00	0.00	0.00	0
ICMP	0.00	0.00	0.00	0.00	0
IP	0.00	0.00	0.00	0.00	0
OSPF	0.00	0.00	0.00	0.00	0
RIP	0.04	0.07	0.08	0.09	7
STP	0.00	0.00	0.00	0.00	0
VRRP	0.00	0.00	0.00	0.00	0

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running, as shown in the following example.

```
Brocade#show process cpu
```

The system has only been up for 6 seconds.

Process Name	5Sec(%)	1Min(%)	5Min(%)	15Min(%)	Runtime(ms)
ARP	0.01	0.00	0.00	0.00	0
BGP	0.00	0.00	0.00	0.00	0
GVRP	0.00	0.00	0.00	0.00	0
ICMP	0.01	0.00	0.00	0.00	1
IP	0.00	0.00	0.00	0.00	0
OSPF	0.00	0.00	0.00	0.00	0
RIP	0.00	0.00	0.00	0.00	0
STP	0.00	0.00	0.00	0.00	0
VRRP	0.00	0.00	0.00	0.00	0

To display utilization statistics for a specific number of seconds, enter a command such as the following.

```
Brocade#show process cpu 2
```

Statistics for last 1 sec and 80 ms

Process Name	Sec(%)	Time(ms)
ARP	0.00	0
BGP	0.00	0
GVRP	0.00	0
ICMP	0.01	1
IP	0.00	0
OSPF	0.00	0
RIP	0.00	0
STP	0.01	0
VRRP	0.00	0

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is for the previous 1 second and 80 milliseconds.

Syntax: `show process cpu [<num>]`

The `<num>` parameter specifies the number of seconds and can be from 1 through 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous five-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying CPU utilization statistics

RIP (IPv6)

Table 208 lists the individual Brocade FastIron switches and the Routing Information Protocol (RIP) for IPv6 features they support. These features are supported with premium IPv6 devices running the full Layer 3 software image

TABLE 208 Supported RIPng features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
RIPng	Yes	No	Yes	Yes	No
Up to 10K RIPng routes	Yes	No	No	No	No
RIPng timers	Yes	No	No	No	No
Route learning and advertising	Yes	No	No	No	No
Route redistribution into RIPng	Yes	No	No	No	No
Route loop prevention: <ul style="list-style-type: none"> • Poison reverse • Split horizon 	Yes	No	No	No	No
Clearing RIPng routes	Yes	No	No	No	No

RIPng overview

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a distance vector (a number representing a distance) to measure the cost of a given route. RIP uses a hop count as its cost or metric.

IPv6 RIP, known as *Routing Information Protocol Next Generation* or RIPng functions similarly to IPv4 RIP Version 2. RIPng supports IPv6 addresses and prefixes and introduces some new commands that are specific to RIPng. This chapter describes the commands that are specific to RIPng. This section does not describe commands that apply to both IPv4 RIP and RIPng.

RIPng maintains a *Routing Information Base (RIB)*, which is a local route table. The local RIB contains the lowest-cost IPv6 routes learned from other RIP routers. In turn, RIPng attempts to add routes from its local RIB into the main IPv6 route table.

NOTE

Brocade IPv6 devices support up to 10,000 RIPng routes.

This section describes the following:

- How to configure RIPng
- How to clear RIPng information from the RIPng route table
- How to display RIPng information and statistics

Summary of configuration tasks

To configure RIPng, you must enable RIPng globally on the Brocade device and on individual router interfaces. The following configuration tasks are optional:

- Change the default settings of RIPng timers
- Configure how the Brocade device learns and advertises routes
- Configure which routes are redistributed into RIPng from other sources
- Configure how the Brocade device distributes routes through RIPng
- Configure poison reverse parameters

RIPng configuration

Before configuring the Brocade device to run RIPng, you must do the following:

- Enable the forwarding of IPv6 traffic on the Brocade device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface on which you plan to enable RIPng. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

For more information about performing these configuration tasks, refer to the chapter [“IPv6 Configuration on FastIron X Series, FCX, and ICX Series Switches”](#) on page 353.

Enabling RIPng

By default, RIPng is disabled. To enable RIPng, you must enable it globally on the Brocade device and also on individual router interfaces.

NOTE

You are required to configure a router ID when running only IPv6 routing protocols.

NOTE

Enabling RIPng globally on the Brocade device does not enable it on individual router interfaces.

To enable RIPng globally, enter the following command.

```
Brocade(config)#ipv6 router rip
Brocade(config-ripng-router)#
```

After you enter this command, the Brocade device enters the RIPng configuration level, where you can access several commands that allow you to configure RIPng.

Syntax: `[no] ipv6 router rip`

To disable RIPng globally, use the **no** form of this command.

After enabling RIPng globally, you must enable it on individual router interfaces. You can enable it on physical as well as virtual routing interfaces. For example, to enable RIPng on Ethernet interface 3/1, enter the following commands.

```
Brocade(config)# interface ethernet 3/1
Brocade(config-if-e100-3/1)# ipv6 rip enable
```

Syntax: `[no] ipv6 rip enable`

To disable RIPng on an individual router interface, use the **no** form of this command.

RIPng timers

[Table 209](#) describes the RIPng timers and provides their defaults.

TABLE 209 RIPng timers

Timer	Description	Default
Update	Amount of time (in seconds) between RIPng routing updates.	30 seconds.
Timeout	Amount of time (in seconds) after which a route is considered unreachable.	180 seconds.
Hold-down	Amount of time (in seconds) during which information about other paths is ignored.	180 seconds.
Garbage-collection	Amount of time (in seconds) after which a route is removed from the routing table.	120 seconds.

You can adjust these timers for RIPng. Before doing so, keep the following caveats in mind:

- If you adjust these RIPng timers, Brocade strongly recommends setting the same timer values for all routers and access servers in the network.
- Setting the update timer to a shorter interval can cause the routers to spend excessive time updating the IPv6 route table.
- Brocade recommends setting the timeout timer value to at least three times the value of the update timer.
- Brocade recommends a shorter hold-down timer interval, because a longer interval can cause delays in RIPng convergence.

Updating RIPng timers

The following example sets updates to be broadcast every 45 seconds. If a route is not heard from in 135 seconds, the route is declared unusable. Further information is suppressed for an additional 10 seconds. Assuming no updates, the route is flushed from the routing table 20 seconds after the end of the hold-down period.

```
Brocade(config)# ipv6 router rip
Brocade(config-ripng-router)# timers 45 135 10 20
```

Syntax: `[no] timers <update-timer> <timeout-timer> <hold-down-timer> <garbage-collection-timer>`

Possible values for the timers are as follows

- Update timer: 3 through 65535 seconds
- Timeout timer: 9 through 65535 seconds
- Hold-down timer: 9 through 65535 seconds
- Garbage-collection timer: 9 through 65535 seconds

NOTE

You must enter a value for each timer, even if you want to retain the current setting of a particular timer.

To return to the default values of the RIPng timers, use the **no** form of this command.

Route learning and advertising parameters

You can configure the following learning and advertising parameters:

- Learning and advertising of RIPng default routes
- Advertising of IPv6 address summaries
- Metric of routes learned and advertised on a router interface

By default, the Brocade device does not learn IPv6 default routes (::/0). You can originate default routes into RIPng, which causes individual router interfaces to include the default routes in their updates. When configuring the origination of the default routes, you can also do the following:

- Suppress all other routes from the updates
- Include all other routes in the updates

Configuring default route learning and advertising

To originate default routes in RIPng and suppress all other routes in updates sent from Ethernet interface 3/1, enter the following commands.

```
Brocade(config)# interface ethernet 3/1
Brocade(config-if-e100-3/1)# ipv6 rip default-information only
```

To originate IPv6 default routes and include all other routes in updates sent from Ethernet interface 3/1, enter the following commands.

```
Brocade(config)# interface ethernet 3/1
Brocade(config-if-e100-3/1)# ipv6 rip default-information originate
```

Syntax: [no] **ipv6 rip default-information only | originate**

The **only** keyword originates the default routes and suppresses all other routes from the updates.

The **originate** keyword originates the default routes and includes all other routes in the updates.

To remove the explicit default routes from RIPng and suppress advertisement of these routes, use the **no** form of this command.

Advertising IPv6 address summaries

You can configure RIPng to advertise a summary of IPv6 addresses from a router interface and to specify an IPv6 prefix that summarizes the routes.

If a route prefix length matches the value specified in the **ipv6 rip summary-address** command, RIPng advertises the prefix specified in the **ipv6 rip summary-address** command instead of the original route.

For example, to advertise the summarized prefix 2001:DB8::/36 instead of the IPv6 address 2001:469e:0:adff:8935:e838:78:e0ff with a prefix length of 64 bits from Ethernet interface 3/1, enter the following commands.

```
Brocade(config)# interface ethernet 3/1
Brocade(config-if-e100-3/1)# ipv6 address 2001:db8:0:adff:8935:e838:78:
e0ff /64
Brocade(config-if-e100-3/1)# ipv6 rip summary-address 2001:db8::/36
```

Syntax: [no] **ipv6 rip summary-address** <ipv6-prefix>/<prefix-length>

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

To stop the advertising of the summarized IPv6 prefix, use the **no** form of this command.

Changing the metric of routes learned and advertised on an interface

A router interface increases the metric of an incoming RIPng route it learns by an offset (the default is 1). The Brocade device then places the route in the route table. When the Brocade device sends an update, it advertises the route with the metric plus the default offset of zero in an outgoing update message.

You can change the metric offset an individual interface adds to a route learned by the interface or advertised by the interface. For example, to change the metric offset for incoming routes learned by Ethernet interface 3/1 to 1 and the metric offset for outgoing routes advertised by the interface to 3, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e100-3/1)#ipv6 rip metric-offset 1
Brocade(config-if-e100-3/1)#ipv6 rip metric-offset out 3
```

In this example, if Ethernet interface 3/1 learns about an incoming route, it will increase the incoming metric by 2 (the default offset of 1 and the additional offset of 1 as specified in this example). If Ethernet interface 3/1 advertises an outgoing route, it will increase the metric by 3 as specified in this example.

Syntax: [no] **ipv6 rip metric-offset** <1 - 16>

Syntax: [no] **ipv6 rip metric-offset out** <1 - 15>

To return the metric offset to its default value, use the **no** form of this command.

Redistributing routes into RIPng

You can configure the Brocade device to redistribute routes from the following sources into RIPng:

- IPv6 static routes
- Directly connected IPv6 networks
- OSPF V3

When you redistribute a route from IPv6 or OSPF V3 into RIPng, the Brocade device can use RIPng to advertise the route to its RIPng neighbors.

When configuring the Brocade device to redistribute routes, you can optionally specify a metric for the redistributed routes. If you do not explicitly configure a metric, the default metric value of 1 is used.

For example, to redistribute OSPF V3 routes into RIPng, enter the following commands.

```
Brocade(config)# ipv6 router rip
Brocade(config-ripng-router)# redistribute ospf
```

Syntax: `redistribute bgp | connected | isis | ospf | static [metric <number>]`

For the metric, specify a numerical value that is consistent with RIPng.

Controlling distribution of routes through RIPng

You can create a prefix list and then apply it to RIPng routing updates that are received or sent on a router interface. Performing this task allows you to control the distribution of routes through RIPng.

For example, to permit the inclusion of routes with the prefix 2001:DB8::/16 in RIPng routing updates sent from Ethernet interface 3/1, enter the following commands.

```
Brocade(config)# ipv6 prefix-list routesfor2001 permit 2001:db8::/16
Brocade(config)# ipv6 router rip
Brocade(config-ripng-router)# distribute-list prefix-list routesfor2001 out
ethernet 3/1
```

To deny prefix lengths greater than 64 bits in routes that have the prefix 3EE0:A99::/64 and allow all other routes received on tunnel interface 3/1, enter the following commands.

```
Brocade(config)# ipv6 prefix-list 3ee0routes deny 3ee0:a99::/64 le 128
Brocade(config)# ipv6 prefix-list 3ee0routes permit ::/0 ge 1 le 128
Brocade(config)# ipv6 router rip
Brocade(config-ripng-router)# distribute-list prefix-list 3ee0routes in
tunnel 1
```

Syntax: `[no] distribute-list prefix-list <name> in | out <interface> <port>`

The `<name>` parameter indicates the name of the prefix list generated using the `ipv6 prefix-list` command.

The `in` keyword indicates that the prefix list is applied to incoming routing updates on the specified interface.

The `out` keyword indicates that the prefix list is applied to outgoing routing updates on the specified interface.

For the *<interface>* parameter, you can specify the **ethernet**, **loopback**, **ve**, or **tunnel** keywords. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number.

To remove the prefix list, use the **no** form of this command.

Configuring poison reverse parameters

By default, poison reverse is disabled on a RIPng router. If poison reverse is enabled, RIPng advertises routes it learns from a particular interface over that same interface with a metric of 16, which means that the route is unreachable.

If poison reverse is enabled on the RIPng router, it takes precedence over split horizon (if it is also enabled).

To enable poison reverse on the RIPng router, enter the following commands.

```
Brocade(config)# ipv6 router rip
Brocade(config-ripng-router)# poison-reverse
```

Syntax: [no] **poison-reverse**

To disable poison reverse, use the **no** form of this command.

By default, if a RIPng interface goes down, the Brocade device does not send a triggered update for the interface IPv6 networks.

To better handle this situation, you can configure a RIPng router to send a triggered update containing the local routes of the disabled interface with an unreachable metric of 16 to the other RIPng routers in the routing domain. You can enable the sending of a triggered update by entering the following commands.

```
Brocade(config)# ipv6 router rip
Brocade(config-ripng-router)# poison-local-routes
```

Syntax: [no] **poison-local-routes**

To disable the sending of a triggered update, use the **no** form of this command.

Clearing RIPng routes from the IPv6 route table

To clear all RIPng routes from the RIPng route table and the IPv6 main route table and reset the routes, enter the following command at the Privileged EXEC level or any of the CONFIG levels of the CLI.

```
Brocade# clear ipv6 rip routes
```

Syntax: **clear ipv6 rip routes**

Displaying the RIPng configuration

To display RIPng configuration information, enter the **show ipv6 rip** command at any CLI level.

```
Brocade# show ipv6 rip
IPv6 rip enabled, port 521
  Administrative distance is 120
  Updates every 30 seconds, expire after 180
  Holddown lasts 180 seconds, garbage collect after 120
  Split horizon is on; poison reverse is off
  Default routes are not generated
  Periodic updates 0, trigger updates 0
  Distribute List, Inbound : Not set
  Distribute List, Outbound : Not set
  Redistribute: CONNECTED
```

Syntax: **show ipv6 rip**

[Table 210](#) describes the information displayed by the **show ipv6 rip** command.

TABLE 210 RIPng configuration fields

Field	Description
IPv6 RIP status/port	The status of RIPng on the Brocade device. Possible status is “enabled” or “disabled.” The UDP port number over which RIPng is enabled.
Administrative distance	The setting of the administrative distance for RIPng.
Updates/expiration	The settings of the RIPng update and timeout timers.
Holddown/garbage collection	The settings of the RIPng hold-down and garbage-collection timers.
Split horizon/poison reverse	The status of the RIPng split horizon and poison reverse features. Possible status is “on” or “off.”
Default routes	The status of RIPng default routes.
Periodic updates/trigger updates	The number of periodic updates and triggered updates sent by the RIPng router.
Distribution lists	The inbound and outbound distribution lists applied to RIPng.
Redistribution	The types of IPv6 routes redistributed into RIPng. The types can include the following: <ul style="list-style-type: none"> • STATIC – IPv6 static routes are redistributed into RIPng. • CONNECTED – Directly connected IPv6 networks are redistributed into RIPng. • OSPF – OSPF V3 routes are redistributed into RIPng.

Displaying RIPng routing table

To display the RIPng routing table, enter the **show ipv6 rip route** command at any CLI level.

```
Brocade# show ipv6 rip route
IPv6 RIP Routing Table - 4 entries:
 2001::/64, from fe80::212:f2ff:fe87:9a40, ve 177      (314)
     RIP, metric 2, tag 0, timers: aging 11
 2001:11:11::/64, from ::, null      (0)
     CONNECTED, metric 1, tag 0, timers: none
 2001:aa:2001:102:1:1::/96, from ::, ve 102      (1)
     LOCAL, metric 1, tag 0, timers: none
 2001:aa:2061:31:1:1::/96, from fe80::2e0:52ff:fe88:8000, ve 100      (58)
     RIP, metric 2, tag 0, timers: aging 43
```

Syntax: **show ipv6 rip route** [*<ipv6-prefix>/<prefix-length>* | *<ipv6-address>*]

The *<ipv6-prefix>/<prefix-length>* parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the *<ipv6-prefix>* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *<prefix-length>* parameter as a decimal value. A slash mark (/) must follow the *<ipv6-prefix>* parameter and precede the *<prefix-length>* parameter.

The *<ipv6-address>* parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

[Table 211](#) describes the information displayed by the **show ipv6 rip route** command.

TABLE 211 RIPng routing table fields

Field	Description
IPv6 RIPng Routing Table entries	The total number of entries in the RIPng routing table.
<i><ipv6-prefix>/<prefix-length></i> OR <i><ipv6-address></i>	The IPv6 prefix and prefix length. The IPv6 address.
Next-hop router	The next-hop router for this Brocade device. If :: appears, the route is originated locally.
Interface	The interface name. If "null" appears, the interface is originated locally.
Serial number	The number identifying the order in which the route was added to the RIPng route table.
Source of route	The source of the route information. The source can be one of the following: <ul style="list-style-type: none"> • LOCAL – Routes configured on local interfaces taking part in RIPng. • RIP – Routes learned by RIPng. • CONNECTED – IPv6 routes redistributed from directly connected networks. • STATIC – IPv6 static routes are redistributed into RIPng. • OSPF – OSPF V3 routes are redistributed into RIPng.
Metric <i><number></i>	The cost of the route. The <i><number></i> parameter indicates the number of hops to the destination.
Tag <i><number></i>	The tag value of the route.
Timers:	Indicates if the hold-down timer or the garbage-collection timer is set.

Displaying RIPvng routing table

OSPF version 2 (IPv4)

[Table 212](#) lists the individual Brocade FastIron switches and the Open Shortest Path First (OSPF) Version 2 (IPv4) features they support. These features are supported in the edge Layer 3 and full Layer 3 software images only.

TABLE 212 Supported OSPF V2 features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
OSPF V2	Yes	Yes	Yes	Yes	Yes
OSPF point-to-point links	Yes	Yes	Yes	Yes	Yes
RFC 1583 and RFC 2178 compliant	Yes	Yes	Yes	Yes	Yes
Support for OSPF RFC 2328 Appendix E	Yes	Yes	Yes	Yes	Yes
Dynamic OSPF activation and configuration	Yes	Yes	Yes	Yes	Yes
Dynamic OSPFmemory	Yes	Yes	Yes	Yes	Yes
OSPF graceful restart	Yes (FSX 800 and FSX 1600 only)	No	Yes (FCX stack only)	Yes (ICX 6610 stack only)	Yes (ICX 6450 stack only)
Assigning OSPF V2 areas	Yes	Yes	Yes	Yes	Yes
Assigning interfaces to an area	Yes	Yes	Yes	Yes	Yes
Timer for OSPF authentication changes	Yes	Yes	Yes	Yes	Yes
Block flooding of outbound LSAs on specific interfaces	Yes	Yes	Yes	Yes	Yes
OSPF non-broadcast interface	Yes	Yes	Yes	Yes	Yes
Virtual links	Yes	Yes	Yes	Yes	Yes
Changing the reference bandwidth for the cost on OSPF interfaces	Yes	Yes	Yes	Yes	Yes
Route redistribution filters	Yes	Yes	Yes	Yes	Yes
Prevent specific OSPF routes from being installed in the IP route table	Yes	Yes	Yes	Yes	Yes
Load sharing	Yes	Yes	Yes	Yes	Yes
Configuring default route origination	Yes	Yes	Yes	Yes	Yes
SPF timers	Yes	Yes	Yes	Yes	Yes
Modifying redistribution metric type	Yes	Yes	Yes	Yes	Yes
Modifying administrative distance	Yes	Yes	Yes	Yes	Yes
OSPF group LSA pacing	Yes	Yes	Yes	Yes	Yes

TABLE 212 Supported OSPF V2 features (Continued)

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
OSPF traps	Yes	Yes	Yes	Yes	Yes
Exit overflow interval	Yes	Yes	Yes	Yes	Yes
Syslog messages	Yes	Yes	Yes	Yes	Yes
Clearing OSPF information	Yes	Yes	Yes	Yes	Yes

This chapter describes how to configure OSPF Version 2 on Brocade Layer 3 Switches using the CLI. OSPF Version 2 is supported on devices running IPv4.

NOTE

The terms *Layer 3 Switch* and *router* are used interchangeably in this chapter and mean the same thing.

OSPF overview

Open Shortest Path First (OSPF) is a link-state routing protocol. The protocol uses link-state advertisements (LSAs) to update neighboring routers regarding its interfaces and information on those interfaces. The router floods these LSAs to all neighboring routers to update them regarding the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

Brocade Layer 3 Switches support the following types of LSAs, which are described in RFC 1583:

- Router link
- Network link
- Summary link
- Autonomous system (AS) summary link
- AS external link
- Not-So-Stubby Area (NSSA) external link
- Grace LSAs

OSPF is built upon a hierarchy of network components. The highest level of the hierarchy is the Autonomous System (AS). An autonomous system is defined as a number of networks, all of which share the same routing and administration characteristics.

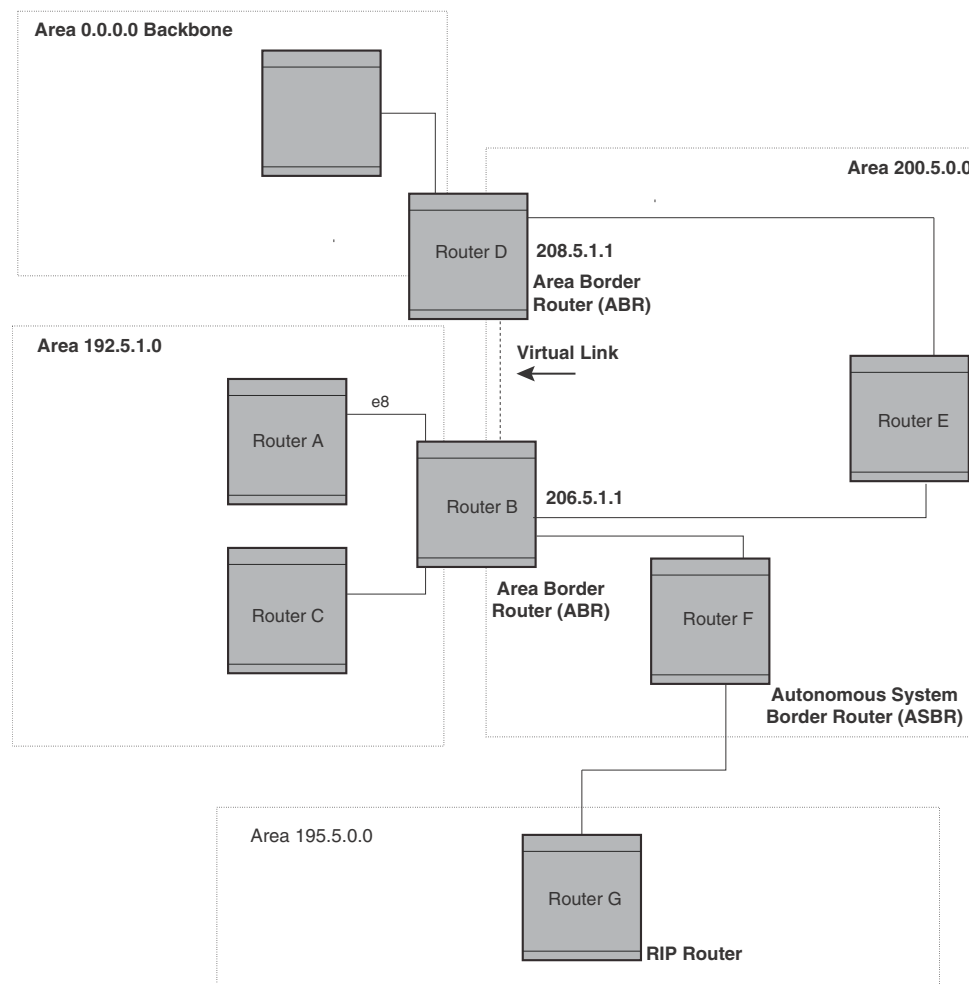
An AS can be divided into multiple areas as shown in [Figure 158](#) on page 1227. Each area represents a collection of contiguous networks and hosts. Areas limit the area to which link-state advertisements are broadcast, thereby limiting the amount of flooding that occurs within the network. An area is represented in OSPF by either an IP address or a number.

You can further limit the broadcast area of flooding by defining an area range. The area range allows you to assign an aggregate value to a range of IP addresses. This aggregate value becomes the address that is advertised instead all of the individual addresses it represents being advertised. You can assign up to 32 ranges in an OSPF area.

An OSPF router can be a member of multiple areas. Routers with membership in multiple areas are known as Area Border Routers (ABRs). Each ABR maintains a separate topological database for each area the router is in. Each topological database contains all of the LSA databases for each router within a given area. The routers within the same area have identical topological databases. The ABR is responsible for forwarding routing information or changes between its border areas.

An Autonomous System Boundary Router (ASBR) is a router that is running multiple protocols and serves as a gateway to routers outside an area and those operating with different protocols. The ASBR is able to import and translate different protocol routes into OSPF through a process known as *redistribution*. For more details on redistribution and configuration examples, refer to [“Enabling route redistribution”](#) on page 1258.

FIGURE 158 OSPF operating in a network



OSPF point-to-point links

One important OSPF process is **Adjacency**. Adjacency occurs when a relationship is formed between neighboring routers for the purpose of exchanging routing information. Adjacent OSPF neighbor routers go beyond the simple Hello packet exchange; they exchange database information. In order to minimize the amount of information exchanged on a particular segment, one of the first steps in creating adjacency is to assign a Designated Router (DR) and a Backup Designated Router (BDR). The Designated Router ensures that there is a central point of contact, thereby improving convergence time within a multi-access segment.

In an OSPF point-to-point network, where a direct Layer 3 connection exists between a single pair of OSPF routers, there is no need for Designated and Backup Designated Routers, as is the case in OSPF multi-access networks. Without the need for Designated and Backup Designated routers, a point-to-point network establishes adjacency and converges faster. The neighboring routers become adjacent whenever they can communicate directly. In contrast, in broadcast and non-broadcast multi-access (NBMA) networks, the Designated Router and Backup Designated Router become adjacent to all other routers attached to the network.

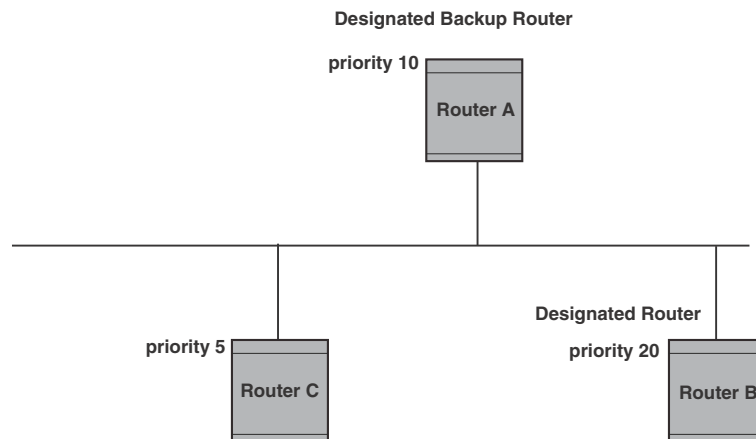
To configure an OSPF point-to-point link, refer to [“Configuring an OSPF point-to-point link”](#) on page 1269.

Designated routers in multi-access networks

In a network that has multiple routers attached, OSPF elects one router to serve as the designated router (DR) and another router on the segment to act as the backup designated router (BDR). This arrangement minimizes the amount of repetitive information that is forwarded on the network by forwarding all messages to the designated router and backup designated routers responsible for forwarding the updates throughout the network.

Designated router election in multi-access networks

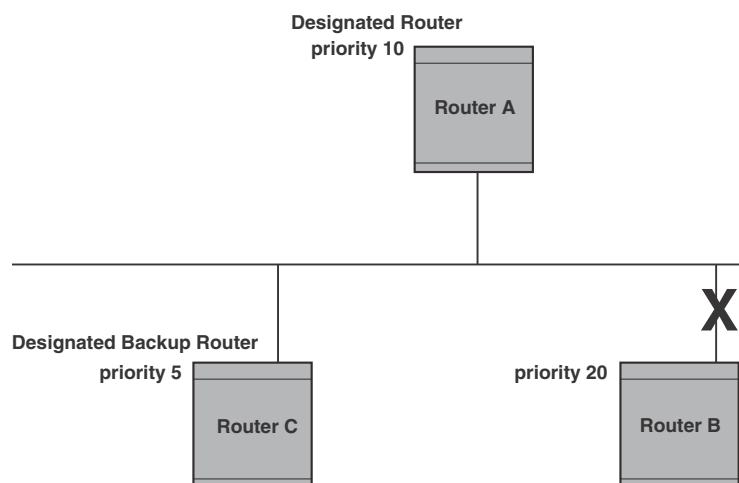
In a network with no designated router and no backup designated router, the neighboring router with the highest priority is elected as the DR, and the router with the next largest priority is elected as the BDR, as shown in [Figure 159](#)

FIGURE 159 Designated and backup router election

If the DR goes off-line, the BDR automatically becomes the DR. The router with the next highest priority becomes the new BDR. This process is shown in [Figure 160](#).

NOTE

Priority is a configurable option at the interface level. You can use this parameter to help bias one router as the DR.

FIGURE 160 Backup designated router becomes designated router

If two neighbors share the same priority, the router with the highest router ID is designated as the DR. The router with the next highest router ID is designated as the BDR.

NOTE

By default, the Brocade router ID is the IP address configured on the lowest numbered loopback interface. If the Layer 3 Switch does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, refer to [“Changing the router ID”](#) on page 976.

When multiple routers on the same network are declaring themselves as DRs, then both priority and router ID are used to select the designated router and backup designated routers.

When only one router on the network claims the DR role despite neighboring routers with higher priorities or router IDs, this router remains the DR. This is also true for BDRs.

The DR and BDR election process is performed when one of the following events occurs:

- An interface is in a waiting state and the wait time expires
- An interface is in a waiting state and a hello packet is received that addresses the BDR
- A change in the neighbor state occurs, such as:
 - A neighbor state transitions from 2 or higher
 - Communication to a neighbor is lost
 - A neighbor declares itself to be the DR or BDR for the first time

OSPF RFC 1583 and 2178 compliance

Brocade routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification. Brocade routers can also be configured to operate with the latest OSPF standard, RFC 2178.

NOTE

For details on how to configure the system to operate with the RFC 2178, refer to [“Modifying the OSPF standard compliance setting”](#) on page 1268.

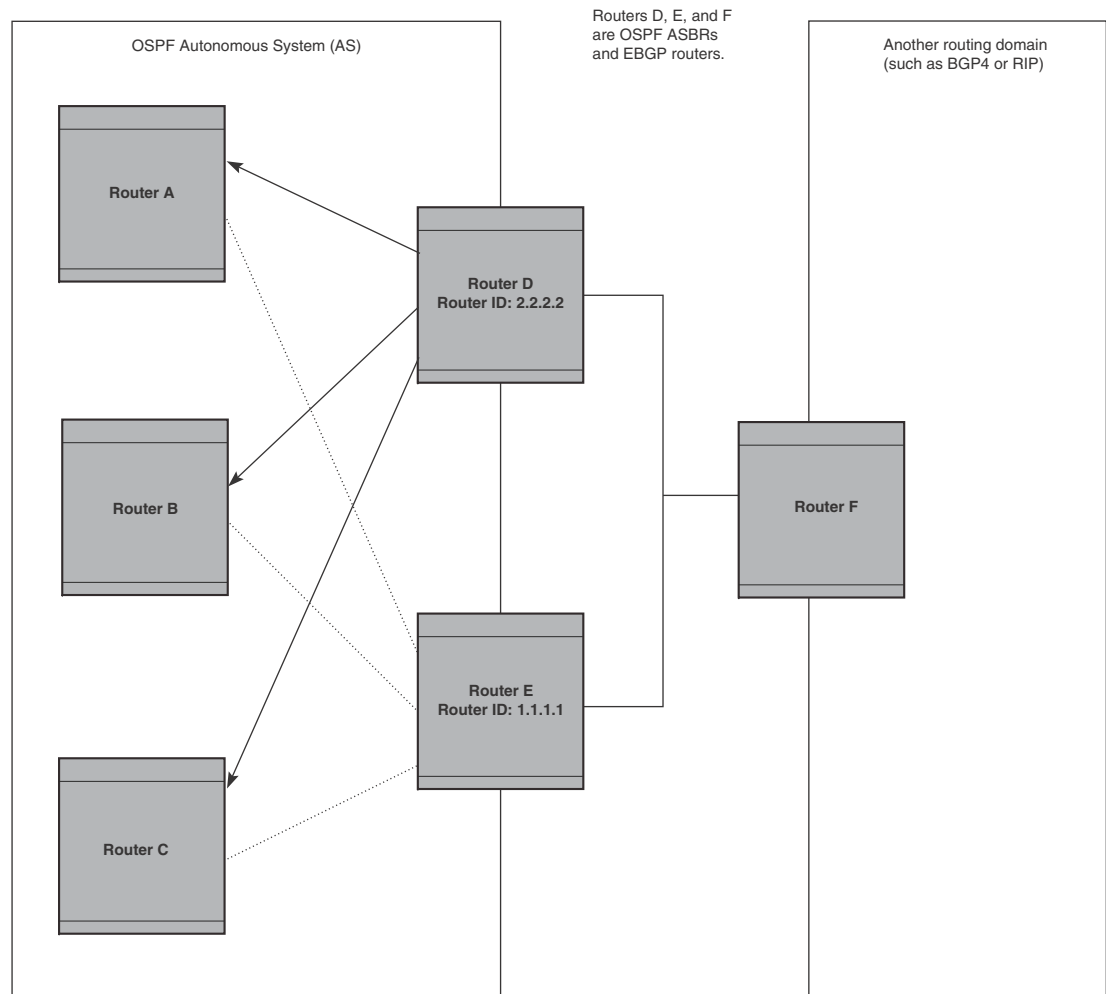
Reduction of equivalent AS External LSAs

An OSPF ASBR uses AS External link advertisements (AS External LSAs) to originate advertisements of a route to another routing domain, such as a BGP4 or RIP domain. The ASBR advertises the route to the external domain by flooding AS External LSAs to all the other OSPF routers (except those inside stub networks) within the local OSPF Autonomous System (AS).

In some cases, multiple ASBRs in an AS can originate equivalent LSAs. The LSAs are equivalent when they have the same cost, the same next hop, and the same destination. Brocade devices optimize OSPF by eliminating duplicate AS External LSAs in this case. The Layer 3 Switch with the lower router ID flushes the duplicate External LSAs from its database and thus does not flood the duplicate External LSAs into the OSPF AS. AS External LSA reduction therefore reduces the size of the Layer 3 Switch link state database.

This enhancement implements the portion of RFC 2328 that describes AS External LSA reduction. This enhancement is enabled by default, requires no configuration, and cannot be disabled.

[Figure 161](#) shows an example of the AS External LSA reduction feature. In this example, Brocade Layer 3 Switches D and E are OSPF ASBRs, and thus communicate route information between the OSPF AS, which contains Routers A, B, and C, and another routing domain, which contains Router F. The other routing domain is running another routing protocol, such as BGP4 or RIP. Routers D, E, and F, therefore, are each running both OSPF and either BGP4 or RIP.

FIGURE 161 AS External LSA reduction

Notice that both Router D and Router E have a route to the other routing domain through Router F. In earlier software releases, if Routers D and E have equal-cost routes to Router F, then both Router D and Router E flood AS External LSAs to Routers A, B, and C advertising the route to Router F. Since both routers are flooding equivalent routes, Routers A, B, and C receive multiple routes with the same cost to the same destination (Router F). For Routers A, B, and C, either route to Router F (through Router D or through Router E) is equally good.

OSPF eliminates the duplicate AS External LSAs. When two or more Brocade Layer 3 Switches configured as ASBRs have equal-cost routes to the same next-hop router in an external routing domain, the ASBR with the highest router ID floods the AS External LSAs for the external domain into the OSPF AS, while the other ASBRs flush the equivalent AS External LSAs from their databases. As a result, the overall volume of route advertisement traffic within the AS is reduced and the Layer 3 Switches that flush the duplicate AS External LSAs have more memory for other OSPF data. In [Figure 161](#), since Router D has a higher router ID than Router E, Router D floods the AS External LSAs for Router F to Routers A, B, and C. Router E flushes the equivalent AS External LSAs from its database.

Algorithm for AS External LSA reduction

Figure 161 shows an example in which the normal AS External LSA reduction feature is in effect. The behavior changes under the following conditions:

- There is one ASBR advertising (originating) a route to the external destination, but one of the following happens:
 - A second ASBR comes on-line
 - A second ASBR that is already on-line begins advertising an equivalent route to the same destination.

In either case above, the router with the higher router ID floods the AS External LSAs and the other router flushes its equivalent AS External LSAs. For example, if Router D is offline, Router E is the only source for a route to the external routing domain. When Router D comes on-line, it takes over flooding of the AS External LSAs to Router F, while Router E flushes its equivalent AS External LSAs to Router F.

- One of the ASBRs starts advertising a route that is no longer equivalent to the route the other ASBR is advertising. In this case, the ASBRs each flood AS External LSAs. Since the LSAs either no longer have the same cost or no longer have the same next-hop router, the LSAs are no longer equivalent, and the LSA reduction feature no longer applies.
- The ASBR with the higher router ID becomes unavailable or is reconfigured so that it is no longer an ASBR. In this case, the other ASBR floods the AS External LSAs. For example, if Router D goes off-line, then Router E starts flooding the AS with AS External LSAs for the route to Router F.

Support for OSPF RFC 2328 Appendix E

Brocade devices provide support for Appendix E in OSPF RFC 2328. Appendix E describes a method to ensure that an OSPF router (such as a Brocade Layer 3 Switch) generates unique link state IDs for type-5 (External) link state advertisements (LSAs) in cases where two networks have the same network address but different network masks.

NOTE

Support for Appendix E of RFC 2328 is enabled automatically and cannot be disabled. No user configuration is required.

Normally, an OSPF router uses the network address alone for the link state ID of the link state advertisement (LSA) for the network. For example, if the router needs to generate an LSA for network 10.1.2.3 255.0.0.0, the router generates ID 10.1.2.3 for the LSA.

However, suppose that an OSPF router needs to generate LSAs for all the following networks:

- 10.0.0.0 255.0.0.0
- 10.0.0.0 255.255.0.0
- 10.0.0.0 255.255.255.0

All three networks have the same network address, 10.0.0.0. Without support for RFC 2328 Appendix E, an OSPF router uses the same link state ID, 10.0.0.0, for the LSAs for all three networks. For example, if the router generates an LSA with ID 10.0.0.0 for network 10.0.0.0 255.0.0.0, this LSA conflicts with the LSA generated for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.255.255.0. The result is multiple LSAs that have the same ID but that contain different route information.

When Appendix E is supported, the router generates the link state ID for a network as follows.

1. Does an LSA with the network address as its ID already exist?
 - No – Use the network address as the ID.
 - Yes – Go to [step 2](#).
2. Compare the networks that have the same network address, to determine which network is more specific. The more specific network is the one that has more contiguous one bits in its network mask. For example, network 10.0.0.0 255.255.0.0 is more specific than network 10.0.0.0 255.0.0.0, because the first network has 16 ones bits (255.255.0.0) whereas the second network has only 8 ones bits (255.0.0.0):
 - For the less specific network, use the networks address as the ID.
 - For the more specific network, use the network broadcast address as the ID. The broadcast address is the network address, with all ones bits in the host portion of the address. For example, the broadcast address for network 10.0.0.0 255.255.0.0 is 10.0.255.255.

If this comparison results in a change to the ID of an LSA that has already been generated, the router generates a new LSA to replace the previous one. For example, if the router has already generated an LSA for network with ID 10.0.0.0 for network 10.0.0.0 255.255.255.0, the router must generate a new LSA for the network, if the router needs to generate an LSA for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.0.0.0.

Dynamic OSPF activation and configuration

OSPF is automatically activated when you enable it. The protocol does not require a software reload.

You can configure and save the following OSPF changes without resetting the system:

- All OSPF interface-related parameters (for example: area, hello timer, router dead time cost, priority, re-transmission time, transit delay)
- All area parameters
- All area range parameters
- All virtual-link parameters
- All global parameters
- Creation and deletion of an area, interface or virtual link

In addition, you can make the following changes without a system reset by first disabling and then re-enabling OSPF operation:

- Changes to address ranges
- Changes to global values for redistribution
- Addition of new virtual links

You also can change the amount of memory allocated to various types of LSA entries. However, these changes require a system reset or reboot.

Dynamic OSPF memory

FastIron devices dynamically allocate memory for Link State Advertisements (LSAs) and other OSPF data structures. This eliminates overflow conditions and does not require a reload to change OSPF memory allocation. So long as the Layer 3 Switch has free (unallocated) dynamic memory, OSPF can use the memory.

To display the current allocations of dynamic memory, use the **show memory** command.

OSPF graceful restart

OSPF graceful restart is a high-availability routing feature that minimizes disruption in traffic forwarding, diminishes route flapping, and provides continuous service during a system restart, including restart events that occur during a switchover, failover, or hitless OS upgrade. During such events, routes remain available between devices.

When OSPF graceful restart is enabled, a restarting router sends special LSAs, called grace LSAs, to its neighbors either before a planned OSPF restart or immediately after an unplanned restart. The grace LSAs specify a grace period for neighbors of the restarting router to continue using the existing routes to and through the router after a restart. When the restarting router comes back up, it continues to use its existing OSPF routes as if nothing happened. In the background, the router relearns its neighbors prior to the restart, recalculates its OSPF routes, and replaces existing routes with new routes as necessary. Once the grace period has passed, adjacent routers resume normal operation.

OSPF graceful restart is enabled globally by default. In this configuration, all OSPF neighbors are subject to the graceful restart capability. Neighbor routers must support the helper mode of OSPF graceful restart, which is enabled by default on all FastIron Layer 3 switches.

NOTE

OSPF graceful restart is supported in FSX 800 and FSX 1600 devices with dual management modules. If the device will function as a restart helper device only, a secondary management module is not required.

NOTE

If a FastIron device is configured for OSPF graceful restart and is intended to be used in switchover or hitless upgrade, the OSPF dead-interval should be changed to 60 seconds on OSPF interfaces to ensure that the graceful restart process succeeds without a timeout. Instructions for changing the OSPF dead-interval are provided in [“Modifying interface defaults”](#) on page 1242.

The Brocade implementation of OSPF graceful restart supports RFC 3623: Graceful OSPF Restart.

For details on how to configure OSPF graceful restart, refer to [“Configuring OSPF graceful restart”](#) on page 1269.

Configuring OSPF

Perform the following steps to begin using OSPF on the router.

1. [“Enabling OSPF on the router”](#) on page 1236
2. [“Assigning OSPF areas”](#) on page 1237
3. [“Assigning an area range \(optional\)”](#) on page 1241
4. [“Assigning interfaces to an area”](#) on page 1242.
5. [“Defining redistribution filters”](#) on page 1252
6. [“Enabling route redistribution”](#) on page 1258.
7. [“Modifying the OSPF standard compliance setting”](#) on page 1268

NOTE

OSPF is automatically enabled without a system reset.

OSPF configuration rules

- Brocade FastIron devices support a maximum of 676 OSPF interfaces.
- If a router is to operate as an ASBR, you must enable the ASBR capability at the system level.
- Redistribution must be enabled on routers configured to operate as ASBRs.
- All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

OSPF parameters

You can modify or set the following global and interface OSPF parameters.

Global parameters:

- Modify the OSPF standard compliance setting
- Assign OSPF areas
- Define an area range
- Define the area virtual link
- Set global default metric for OSPF
- Change the reference bandwidth for the default cost of OSPF interfaces
- Disable or re-enable load sharing
- Enable or disable default-information-originate
- Modify Shortest Path First (SPF) timers
- Define external route summarization
- Modify the redistribution metric type
- Define deny redistribution
- Define permit redistribution

- Enable redistribution
- Change the LSA pacing interval
- Modify OSPF Traps generated
- Modify database overflow interval

Interface parameters:

- Assign interfaces to an area
- Define the authentication key for the interface
- Change the authentication-change interval
- Modify the cost for a link
- Modify the dead interval
- Modify MD5 authentication key parameters
- Modify the priority of the interface
- Modify the retransmit interval for the interface
- Modify the transit delay of the interface

NOTE

When using the CLI, you set global level parameters at the OSPF CONFIG level of the CLI. To reach that level, enter **router ospf...** at the global CONFIG level. Interface parameters for OSPF are set at the interface CONFIG level using the CLI command, **ip ospf...**

When using the Web Management Interface, you set OSPF global parameters using the OSPF configuration panel. All other parameters are accessed through links accessed from the OSPF configuration sheet.

Enabling OSPF on the router

When you enable OSPF on the router, the protocol is automatically activated. To enable OSPF on the router, enter the following CLI command.

```
Brocade(config)#router ospf
```

This command launches you into the OSPF router level where you can assign areas and modify OSPF global parameters.

Syntax: **router ospf**

Note regarding disabling OSPF

If you disable OSPF, the Layer 3 Switch removes all the configuration information for the disabled protocol from the running-config. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

NOTE

If you do not want to delete the OSPF configuration information, use the CLI command **clear ip ospf all** instead of **no router ospf**. Refer to [“Resetting OSPF”](#) on page 1237.

When you enter the **no router ospf** command, the CLI displays a warning message such as the following.

```
Brocade(config-ospf-router)#no router ospf
router ospf mode now disabled. All ospf config data will be lost when writing to
flash!
```

The Web Management Interface does not display a warning message.

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (for example, **router ospf**), or by selecting the Web management option to enable the protocol. If you have already saved the configuration to the startup-config file and reloaded the software, the information is gone.

If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup-config file containing the protocol configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

Resetting OSPF

The **clear ip ospf all** command globally resets (disables then re-enables) OSPF without deleting the OSPF configuration information. This command is equivalent to entering the commands **no router ospf** followed by **router ospf**. Whereas the **no router ospf** command disables OSPF and removes all the configuration information for the disabled protocol from the running-config, the **router ospf** command re-enables OSPF and restores the OSPF configuration information.

The **clear ip ospf all** command is useful if you are testing an OSPF configuration and are likely to disable and re-enable the protocol. This way, you do not have to save the configuration after disabling the protocol, and you do not have to restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

To reset OSPF without deleting the OSPF configuration, enter the following command at the Global CONFIG level or at the Router OSPF level of the CLI.

```
Brocade#clear ip ospf all
```

Syntax: **clear ip ospf all**

Assigning OSPF areas

Once OSPF is enabled on the system, you can assign areas. Assign an IP address or number as the area ID for each area. The area ID is representative of all IP addresses (subnets) on a router port. Each port on a router can support one area.

An area can be normal, a stub, or a Not-So-Stubby Area (NSSA):

- Normal – OSPF routers within a normal area can send and receive External Link State Advertisements (LSAs).
- Stub – OSPF routers within a stub area cannot send or receive External LSAs. In addition, OSPF routers in a stub area must use a default route to the area Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.

- NSSA – The ASBR of an NSSA can import external route information into the area:
 - ASBRs redistribute (import) external routes into the NSSA as type 7 LSAs. Type-7 External LSAs are a special type of LSA generated only by ASBRs within an NSSA, and are flooded to all the routers within only that NSSA.
 - ABRs translate type 7 LSAs into type 5 External LSAs, which can then be flooded throughout the AS. You can configure address ranges on the ABR of an NSSA so that the ABR converts multiple type-7 External LSAs received from the NSSA into a single type-5 External LSA.

When an NSSA contains more than one ABR, OSPF elects one of the ABRs to perform the LSA translation for NSSA. OSPF elects the ABR with the highest router ID. If the elected ABR becomes unavailable, OSPF automatically elects the ABR with the next highest router ID to take over translation of LSAs for the NSSA. The election process for NSSA ABRs is automatic.

Example

To set up the OSPF areas shown in [Figure 158](#) on page 1227, enter the following commands.

```
Brocade(config-ospf-router)#area 192.5.1.0
Brocade(config-ospf-router)#area 200.5.0.0
Brocade(config-ospf-router)#area 195.5.0.0
Brocade(config-ospf-router)#area 0.0.0.0
Brocade(config-ospf-router)#write memory
```

Syntax: `area <num> | <ip-addr>`

The `<num> | <ip-addr>` parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 through 18.

NOTE

You can assign one area on a router interface. For example, if the system or chassis module has 16 ports, 16 areas are supported on the chassis or module.

Assigning a totally stubby area

By default, the Layer 3 Switch sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of link state advertisements (LSAs) sent into a stub area by configuring the Layer 3 Switch to stop sending summary LSAs (type 3 LSAs) into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs, but the Layer 3 Switch still accepts summary LSAs from OSPF neighbors and floods them to other neighbors. The Layer 3 Switch can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you enter a command or apply a Web management option to disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the Layer 3 Switch flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE

This feature applies only when the Layer 3 Switch is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

This feature does not apply to Not-So-Stubby Areas (NSSAs).

To disable summary LSAs for a stub area, enter commands such as the following.

```
Brocade(config-ospf-router)#area 40 stub 99 no-summary
```

Syntax: `area <num> | <ip-addr> stub <cost> [no-summary]`

The `<num> | <ip-addr>` parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 through 18.

The `stub <cost>` parameter specifies an additional cost for using a route to or from this area and can be from 1 through 16777215. There is no default. Normal areas do not use the cost parameter.

The `no-summary` parameter applies only to stub areas and disables summary LSAs from being sent into the area.

NOTE

You can assign one area on a router interface. For example, if the system or chassis module has 16 ports, 16 areas are supported on the chassis or module.

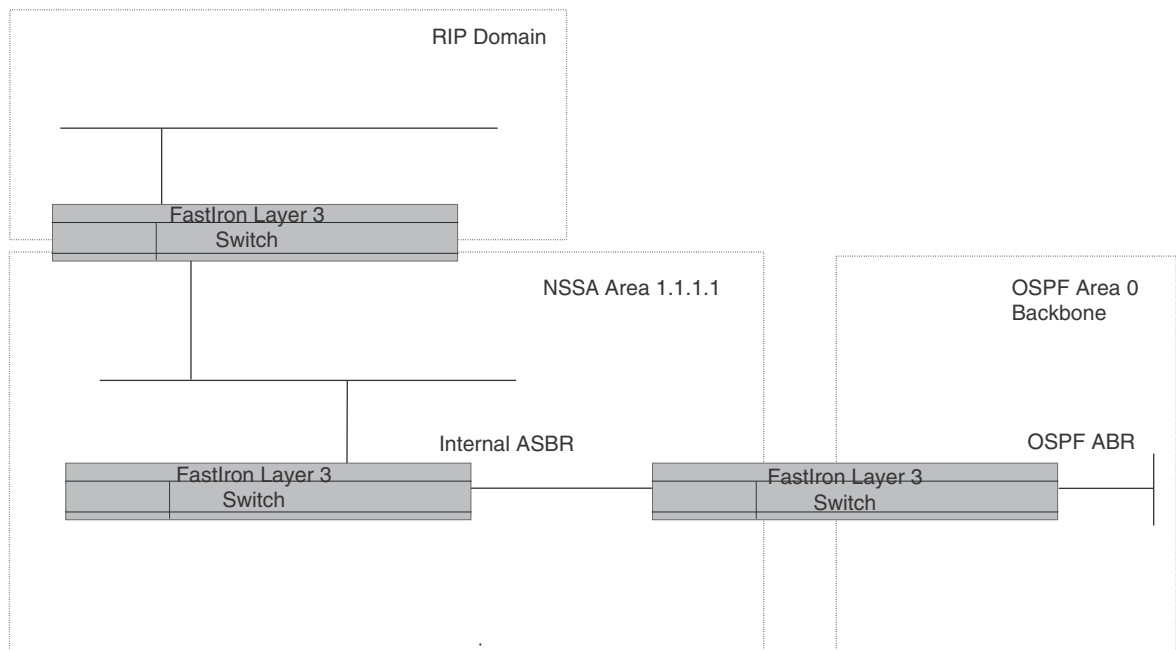
Assigning a Not-So-Stubby Area (NSSA)

The OSPF Not-So-Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information from the NSSA into other areas such as the backbone.

NSSAs are especially useful when you want to summarize Type-5 External LSAs (external routes) before forwarding them into an OSPF area. The OSPF specification (RFC 2328) prohibits summarization of Type-5 LSAs and requires OSPF to flood Type-5 LSAs throughout a routing domain. When you configure an NSSA, you can specify an address range for aggregating the external routes that the NSSA's ABR exports into other areas.

The Brocade implementation of NSSA is based on RFC 1587.

[Figure 162](#) shows an example of an OSPF network containing an NSSA.

FIGURE 162 OSPF network containing an NSSA

This example shows two routing domains, a RIP domain and an OSPF domain. The ASBR inside the NSSA imports external routes from RIP into the NSSA as Type-7 LSAs, which the ASBR floods throughout the NSSA.

The ABR translates the Type-7 LSAs into Type-5 LSAs. If an area range is configured for the NSSA, the ABR also summarizes the LSAs into an aggregate LSA before flooding the Type-5 LSAs into the backbone.

Since the NSSA is partially “stubby” the ABR does not flood external LSAs from the backbone into the NSSA. To provide access to the rest of the Autonomous System (AS), the ABR generates a default Type-7 LSA into the NSSA.

Configuring an NSSA

To configure OSPF area 1.1.1.1 as an NSSA, enter the following commands.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#area 1.1.1.1 nssa 1
Brocade(config-ospf-router)#write memory
```

Syntax: `area <num> | <ip-addr> nssa <cost> | default-information-originate`

The `<num> | <ip-addr>` parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 through 18.

The `nssa <cost> | default-information-originate` parameter specifies that this is a Not-So-Stubby Area (NSSA). The `<cost>` specifies an additional cost for using a route to or from this NSSA and can be from 1 through 16777215. There is no default. Normal areas do not use the cost parameter. Alternatively, the `default-information-originate` parameter causes the Layer 3 Switch to inject the default route into the NSSA.

NOTE

The Layer 3 Switch does not inject the default route into an NSSA by default.

NOTE

You can assign one area on a router interface. For example, if the system or chassis module has 16 ports, 16 areas are supported on the chassis or module.

To configure additional parameters for OSPF interfaces in the NSSA, use the **ip ospf area...** command at the interface level of the CLI.

Configuring a summary address for the NSSA

If you want the ABR that connects the NSSA to other areas to summarize the routes in the NSSA before translating them into Type-5 LSAs and flooding them into the other areas, configure a summary address. The ABR creates an aggregate value based on the summary address. The aggregate value becomes the address that the ABR advertises instead of advertising the individual addresses represented by the aggregate.

To configure a summary address in NSSA 1.1.1.1, enter the following commands. This example assumes that you have already configured NSSA 1.1.1.1.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#summary-address 209.157.22.1 255.255.0.0
Brocade(config-ospf-router)#write memory
```

Syntax: [no] **summary-address** <ip-addr> <ip-mask>

The <ip-addr> parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The <ip-mask> parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 209.157 are summarized into a single route.

Assigning an area range (optional)

You can assign a range for an area, but it is not required. Ranges allow a specific IP address and mask to represent a range of IP addresses within an area, so that only that reference range address is advertised to the network, instead of all the addresses within that range. Each area can have up to 32 range addresses.

Example

To define an area range for subnets on 193.45.5.1 and 193.45.6.2, enter the following commands.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#area 192.45.5.1 range 193.45.0.0 255.255.0.0
Brocade(config-ospf-router)#area 193.45.6.2 range 193.45.0.0 255.255.0.0
```

Syntax: **area** <num> | <ip-addr> **range** <ip-addr> <ip-mask>

The <num> | <ip-addr> parameter specifies the area number, which can be in IP address format.

The **range** <ip-addr> parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The `<ip-mask>` parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 193.45 are summarized into a single route.

Assigning interfaces to an area

Once you define OSPF areas, you can assign interfaces to the areas. All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

To assign interface 1/8 to area 195.5.0.0 and then save the changes, enter the following commands.

```
Brocade(config-ospf-router)#interface e 1/8
Brocade(config-if-1/8)#ip ospf area 195.5.0.0
Brocade(config-if-1/8)#write memory
```

Modifying interface defaults

OSPF has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

Port default values can be modified using the following commands at the interface configuration level of the CLI:

- **ip ospf area** `<ip-addr>`
- **ip ospf auth-change-wait-time** `<secs>`
- **ip ospf authentication-key** `[0 | 1] <string>`
- **ip ospf cost** `<num>`
- **ip ospf dead-interval** `<value>`
- **ip ospf hello-interval** `<value>`
- **ip ospf md5-authentication key-activation-wait-time** `<num>` | **key-id** `<num>` `[0 | 1]` **key** `<string>`
- **ip ospf passive**
- **ip ospf priority** `<value>`
- **ip ospf retransmit-interval** `<value>`
- **ip ospf transmit-delay** `<value>`

For a complete description of these parameters, see the summary of OSPF port parameters in the next section.

OSPF interface parameters

The following parameters apply to OSPF interfaces.

Area: Assigns an interface to a specific area. You can assign either an IP address or number to represent an OSPF Area ID. If you assign a number, it can be any value from 0 through 2,147,483,647.

Auth-change-wait-time: OSPF gracefully implements authentication changes to allow all routers to implement the change and thus prevent disruption to neighbor adjacencies. During the authentication-change interval, both the old and new authentication information is supported. The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 through 14400 seconds.

Authentication-key: OSPF supports three methods of authentication for each interface—none, simple password, and MD5. Only one method of authentication can be active on an interface at a time. The default authentication value is none, meaning no authentication is performed.

The simple password method of authentication requires you to configure an alphanumeric password on an interface. The simple password setting takes effect immediately. All OSPF packets transmitted on the interface contain this password. Any OSPF packet received on the interface is checked for this password. If the password is not present, then the packet is dropped. The password can be up to eight characters long.

The MD5 method of authentication requires you to configure a key ID and an MD5 Key. The key ID is a number from 1 through 255 and identifies the MD5 key that is being used. The MD5 key can be up to sixteen alphanumeric characters long.

Cost: Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps and 1000 Mbps (1 Gbps) links. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for both 100 Mbps and 1000 Mbps links is 1, because the speed of 1000 Mbps was not in use at the time the OSPF cost formula was devised.

Dead-interval: Indicates the number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The value can be from 1 through 65535 seconds. The default is 40 seconds.

Hello-interval: Represents the length of time between the transmission of hello packets. The value can be from 1 through 65535 seconds. The default is 10 seconds.

MD5-authentication activation wait time: The number of seconds the Layer 3 Switch waits until placing a new MD5 key into effect. The wait time provides a way to gracefully transition from one MD5 key to another without disturbing the network. The wait time can be from 0 through 14400 seconds. The default is 300 seconds (5 minutes).

MD5-authentication key ID and key: A method of authentication that requires you to configure a key ID and an MD5 key. The key ID is a number from 1 through 255 and identifies the MD5 key that is being used. The MD5 key consists of up to 16 alphanumeric characters. The MD5 is encrypted and included in each OSPF packet transmitted.

Passive: When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network. OSPF interfaces are active by default.

NOTE

This option affects all IP subnets configured on the interface. If you want to disable OSPF updates only on some of the IP subnets on the interface, use the **ospf-ignore** or **ospf-passive** parameter with the **ip address** command. Refer to [“Assigning an IP address to an Ethernet port”](#) on page 965.

Priority: Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The value can be from 0 through 255. The default is 1. If you set the priority to 0, the Layer 3 Switch does not participate in DR and BDR election.

Retransmit-interval: The time between retransmissions of link-state advertisements (LSAs) to adjacent routers for this interface. The value can be from 0 through 3600 seconds. The default is 5 seconds.

Transit-delay: The time it takes to transmit Link State Update packets on this interface. The value can be from 0 through 3600 seconds. The default is 1 second.

Encrypted display of the authentication string or MD5 authentication key

The optional **0 | 1** parameter with the **authentication-key** and **md5-authentication key-id** parameters affects encryption.

For added security, FastIron devices encrypt display of the password or authentication string. Encryption is enabled by default. The software also provides an optional parameter to disable encryption of a password or authentication string, on an individual OSPF area or OSPF interface basis.

When encryption of the passwords or authentication strings is enabled, they are encrypted in the CLI regardless of the access level you are using. In the Web Management Interface, the passwords or authentication strings are encrypted at the read-only access level but are visible at the read-write access level.

The encryption option can be omitted (the default) or can be one of the following:

- **0** – Disables encryption for the password or authentication string you specify with the command. The password or string is shown as clear text in the running-config and the startup-config file. Use this option if you do not want display of the password or string to be encrypted.
- **1** – Assumes that the password or authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE

If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

If you want to display the authentication string in the output of the **show ip ospf interface** command, enter the following commands.

```
Brocade(config)# enable password-display
Brocade# show ip ospf interface 1.1.1.1
```

The **enable password-display** command enables display of the authentication string, but only in the output of the **show ip ospf interface** command. Display of the string is still encrypted in the startup-config file and running-config. Enter the command at the global CONFIG level of the CLI.

Changing the timer for OSPF authentication changes

When you make an OSPF authentication change, the software uses the authentication-change timer to gracefully implement the change. The software implements the change in the following ways:

- Outgoing OSPF packets – After you make the change, the software continues to use the old authentication to send packets, during the remainder of the current authentication-change interval. After this, the software uses the new authentication for sending packets.
- Inbound OSPF packets – The software accepts packets containing the new authentication and continues to accept packets containing the older authentication for two authentication-change intervals. After the second interval ends, the software accepts packets only if they contain the new authentication key.

The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 through 14400 seconds.

OSPF provides graceful authentication change for all the following types of authentication changes in OSPF:

- Changing authentication methods from one of the following to another of the following:
 - Simple text password
 - MD5 authentication
 - No authentication
- Configuring a new simple text password or MD5 authentication key
- Changing an existing simple text password or MD5 authentication key

To change the authentication-change interval, enter a command such as the following at the interface configuration level of the CLI.

```
Brocade(config-if-2/5)#ip ospf auth-change-wait-time 400
```

Syntax: [no] ip ospf auth-change-wait-time <secs>

The <secs> parameter specifies the interval and can be from 0 through 14400 seconds. The default is 300 seconds (5 minutes).

NOTE

For backward compatibility, the **ip ospf md5-authentication key-activation-wait-time <seconds>** command is still supported.

Block flooding of outbound LSAs on specific OSPF interfaces

By default, the Layer 3 Switch floods all outbound LSAs on all the OSPF interfaces within an area. You can configure a filter to block outbound LSAs on an OSPF interface. This feature is particularly useful when you want to block LSAs from some, but not all, of the interfaces attached to the area.

After you apply filters to block the outbound LSAs, the filtering occurs during the database synchronization and flooding.

If you remove the filters, the blocked LSAs are automatically re-flooded. You do not need to reset OSPF to re-flood the LSAs.

NOTE

You cannot block LSAs on virtual links.

To apply a filter to an OSPF interface to block flooding of outbound LSAs on the interface, enter the following commands at the Interface configuration level for that interface.

```
Brocade(config-if-1/1)#ip ospf database-filter all out
Brocade(config-if-1/1)#clear ip ospf all
```

The first command in this example blocks all outbound LSAs on the OSPF interface configured on port 1/1. The second command resets OSPF and places the command into effect immediately.

Syntax: [no] ip ospf database-filter all out

To remove the filter, enter a command such as the following.

```
Brocade(config-if-1/1)#no ip ospf database-filter all out
```

Configuring an OSPF non-broadcast interface

Layer 3 switches support Non-Broadcast Multi-Access (NBMA) networks. This feature enables you to configure an interface on a Brocade device to send OSPF traffic to its neighbor as unicast packets rather than broadcast packets.

OSPF routers generally use broadcast packets to establish neighbor relationships and broadcast route updates on Ethernet and virtual interfaces (VEs). In this release, as an alternative, you can configure the Brocade device to use unicast packets for this purpose. This can be useful in situations where multicast traffic is not feasible (for example when a firewall does not allow multicast packets).

On a non-broadcast interface, the routers at the other end of this interface must also be configured as non-broadcast and neighbor routers. There is no restriction on the number of routers sharing a non-broadcast interface (for example, through a hub or switch).

NOTE

Only Ethernet interfaces or VEs can be configured as non-broadcast interfaces.

To configure an OSPF interface as a non-broadcast interface, enable the feature on a physical interface or a VE, following the **ip ospf area** statement, and then specify the IP address of the neighbor in the OSPF configuration. The non-broadcast interface configuration must be done on the OSPF routers on both ends of the link.

For example, the following commands configure VE 20 as a non-broadcast interface.

```
Brocade(config)#int ve 20
Brocade(config-vif-20)#ip ospf area 0
Brocade(config-vif-20)#ip ospf network non-broadcast
Brocade(config-vif-20)#exit
```

Syntax: [no] ip ospf network non-broadcast

The following commands specify 1.1.20.1 as an OSPF neighbor address. The address specified must be in the same subnet as a non-broadcast interface.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#neighbor 1.1.20.1
```

For example, to configure the feature in a network with three routers connected by a hub or switch, each router must have the linking interface configured as a non-broadcast interface, and both of the other routers must be specified as neighbors.

The output of the **show ip ospf interface** command has been enhanced to display information about non-broadcast interfaces and neighbors that are configured in the same subnet.

Example of specifying OSPF neighbor address

```
Brocade#show ip ospf interface
v20,OSPF enabled
    IP Address 1.1.20.4, Area 0
    OSPF state BD, Pri 1, Cost 1, Options 2, Type non-broadcast Events 6
    Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
    DR:  Router ID 1.1.13.1      Interface Address 1.1.20.5
    BDR: Router ID 2.2.2.1      Interface Address 1.1.20.4
    Neighbor Count = 1, Adjacent Neighbor Count= 2
    Non-broadcast neighbor config: 1.1.20.1, 1.1.20.2, 1.1.20.3, 1.1.20.5,
    Neighbor:      1.1.20.5
    Authentication-Key:None
    MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

In the Type field, “non-broadcast” indicates that this is a non-broadcast interface. When the interface type is non-broadcast, the Non-broadcast neighbor config field displays the neighbors that are configured in the same subnet. If no neighbors are configured in the same subnet, a message such as the following is displayed.

```
***Warning! no non-broadcast neighbor config in 1.1.100.1 255.255.255.0
```

Assigning virtual links

All ABRs (area border routers) must have either a direct or indirect link to the OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to the area backbone, the ABR can configure a **virtual link** to another router within the same area, which has a physical connection to the area backbone.

The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection), and the ABR requiring a logical connection to the backbone.

Two parameters fields must be defined for all virtual links—transit area ID and neighbor router:

- The *transit area ID* represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- The *neighbor router* field is the router ID (IP address) of the router that is physically connected to the backbone, when assigned from the router interface requiring a logical connection. When assigning the parameters from the router with the physical connection, the router ID is the IP address of the router requiring a logical connection to the backbone.

NOTE

By default, the Brocade router ID is the IP address configured on the lowest numbered loopback interface. If the Layer 3 Switch does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, refer to [“Changing the router ID”](#) on page 976.

NOTE

When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

FIGURE 163 Defining OSPF virtual links within a network

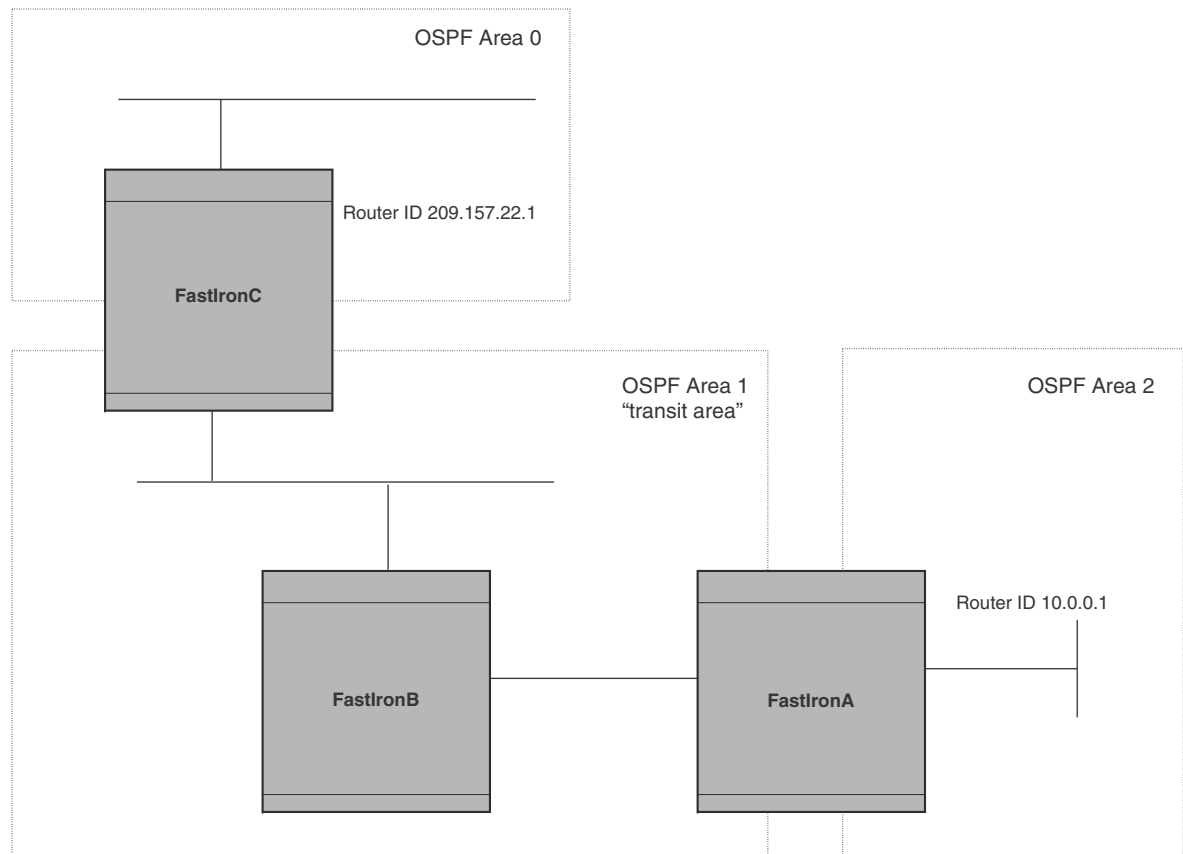
**Example**

Figure 163 shows an OSPF area border router, FastIronA, that is cut off from the backbone area (area 0). To provide backbone access to FastIronA, you can add a virtual link between FastIronA and FastIronC using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on FastIronA, enter the following commands.

```
BrocadeA(config-ospf-router)#area 1 virtual-link 209.157.22.1
BrocadeA(config-ospf-router)#write memory
```

Enter the following commands to configure the virtual link on FastIronC.

```
BrocadeC(config-ospf-router)#area 1 virtual-link 10.0.0.1
BrocadeC(config-ospf-router)#write memory
```

Syntax: `area <ip-addr> | <num> virtual-link <router-id>`

`[authentication-key | dead-interval | hello-interval | retransmit-interval | transmit-delay <value>]`

The **area** *<ip-addr> | <num>* parameter specifies the transit area.

The *<router-id>* parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a Brocade Layer 3 Switch, enter the **show ip** command.

Refer to [“Modifying virtual link parameters”](#) on page 1249 for descriptions of the optional parameters.

Modifying virtual link parameters

OSPF has some parameters that you can modify for virtual links. Notice that these are the same parameters as the ones you can modify for physical interfaces.

You can modify default values for virtual links using the following CLI command at the OSPF router level of the CLI, as shown in the following syntax.

Syntax: **area** *<num> | <ip-addr>* **virtual-link** *<ip-addr>* [**authentication-key** [0 | 1] *<string>*] [**dead-interval** *<num>*] [**hello-interval** *<num>*] [**md5-authentication key-activation-wait-time** *<num>* | **key-id** *<num>*] [0 | 1] **key** *<string>*] [**retransmit-interval** *<num>*] [**transmit-delay** *<num>*]

The parameters are described in the next section.

Virtual link parameter descriptions

You can modify the following virtual link interface parameters.

Authentication Key: This parameter allows you to assign different authentication methods on a port-by-port basis. OSPF supports three methods of authentication for each interface—none, simple password, and MD5. Only one method of authentication can be active on an interface at a time.

The simple password method of authentication requires you to configure an alphanumeric password on an interface. The password can be up to eight characters long. The simple password setting takes effect immediately. All OSPF packets transmitted on the interface contain this password. All OSPF packets received on the interface are checked for this password. If the password is not present, the packet is dropped.

The MD5 method of authentication encrypts the authentication key you define. The authentication is included in each OSPF packet transmitted.

MD5 Authentication Key: When simple authentication is enabled, the key is an alphanumeric password of up to eight characters. When MD5 is enabled, the key is an alphanumeric password of up to 16 characters that is later encrypted and included in each OSPF packet transmitted. You must enter a password in this field when the system is configured to operate with either simple or MD5 authentication.

MD5 Authentication Key ID: The Key ID is a number from 1 through 255 and identifies the MD5 key that is being used. This parameter is required to differentiate among multiple keys defined on a router.

MD5 Authentication Wait Time: This parameter determines when a newly configured MD5 authentication key is valid. This parameter provides a graceful transition from one MD5 key to another without disturbing the network. All new packets transmitted after the key activation wait time interval use the newly configured MD5 Key. OSPF packets that contain the old MD5 key are accepted for up to five minutes after the new MD5 key is in operation.

The range for the key activation wait time is from 0 through 14400 seconds. The default value is 300 seconds.

Hello Interval: The length of time between the transmission of hello packets. The range is 1 through 65535 seconds. The default is 10 seconds.

Retransmit Interval: The interval between the re-transmission of link state advertisements to router adjacencies for this interface. The range is 0 through 3600 seconds. The default is 5 seconds.

Transmit Delay: The period of time it takes to transmit Link State Update packets on the interface. The range is 0 through 3600 seconds. The default is 1 second.

Dead Interval: The number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The range is 1 through 65535 seconds. The default is 40 seconds.

Encrypted display of the authentication string or MD5 authentication key

The optional **0** | **1** parameter with the **authentication-key** and **md5-authentication key-id** parameters affects encryption.

For added security, FastIron devices encrypt display of the password or authentication string. Encryption is enabled by default. The software also provides an optional parameter to disable encryption of a password or authentication string, on an individual OSPF area or OSPF interface basis.

When encryption of the passwords or authentication strings is enabled, they are encrypted in the CLI regardless of the access level you are using. In the Web Management Interface, the passwords or authentication strings are encrypted at the read-only access level but are visible at the read-write access level.

The encryption option can be omitted (the default) or can be one of the following:

- **0** – Disables encryption for the password or authentication string you specify with the command. The password or string is shown as clear text in the running-config and the startup-config file. Use this option if you do not want display of the password or string to be encrypted.
- **1** – Assumes that the password or authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE

If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

Changing the reference bandwidth for the cost on OSPF interfaces

Each interface on which OSPF is enabled has a cost associated with it. The Layer 3 Switch advertises its interfaces and their costs to OSPF neighbors. For example, if an interface has an OSPF cost of ten, the Layer 3 Switch advertises the interface with a cost of ten to other OSPF routers.

By default, an interface OSPF cost is based on the port speed of the interface. The cost is calculated by dividing the reference bandwidth by the port speed. The default reference bandwidth is 100 Mbps, which results in the following default costs:

- 10 Mbps port – 10
- All other port speeds – 1

You can change the reference bandwidth, to change the costs calculated by the software.

The software uses the following formula to calculate the cost.

$$\text{Cost} = \text{reference-bandwidth} / \text{interface-speed}$$

If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port cost = $100/10 = 10$
- 100 Mbps port cost = $100/100 = 1$
- 1000 Mbps port cost = $100/1000 = 0.10$, which is rounded up to 1
- 155 Mbps port cost = $100/155 = 0.65$, which is rounded up to 1
- 622 Mbps port cost = $100/622 = 0.16$, which is rounded up to 1
- 2488 Mbps port cost = $100/2488 = 0.04$, which is rounded up to 1

For 10 Gbps OSPF interfaces, in order to differentiate the costs between 100 Mbps, 1000 Mbps, and 10,000 Mbps interfaces, you can set the auto-cost reference bandwidth to 10000, whereby each slower link is given a higher cost, as follows:

- 10 Mbps port cost = $10000/10 = 1000$
- 100 Mbps port cost = $10000/100 = 100$
- 1000 Mbps port cost = $10000/1000 = 10$
- 10000 Mbps port cost = $10000/10000 = 1$

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- Trunk group – The combined bandwidth of all the ports.
- Virtual interface – The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

The default reference bandwidth is 100 Mbps. You can change the reference bandwidth to a value from 1 through 4294967.

If a change to the reference bandwidth results in a cost change to an interface, the Layer 3 Switch sends a link-state update to update the costs of interfaces advertised by the Layer 3 Switch.

NOTE

If you specify the cost for an individual interface, the cost you specify overrides the cost calculated by the software.

Interface types to which the reference bandwidth does not apply

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 0.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.
- The bandwidth for tunnel interfaces is 9 Kbps and is not affected by the auto-cost feature.

Changing the reference bandwidth

To change the reference bandwidth, enter the **auto-cost reference-bandwidth** command at the OSPF configuration level of the CLI.

```
Brocade(config-ospf-router)#auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port cost = $500/10 = 50$
- 100 Mbps port cost = $500/100 = 5$
- 1000 Mbps port cost = $500/1000 = 0.5$, which is rounded up to 1
- 155 Mbps port cost = $500/155 = 3.23$, which is rounded up to 4
- 622 Mbps port cost = $500/622 = 0.80$, which is rounded up to 1
- 2488 Mbps port cost = $500/2488 = 0.20$, which is rounded up to 1

The costs for 10 Mbps, 100 Mbps, and 155 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: [no] **auto-cost reference-bandwidth** <num>

The <num> parameter specifies the reference bandwidth and can be a value from 1 through 4294967. The default is 100. For 10 Gbps OSPF interfaces, in order to differentiate the costs between 100 Mbps, 1000 Mbps, and 10,000 Mbps interfaces, set the auto-cost reference bandwidth to 10000, whereby each slower link is given a higher cost.

To restore the reference bandwidth to its default value and thus restore the default costs of interfaces to their default values, enter the following command.

```
Brocade(config-ospf-router)#no auto-cost reference-bandwidth
```

Defining redistribution filters

Route redistribution imports and translates different protocol routes into a specified protocol type. On Brocade routers, redistribution is supported for static routes, OSPF, RIP, and BGP4. When you configure redistribution for RIP, you can specify that static, OSPF, or BGP4 routes are imported into RIP routes. Likewise, OSPF redistribution supports the import of static, RIP, and BGP4 routes into OSPF routes. BGP4 supports redistribution of static, RIP, and OSPF routes into BGP4.

NOTE

The Layer 3 Switch advertises the default route into OSPF even if redistribution is not enabled, and even if the default route is learned through an IBGP neighbor. IBGP routes (including the default route) are not redistributed into OSPF by OSPF redistribution (for example, by the OSPF **redistribute** command).

In [Figure 164](#) on page 1253, an administrator wants to configure the FastIron Layer 3 Switch acting as the ASBR (Autonomous System Boundary Router) between the RIP domain and the OSPF domain to redistribute routes between the two domains.

NOTE

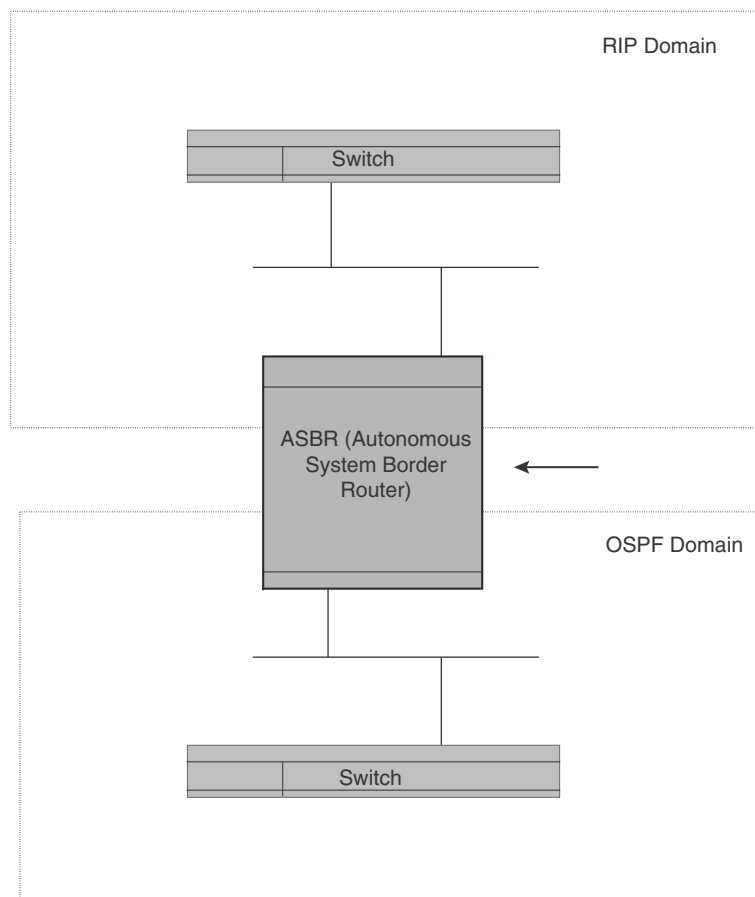
The ASBR must be running both RIP and OSPF protocols to support this activity.

To configure for redistribution, define the redistribution tables with deny and permit redistribution filters. Use the **deny redistribute** and **permit redistribute** commands for OSPF at the OSPF router level.

NOTE

Do not enable redistribution until you have configured the redistribution filters. If you enable redistribution before you configure the redistribution filters, the filters will not take affect and all routes will be distributed.

FIGURE 164 Redistributing OSPF and static routes to RIP routes



Example of redefining distribution filters

To configure the FastIron Layer 3 Switch acting as an ASBR in [Figure 164](#) to redistribute OSPF, BGP4, and static routes into RIP, enter the following commands.

```
BrocadeASBR(config)#router rip
BrocadeASBR(config-rip-router)#permit redistribute 1 all
BrocadeASBR(config-rip-router)#write memory
```

NOTE

Redistribution is permitted for all routes by default, so the **permit redistribute 1 all** command in the example above is shown for clarity but is not required.

You also have the option of specifying import of just OSPF, BGP4, or static routes, as well as specifying that only routes for a specific network or with a specific cost (metric) be imported, as shown in the following command syntax.

Syntax: **deny** | **permit redistribute** <filter-num> **all** | **bgp** | **connected** | **rip** | **static**
[**address** <ip-addr> <ip-mask> [**match-metric** <value> [**set-metric** <value>]]]

Example

To redistribute RIP, static, and BGP4 routes into OSPF, enter the following commands on the Layer 3 Switch acting as an ASBR.

```
BrocadeASBR(config)#router ospf
BrocadeASBR(config-ospf-router)#permit redistribute 1 all
BrocadeASBR(config-ospf-router)#write memory
```

Syntax: **deny** | **permit redistribute** <filter-num> **all** | **bgp** | **connected** | **rip** | **static**
address <ip-addr> <ip-mask>
[**match-metric** <value> | **set-metric** <value>]

NOTE

Redistribution is permitted for all routes by default, so the **permit redistribute 1 all** command in the example above is shown for clarity but is not required.

You also have the option of specifying import of just OSPF, BGP4, or static routes, as well as specifying that only routes for a specific network or with a specific cost (metric) be imported, as shown in the following command syntax.

For example, to enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#redistribution rip
Brocade(config-ospf-router)#redistribution static
Brocade(config-ospf-router)#write memory
```

Syntax: [**no**] **redistribution** **bgp** | **connected** | **rip** | **static** [**route-map** <map-name>]

NOTE

The **redistribution** command does not perform the same function as the **permit redistribute** and **deny redistribute** commands. The **redistribute** commands allow you to control redistribution of routes by filtering on the IP address and network mask of a route. The **redistribution** commands enable redistribution for routes of specific types (static, directly connected, and so on). Configure all your redistribution filters before enabling redistribution.

NOTE

Do not enable redistribution until you have configured the redistribution filters. If you enable redistribution before you configure the redistribution filters, the filters will not take effect and all routes will be distributed.

Preventing specific OSPF routes from being installed in the IP route table

By default, all OSPF routes in the OSPF route table are eligible for installation in the IP route table. You can configure a distribution list to explicitly deny specific routes from being eligible for installation in the IP route table.

NOTE

This feature does not block receipt of LSAs for the denied routes. The Layer 3 Switch still receives the routes and installs them in the OSPF database. The feature only prevents the software from installing the denied OSPF routes into the IP route table.

To configure an OSPF distribution list:

- Configure a standard or extended ACL that identifies the routes you want to deny. Using a standard ACL lets you deny routes based on the destination network, but does not filter based on the network mask. To also filter based on the destination network network mask, use an extended ACL.
- Configure an OSPF distribution list that uses the ACL as input.

NOTE

If you change the ACL after you configure the OSPF distribution list, you must clear the IP route table to place the changed ACL into effect. To clear the IP route table, enter the **clear ip route** command at the Privileged EXEC level of the CLI.

The following sections show how to use the CLI to configure an OSPF distribution list. Separate examples are provided for standard and extended ACLs.

NOTE

The examples show named ACLs. However, you also can use a numbered ACL as input to the OSPF distribution list.

Using a standard ACL as input to the distribution list

To use a standard ACL to configure an OSPF distribution list for denying specific routes, enter commands such as the following.

```
Brocade(config)#ip access-list standard no_ip
Brocade(config-std-nACL)#deny 4.0.0.0 0.255.255.255
Brocade(config-std-nACL)#permit any any
Brocade(config-std-nACL)#exit
Brocade(config)#router ospf
Brocade(config-ospf-router)#distribute-list no_ip in
```

The first three commands configure a standard ACL that denies routes to any 4.x.x.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 4.x.x.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

Syntax: [no] **distribute-list** <ACL-name> | <ACL-id> in [<interface type>] [<interface number>]

Syntax: [no] **ip access-list standard** <ACL-name> | <ACL-id>

Syntax: **deny** | **permit** <source-ip> <wildcard>

The <ACL-name> | <ACL-id> parameter specifies the ACL name or ID.

The **in** command applies the ACL to incoming route updates.

The <interface number> parameter specifies the interface number on which to apply the ACL. Enter only one valid interface number. If necessary, use the **show interface brief** command to display a list of valid interfaces. If you do not specify an interface, the Brocade device applies the ACL to all incoming route updates.

If you do not specify an interface type and interface number, the device applies the OSPF distribution list to all incoming route updates.

The **deny** | **permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The <source-ip> parameter specifies the source address for the policy. Because this ACL is input to an OSPF distribution list, the <source-ip> parameter actually is specifying the destination network of the route.

The <wildcard> parameter specifies the portion of the source address to match against. The <wildcard> is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 4.0.0.0 0.255.255.255 mean that all 4.x.x.x networks match the ACL.

If you want the policy to match on all destination networks, enter **any any**.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of “4.0.0.0 0.255.255.255” as “4.0.0.0/8”. The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros.

NOTE

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in “/<mask-bits>” format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

Using an extended ACL as input to the distribution list

You can use an extended ACL with an OSPF distribution list to filter OSPF routes based on the network mask of the destination network.

To use an extended ACL to configure an OSPF distribution list for denying specific routes, enter commands such as the following.

```
Brocade(config)#ip access-list extended no_ip
Brocade(config-ext-nACL)#deny ip 4.0.0.0 0.255.255.255 255.255.0.0 0.0.255.255
Brocade(config-ext-nACL)#permit ip any any
Brocade(config-ext-nACL)#exit
Brocade(config)#router ospf
Brocade(config-ospf-router)#distribute-list no_ip in
```

The first three commands configure an extended ACL that denies routes to any 4.x.x.x destination network with a 255.255.0.0 network mask and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 4.x.x.x destination network with network mask 255.255.0.0 from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

Syntax: [no] ip access-list extended <ACL-name> | <ACL-id>

Syntax: deny | permit <ip-protocol> <source-ip> <wildcard> <destination-ip> <wildcard>

The <ACL-name> | <ACL-id> parameter specifies the ACL name or ID.

The **deny** | **permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The <ip-protocol> parameter indicates the type of IP packet you are filtering. When using an extended ACL as input for an OSPF distribution list, specify **ip**.

Because this ACL is input to an OSPF distribution list, the <source-ip> parameter actually specifies the destination network of the route.

The <wildcard> parameter specifies the portion of the source address to match against. The <wildcard> is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 4.0.0.0 0.255.255.255 mean that all 4.x.x.x networks match the ACL.

If you want the policy to match on all network addresses, enter **any any**.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of “4.0.0.0 0.255.255.255” as “4.0.0.0/8”. The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros.

NOTE

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in “/<mask-bits>” format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** commands.

Because this ACL is input to an OSPF distribution list, the <destination-ip> parameter actually specifies the subnet mask of the route.

The <wildcard> parameter specifies the portion of the subnet mask to match against. For example, the <destination-ip> and <wildcard> values 255.255.255.255 0.0.0.255 mean that subnet mask /24 and longer match the ACL.

If you want the policy to match on all network masks, enter **any any**.

Modifying the default metric for redistribution

The default metric is a global parameter that specifies the cost applied to all OSPF routes by default. The default value is 10. You can assign a cost from 1 through 15.

NOTE

You also can define the cost on individual interfaces. The interface cost overrides the default cost.

To assign a default metric of 4 to all routes imported into OSPF, enter the following commands.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#default-metric 4
```

Syntax: **default-metric** <value>

The <value> can be from 1 through 16,777,215. The default is 10.

Enabling route redistribution

To enable route redistribution, use one of the following methods.

NOTE

Do not enable redistribution until you have configured the redistribution filters. Otherwise, you might accidentally overload the network with routes you did not intend to redistribute.

To enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#redistribution rip
Brocade(config-ospf-router)#redistribution static
Brocade(config-ospf-router)#write memory
```


Example using a route map

To configure a route map and use it for redistribution of routes into OSPF, enter commands such as the following.

```
Brocade(config)#ip route 1.1.0.0 255.255.0.0 207.95.7.30
Brocade(config)#ip route 1.2.0.0 255.255.0.0 207.95.7.30
Brocade(config)#ip route 1.3.0.0 255.255.0.0 207.95.7.30
Brocade(config)#ip route 4.1.0.0 255.255.0.0 207.95.6.30
Brocade(config)#ip route 4.2.0.0 255.255.0.0 207.95.6.30
Brocade(config)#ip route 4.3.0.0 255.255.0.0 207.95.6.30
Brocade(config)#ip route 4.4.0.0 255.255.0.0 207.95.6.30 5
Brocade(config)#route-map abc permit 1
Brocade(config-route-map abc)#match metric 5
Brocade(config-route-map abc)#set metric 8
Brocade(config-route-map abc)#router ospf
Brocade(config-ospf-router)#redistribution static route-map abc
```

The commands in this example configure some static IP routes, then configure a route map and use the route map for redistributing static IP routes into OSPF.

The **ip route** commands configure the static IP routes. The **route-map** command begins configuration of a route map called “abc”. The number indicates the route map entry (called the “instance”) you are configuring. A route map can contain multiple entries. The software compares packets to the route map entries in ascending numerical order and stops the comparison once a match is found.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribution static** command enables redistribution of static IP routes into OSPF, and uses route map “abc” to control the routes that are redistributed. In this example, the route map allows a static IP route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route table.

Syntax: [no] redistribution bgp | connected | rip | static [route-map <map-name>]

The **bgp | connected | rip | static** parameter specifies the route source.

The **route-map <map-name>** parameter specifies the route map name. The following match parameters are valid for OSPF redistribution:

- **match ip address | next-hop <ACL-num>**
- **match metric <num>**
- **match tag <tag-value>**

The following set parameters are valid for OSPF redistribution:

- **set ip next hop <ip-addr>**
- **set metric [+ | -]<num> | none**
- **set metric-type type-1 | type-2**
- **set tag <tag-value>**

NOTE

You must configure the route map before you configure a redistribution filter that uses the route map.

NOTE

When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

NOTE

For an external route that is redistributed into OSPF through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map. The **default-metric <num>** command has no effect on the route. This behavior is different from a route that is redistributed without using a route map. For a route redistributed without using a route map, the metric is set by the **default-metric <num>** command.

The following command shows the result of the redistribution filter. Because only one of the static IP routes configured above matches the route map, only one route is redistributed. Notice that the route metric is 5 before redistribution but is 8 after redistribution.

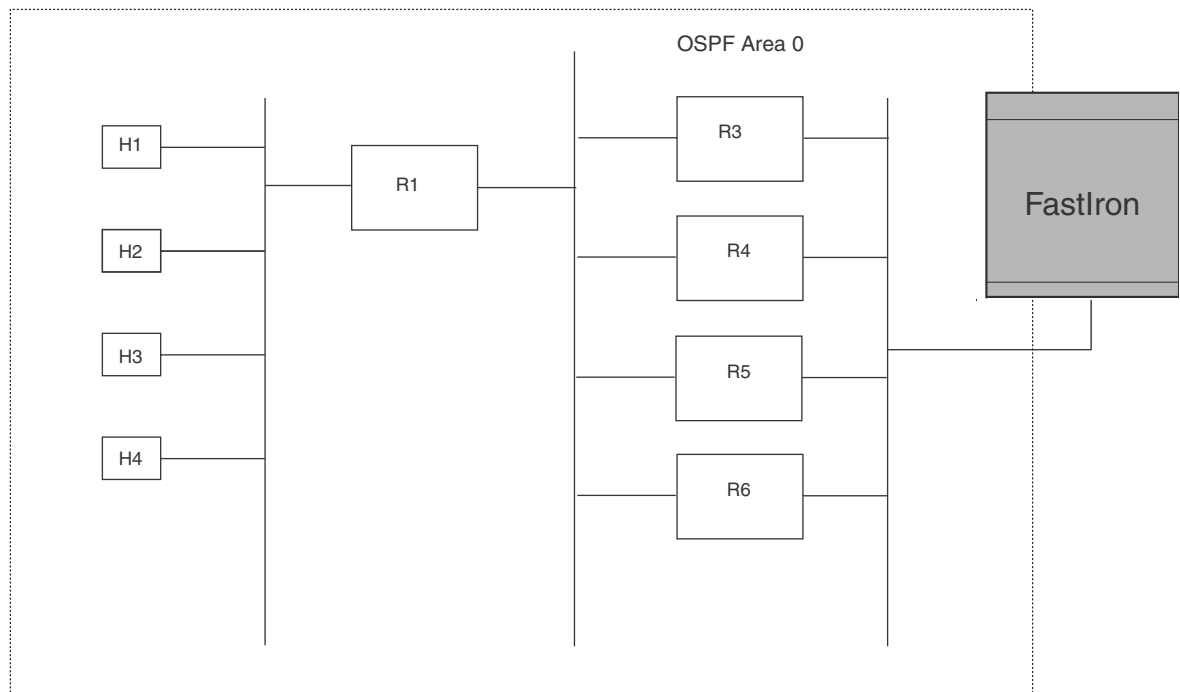
```
Brocade#show ip ospf database external extensive
```

Index	Aging	LS ID	Router	Netmask	Metric	Flag
1	2	4.4.0.0	10.10.10.60	ffff0000	80000008	0000

Disabling or re-enabling load sharing

Brocade routers can load share among up to eight equal-cost IP routes to a destination. By default, IP load sharing is enabled. The default is 4 equal-cost paths but you can specify from 2 to 6 paths.

The router software can use the route information it learns through OSPF to determine the paths and costs. [Figure 165](#) shows an example of an OSPF network containing multiple paths to a destination (in this case, R1).

FIGURE 165 Example OSPF network with four equal-cost paths

In the example in [Figure 165](#), the Brocade switch has four paths to R1:

- FI->R3
- FI->R4
- FI->R5
- FI->R6

Normally, the Brocade switch will choose the path to the R1 with the lower metric. For example, if R3 metric is 1400 and R4 metric is 600, the Brocade switch will always choose R4.

However, suppose the metric is the same for all four routers in this example. If the costs are the same, the switch now has four equal-cost paths to R1. To allow the switch to load share among the equal cost routes, enable IP load sharing. The software supports four equal-cost OSPF paths by default when you enable load sharing. You can specify from 2 to 6 paths.

NOTE

The Brocade switch is not source routing in these examples. The switch is concerned only with the paths to the next-hop routers, not the entire paths to the destination hosts.

OSPF load sharing is enabled by default when IP load sharing is enabled. To configure IP load sharing parameters, refer to [“Configuring IP load sharing”](#) on page 1001.

Configuring external route summarization

When the Layer 3 Switch is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the Layer 3 Switch, no action is taken if the Layer 3 Switch has already advertised the aggregate route; otherwise the Layer 3 Switch advertises the aggregate route. If an imported route that falls within a configured address range is removed by the Layer 3 Switch, no action is taken if there are other imported routes that fall within the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The Layer 3 Switch sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external LSDB overflow condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the Layer 3 Switch exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE

If you use redistribution filters in addition to address ranges, the Layer 3 Switch applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE

If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

To configure a summary address for OSPF routes, enter commands such as the following.

```
Brocade(config-ospf-router)#summary-address 10.1.0.0 255.255.0.0
```

The command in this example configures summary address 10.1.0.0, which includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. For all of these networks, only the address 10.1.0.0 (the parent route) is advertised in external LSAs. However, if the parent route has not been configured with a summary address, or if the summary address for the parent route is configured after the child route, the Layer 3 switch will advertise all routes. For example:

```
router ospf
area 0
summary-address 10.1.1.0 255.255.0.0 -> Advertised
summary-address 10.1.2.0 255.255.0.0 -> Advertised
summary-address 10.0.0.0 255.0.0.0 -> Advertised
```

Syntax: `summary-address <ip-addr> <ip-mask>`

The `<ip-addr>` parameter specifies the network address.

The `<ip-mask>` parameter specifies the network mask.

To display the configured summary addresses, use the **show ip ospf config** command at any level of the CLI. The summary addresses display at the bottom of the output as shown in the following example.

```
Brocade#show ip ospf config

some lines omitted for brevity...

OSPF Redistribution Address Ranges currently defined:
Range-Address      Subnetmask
1.0.0.0            255.0.0.0
1.0.1.0            255.255.255.0
1.0.2.0            255.255.255.0
```

Syntax: **show ip ospf config**

Configuring default route origination

When the Layer 3 Switch is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPF routing domain. This feature is called “default route origination” or “default information origination”.

By default, Brocade Layer 3 Switches do not advertise the default route into the OSPF domain. If you want the Layer 3 Switch to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the Layer 3 Switch advertises a type 5 default route that is flooded throughout the AS (except stub areas and NSSAs). In addition, internal NSSA ASBRs advertise their default routes as translatable type 7 default routes.

The Layer 3 Switch advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor.

NOTE

Brocade Layer 3 Switches never advertise the OSPF default route, regardless of other configuration parameters, unless you explicitly enable default route origination using the following method.

If the Layer 3 Switch is an ASBR, you can use the “always” option when you enable the default route origination. The always option causes the ASBR to create and advertise a default route if it does not already have one configured.

If default route origination is enabled and you disable it, the default route originated by the Layer 3 Switch is flushed. Default routes generated by other OSPF routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

NOTE

The ABR (Layer 3 Switch) will not inject the default route into an NSSA by default and the command described in this section will not cause the Layer 3 Switch to inject the default route into the NSSA. To inject the default route into an NSSA, use the **area <num> | <ip-addr> nssa default-information-originate** command. Refer to [“Assigning a Not-So-Stubby Area \(NSSA\)”](#) on page 1239.

To enable default route origination, enter the **default-information-originate** command.

```
Brocade(config-ospf-router)#default-information-originate
```

To disable the feature, enter the **no default-information-originate** command.

```
Brocade(config-ospf-router)#no default-information-originate
```

Syntax: **[no] default-information-originate [always] [metric <value>] [metric-type <type>]**

The **always** parameter advertises the default route regardless of whether the router has a default route. This option is disabled by default.

The **metric <value>** parameter specifies a metric for the default route. If this option is not used, the default metric is used for the route.

The **metric-type <type>** parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The <type> can be one of the following:

- 1 – Type 1 external route
- 2 – Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE

If you specify a metric and metric type, the values you specify are used even if you do not use the **always** option.

Modifying SPF timers

The Layer 3 Switch uses the following timers when calculating the shortest path for OSPF routes:

- **SPF delay** – When the Layer 3 Switch receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits five seconds. You can configure the SPF delay to a value from 0 through 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- **SPF hold time** – The Layer 3 Switch waits for a specific amount of time between consecutive SPF calculations. By default, the Layer 3 Switch waits ten seconds. You can configure the SPF hold time to a value from 0 through 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the delay and hold time to lower values to cause the Layer 3 Switch to change to alternate paths more quickly in the event of a route failure. Note that lower values require more CPU processing time.

You can change one or both of the timers. To do so, enter commands such as the following.

```
Brocade(config-ospf-router)#timers spf 10 20
```

The command in this example changes the SPF delay to 10 seconds and changes the SPF hold time to 20 seconds.

Syntax: **timers spf <delay> <hold-time>**

The <delay> parameter specifies the SPF delay.

The <hold-time> parameter specifies the SPF hold time.

To set the timers back to their default values, enter a command such as the following.

```
Brocade(config-ospf-router)#no timers spf 10 20
```

Modifying the redistribution metric type

The redistribution metric type is used by default for all routes imported into OSPF unless you specify different metrics for individual routes using redistribution filters. Type 2 specifies a big metric (three bytes). Type 1 specifies a small metric (two bytes). The default value is type 2.

To modify the default value to type 1, enter the following command.

```
Brocade(config-ospf-router)#metric-type type1
```

Syntax: `metric-type type1 | type2`

Administrative distance

Brocade Layer 3 Switches can learn about networks from various protocols, including Border Gateway Protocol version 4 (BGP4), RIP, and OSPF. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. The default administrative distance for OSPF routes is 110. Refer to [“Changing administrative distances”](#) on page 1374 for a list of the default distances for all route sources.

The router selects one route over another based on the source of the route information. To do so, the router can use the administrative distances assigned to the sources. You can bias the Layer 3 Switch decision by changing the default administrative distance for RIP routes.

Configuring administrative distance based on route type

You can configure a unique administrative distance for each type of OSPF route. For example, you can use this feature to prefer a static route over an OSPF inter-area route but you also want to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the Layer 3 Switch has multiple routes for the same network from different protocols. The Layer 3 Switch prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all these OSPF route types is 110.

NOTE

This feature does not influence the choice of routes within OSPF. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route distance is greater than the inter-area route distance.

To change the default administrative distances for inter-area routes, intra-area routes, and external routes, enter the following command.

```
Brocade(config-ospf-router)#distance external 100
Brocade(config-ospf-router)#distance inter-area 90
Brocade(config-ospf-router)#distance intra-area 80
```

Syntax: `[no] distance external | inter-area | intra-area <distance>`

The **external | inter-area | intra-area** parameter specifies the route type for which you are changing the default administrative distance.

The **<distance>** parameter specifies the new distance for the specified route type. Unless you change the distance for one of the route types using commands such as those shown above, the default is 110.

To reset the administrative distance to its system default (110), enter a command such as the following.

```
Brocade(config-ospf-router)#no distance external 100
```

Configuring OSPF group Link State Advertisement (LSA) pacing

The Layer 3 Switch paces LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA refresh timer expires. The accumulated LSAs constitute a group, which the Layer 3 Switch refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the Layer 3 Switch refreshes an accumulated group of LSAs, is configurable to a range from 10 through 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the Layer 3 Switch refreshes the group of accumulated LSAs and sends the group together in the same packets.

Usage guidelines for configuring OSPF group LSA pacing

The pacing interval is inversely proportional to the number of LSAs the Layer 3 Switch is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 to 100 LSAs), increasing the pacing interval to 10 to 20 minutes might enhance performance slightly.

Changing the LSA pacing interval

To change the LSA pacing interval to two minutes (120 seconds), enter the following command.

```
Brocade(config-ospf-router)#timers lsa-group-pacing 120
```

Syntax: [no] **timers lsa-group-pacing** <secs>

The **<secs>** parameter specifies the number of seconds and can be from 10 through 1800 (30 minutes). The default is 240 seconds (4 minutes).

To restore the pacing interval to its default value, enter the following command.

```
Brocade(config-ospf-router)#no timers lsa-group-pacing
```

Modifying OSPF traps generated

OSPF traps as defined by RFC 1850 are supported on Brocade routers. OSPF trap generation is enabled on the router, by default.

When using the CLI, you can disable all or specific OSPF trap generation by entering the following CLI command.

```
Brocade(config-ospf-router)#no snmp-server trap ospf
```


Syntax: `[no] snmp-server trap ospf`

To later re-enable the trap feature, enter **snmp-server trap ospf**.

To disable a specific OSPF trap, enter the command as **no snmp-server trap ospf <ospf-trap>**.

These commands are at the OSPF router level of the CLI.

Here is a summary of OSPF traps supported on Brocade routers, their corresponding CLI commands, and their associated MIB objects from RFC 1850:

- **interface-state-change-trap** – [MIB object: OspfIfStateChange]
- **virtual-interface-state-change-trap** – [MIB object: OspfVirtIfStateChange]
- **neighbor-state-change-trap** – [MIB object: ospfNbrStateChange]
- **virtual-neighbor-state-change-trap** – [MIB object: ospfVirtNbrStateChange]
- **interface-config-error-trap** – [MIB object: ospfIfConfigError]
- **virtual-interface-config-error-trap** – [MIB object: ospfVirtIfConfigError]
- **interface-authentication-failure-trap** – [MIB object: ospfIfAuthFailure]
- **virtual-interface-authentication-failure-trap** – [MIB object: ospfVirtIfAuthFailure]
- **interface-receive-bad-packet-trap** – [MIB object: ospfIfRxBadPacket]
- **virtual-interface-receive-bad-packet-trap** – [MIB object: ospfVirtIfRxBadPacket]
- **interface-retransmit-packet-trap** – [MIB object: ospfTxRetransmit]
- **virtual-interface-retransmit-packet-trap** – [MIB object: ospfVirtIfTxRetransmit]
- **originate-lsa-trap** – [MIB object: ospfOriginateLsa]
- **originate-maxage-lsa-trap** – [MIB object: ospfMaxAgeLsa]
- **link-state-database-overflow-trap** – [MIB object: ospfLsdbOverflow]
- **link-state-database-approaching-overflow-trap** – [MIB object: ospfLsdbApproachingOverflow]

Example

To stop an OSPF trap from being collected, use the CLI command: **no trap <ospf-trap>**, at the OSPF router level of the CLI. To disable reporting of the neighbor-state-change-trap, enter the following command.

```
Brocade(config-ospf-router)#no trap neighbor-state-change-trap
```

Example

To reinstate the trap, enter the following command.

```
Brocade(config-ospf-router)#trap neighbor-state-change-trap
```

Syntax: `[no] trap <ospf-trap>`

Specifying the types of OSPF Syslog messages to log

You can specify which kinds of OSPF-related Syslog messages are logged. By default, the only OSPF messages that are logged are those indicating possible system errors. If you want other kinds of OSPF messages to be logged, you can configure the Brocade device to log them.

NOTE

This feature is not supported on FWS devices.

For example, to specify that all OSPF-related Syslog messages be logged, enter the following commands.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#log all
```

Syntax: [no] **log all** | **adjacency** | **bad_packet** [checksum] | **database** | **memory** | **retransmit**

The **all** option causes all OSPF-related Syslog messages to be logged. If you later disable this option with the **no log all** command, the OSPF logging options return to their default settings.

The **adjacency** option logs essential OSPF neighbor state changes, especially on error cases. This option is disabled by default.

The **bad_packet checksum** option logs all OSPF packets that have checksum errors. This option is enabled by default.

The **bad_packet** option logs all other bad OSPF packets. This option is disabled by default.

The **database** option logs OSPF LSA-related information. This option is disabled by default.

The **memory** option logs abnormal OSPF memory usage. This option is enabled by default.

The **retransmit** option logs OSPF retransmission activities. This option is disabled by default.

Modifying the OSPF standard compliance setting

Brocade routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification.

To configure a router to operate with the latest OSPF standard, RFC 2178, enter the following commands.

```
Brocade(config)#router ospf
Brocade(config-ospf-router)#no rfc1583-compatibility
```

Syntax: [no] **rfc1583-compatibility**

Modifying the exit overflow interval

If a database overflow condition occurs on a router, the router eliminates the condition by removing entries that originated on the router. The exit overflow interval allows you to set how often a Layer 3 Switch checks to see if the overflow condition has been eliminated. The default value is 0. The range is 0 through 86400 seconds (24 hours). If the configured value of the database overflow interval is zero, then the router never leaves the database overflow condition.

NOTE

FastIron devices dynamically allocate OSPF memory as needed. Refer to [“Dynamic OSPF memory”](#) on page 1234.

To modify the exit overflow interval to 60 seconds, enter the following command.

```
Brocade(config-ospf-router)#database-overflow-interval 60
```

Syntax: **database-overflow-interval** <value>

The <value> can be from 0 through 86400 seconds. The default is 0 seconds.

Configuring an OSPF point-to-point link

In an OSPF point-to-point link, a direct Layer 3 connection exists between a single pair of OSPF routers, without the need for Designated and Backup Designated routers. In a point-to-point link, neighboring routers become adjacent whenever they can communicate directly. In contrast, in broadcast and non-broadcast multi-access (NBMA) networks, the Designated Router and the Backup Designated Router become adjacent to all other routers attached to the network.

Configuration notes and limitations for OSPF point-to-point link

- This feature is supported on Gbps Ethernet and 10 Gbps Ethernet interfaces.
- This feature is supported on physical interfaces. It is not supported on virtual interfaces.
- Brocade supports numbered point-to-point networks, meaning the OSPF router must have an IP interface address which uniquely identifies the router over the network. Brocade does not support unnumbered point-to-point networks.

Configuration syntax for OSPF point-to-point link

To configure an OSPF point-to-point link, enter commands such as the following.

```
Brocade(config)#interface eth 1/5
Brocade(config-if-1/5)#ip ospf network point-to-point
```

This command configures an OSPF point-to-point link on Interface 5 in slot 1.

Syntax: [no] ip ospf network point-to-point

Viewing configured OSPF point-to-point links

Refer to [“Displaying OSPF neighbor information”](#) on page 1275 and [“Displaying OSPF interface information”](#) on page 1278.

Configuring OSPF graceful restart

By default, OSPF graceful restart is enabled for the global instance.

For information about how to display OSPF graceful restart information, refer to [“Displaying OSPF graceful restart information”](#) on page 1285.

Enabling and disabling OSPF graceful restart

OSPF graceful restart is enabled by default on a FastIron Layer 3 switch. To disable it, use the following commands.

```
Brocade(config)# router ospf
Brocade(config-ospf-router)# no graceful-restart
```

To re-enable OSPF graceful restart after it has been disabled, enter the following commands.

```
Brocade(config)# router ospf
Brocade(config-ospf-router)# graceful-restart
```

Syntax: [no] graceful-restart

Configuring the OSPF graceful restart time

Use the following commands to specify the maximum amount of time advertised to a neighbor router to maintain routes from and forward traffic to a restarting router.

```
Brocade(config) router ospf
Brocade(config-ospf-router)# graceful-restart restart-time 120
```

Syntax: [no] graceful-restart restart-time <seconds>

The <seconds> variable sets the maximum restart wait time advertised to neighbors.

Possible values are from 10 through 1800 seconds.

The default value is 120 seconds.

Disabling OSPF graceful restart helper mode

By default, a FastIron Layer 3 switch supports other restarting routers as a helper. You can prevent your FastIron router from participating in OSPF graceful restart by using the following commands.

```
Brocade(config) router ospf
Brocade(config-ospf-router)# graceful-restart helper-disable
```

Syntax: [no] graceful-restart helper-disable

This command disables OSPF graceful restart helper mode.

The default behavior is to help the restarting neighbors.

Clearing OSPF information

The following kinds of OSPF information can be cleared from a Brocade OSPF link state database and OSPF routing table:

- Routes received from OSPF neighbors. You can clear routes from all OSPF neighbors, or an individual OSPF neighbor, specified either by the neighbor IP address or its router ID
- OSPF topology information, including all routes in the OSPF routing table
- All routes in the OSPF routing table that were redistributed from other protocols
- OSPF area information, including routes received from OSPF neighbors within an area, as well as routes imported into the area. You can clear area information for all OSPF areas, or for a specified OSPF area

The OSPF information is cleared dynamically when you enter the command; you do not need to remove statements from the Brocade configuration or reload the software for the change to take effect.

Clearing OSPF neighbor information

To clear information on the Brocade device about all OSPF neighbors, enter the following command.

```
Brocade#clear ip ospf neighbor
```

Syntax: clear ip ospf neighbor [ip <ip-addr> | id <ip-addr>]

This command clears all OSPF neighbors and the OSPF routes exchanged with the neighbors in the Brocade OSPF link state database. After this information is cleared, adjacencies with all neighbors are re-established, and routes with these neighbors exchanged again.

To clear information on the Brocade device about OSPF neighbor 10.10.10.1, enter the following command.

```
Brocade#clear ip ospf neighbor ip 10.10.10.1
```

This command clears the OSPF neighbor and the OSPF routes exchanged with neighbor 10.10.10.1 in the OSPF link state database in the Brocade device. After this information is cleared, the adjacency with the neighbor is re-established, and routes are exchanged again.

The neighbor router can be specified either by its IP address or its router ID. To specify the neighbor router using its IP address, use the **ip** *<ip-addr>* parameter. To specify the neighbor router using its router ID, use the **id** *<ip-addr>* parameter.

Clearing OSPF topology information

To clear OSPF topology information on the Brocade device, enter the following command.

```
Brocade#clear ip ospf topology
```

Syntax: clear ip ospf topology

This command clears all OSPF routes from the OSPF routing table, including intra-area, (which includes ABR and ASBR intra-area routes), inter-area, external type 1, external type 2, OSPF default, and OSPF summary routes.

After you enter this command, the OSPF routing table is rebuilt, and valid routes are recomputed from the OSPF link state database. When the OSPF routing table is cleared, OSPF routes in the global routing table are also recalculated. If redistribution is enabled, the routes are imported again.

Clearing redistributed routes from the OSPF routing table

To clear all routes in the OSPF routing table that were redistributed from other protocols, enter the following command.

```
Brocade#clear ip ospf redistribution
```

Syntax: clear ip ospf redistribution

This command clears all routes in the OSPF routing table that are redistributed from other protocols, including direct connected, static, RIP, and BGP. To import redistributed routes from other protocols, use the **redistribution** command at the OSPF configuration level.

Clearing information for OSPF areas

To clear information on the Brocade device about all OSPF areas, enter the following command.

```
Brocade#clear ip ospf
```

This command clears all OSPF areas, all OSPF neighbors, and the entire OSPF routing table. After this information has been cleared, adjacencies with all neighbors are re-established, and all OSPF routes are re-learned.

To clear information on the Brocade device about OSPF area 1, enter the following command.

```
Brocade#clear ip ospf area 1
```

This command clears information about the specified area ID. Information about other OSPF areas is not affected. The command clears information about all OSPF neighbors belonging to the specified area, as well as all routes imported into the specified area. Adjacencies with neighbors belonging to the area are re-established, and routes imported into the area are re-learned.

Syntax: `clear ip ospf [area <area-id>]`

The <area-id> can be specified in decimal format or in IP address format.

Displaying OSPF information

You can use CLI commands and Web management options to display the following OSPF information:

- Trap, area, and interface information – refer to [“Displaying general OSPF configuration information”](#) on page 1272.
- CPU utilization statistics – refer to [“Displaying CPU utilization statistics”](#) on page 1273.
- Area information – refer to [“Displaying OSPF area information”](#) on page 1275.
- Neighbor information – refer to [“Displaying OSPF neighbor information”](#) on page 1275.
- Interface information – refer to [“Displaying OSPF interface information”](#) on page 1278.
- Route information – refer to [“Displaying OSPF route information”](#) on page 1279.
- External link state information – refer to [“Displaying OSPF external link state information”](#) on page 1281.
- Link state information – refer to [“Displaying OSPF link state information”](#) on page 1282.
- Virtual Neighbor information – refer to [“Displaying OSPF virtual neighbor information”](#) on page 1283.
- Virtual Link information – refer to [“Displaying OSPF virtual link information”](#) on page 1284.
- ABR and ASBR information – refer to [“Displaying OSPF ABR and ASBR information”](#) on page 1284.
- Trap state information – refer to [“Displaying OSPF trap status”](#) on page 1284.
- OSPF graceful restart - refer to [“Displaying OSPF graceful restart information”](#) on page 1285.

Displaying general OSPF configuration information

To display general OSPF configuration information, enter the following command at any CLI level.

```
Brocade#show ip ospf config
Router OSPF: Enabled
Redistribution: Disabled
Default OSPF Metric: 10
OSPF Redistribution Metric: Type2

OSPF External LSA Limit: 25000

OSPF Database Overflow Interval: 0
```

RFC 1583 Compatibility: Enabled

Router id: 192.85.2.1

Interface State Change Trap:	Enabled
Virtual Interface State Change Trap:	Enabled
Neighbor State Change Trap:	Enabled
Virtual Neighbor State Change Trap:	Enabled
Interface Configuration Error Trap:	Enabled
Virtual Interface Configuration Error Trap:	Enabled
Interface Authentication Failure Trap:	Enabled
Virtual Interface Authentication Failure Trap:	Enabled
Interface Receive Bad Packet Trap:	Enabled
Virtual Interface Receive Bad Packet Trap:	Enabled
Interface Retransmit Packet Trap:	Enabled
Virtual Interface Retransmit Packet Trap:	Enabled
Originate LSA Trap:	Enabled
Originate MaxAge LSA Trap:	Enabled
Link State Database Overflow Trap:	Enabled
Link State Database Approaching Overflow Trap:	Enabled

OSPF Area currently defined:

Area-ID	Area-Type	Cost
0	normal	0

OSPF Interfaces currently defined:

Ethernet Interface: 3/1-3/2

ip ospf md5-authentication-key-activation-wait-time 300

ip ospf area 0

Ethernet Interface: v1

ip ospf md5-authentication-key-activation-wait-time 300

ip ospf area 0

Ethernet Interface: 2/1

ip ospf auth-change-wait-time 300

ip ospf cost 40

ip ospf area 0

Syntax: show ip ospf config

Displaying CPU utilization statistics

You can display CPU utilization statistics for OSPF and other IP protocols.

To display CPU utilization statistics for OSPF for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI.

Displaying OSPF information

```
Brocade#show process cpu
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ARP              0.01        0.03        0.09        0.22          9
BGP              0.04        0.06        0.08        0.14         13
GVRP             0.00        0.00        0.00        0.00          0
ICMP             0.00        0.00        0.00        0.00          0
IP               0.00        0.00        0.00        0.00          0
OSPF           0.03       0.06       0.09       0.12        11
RIP              0.00        0.00        0.00        0.00          0
STP              0.00        0.00        0.00        0.00          0
VRRP             0.00        0.00        0.00        0.00          0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example.

```
Brocade#show process cpu
The system has only been up for 6 seconds.
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ARP              0.01        0.00        0.00        0.00          0
BGP              0.00        0.00        0.00        0.00          0
GVRP             0.00        0.00        0.00        0.00          0
ICMP             0.01        0.00        0.00        0.00          1
IP               0.00        0.00        0.00        0.00          0
OSPF             0.00        0.00        0.00        0.00          0
RIP              0.00        0.00        0.00        0.00          0
STP              0.00        0.00        0.00        0.00          0
VRRP             0.00        0.00        0.00        0.00          0
```

To display utilization statistics for a specific number of seconds, enter the **show process cpu** command.

```
Brocade#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name    Sec(%)    Time(ms)
ARP              0.00        0
BGP              0.00        0
GVRP             0.00        0
ICMP             0.01        1
IP               0.00        0
OSPF             0.00        0
RIP              0.00        0
STP              0.01        0
VRRP             0.00        0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: **show process cpu** [*<num>*]

The *<num>* parameter specifies the number of seconds and can be from 1 through 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying OSPF area information

To display OSPF area information, enter the **show ip ospf area** command at any CLI level.

```
Brocade#show ip ospf area
Indx Area      Type Cost  SPFR ABR ASBR LSA Chksum(Hex)
1   0.0.0.0    normal 0     1    0   0   1   0000781f
2  192.147.60.0 normal 0     1    0   0   1   0000fee6
3  192.147.80.0 stub   1     1    0   0   2   000181cd
```

Syntax: **show ip ospf area** [*<area-id>*] | [*<num>*]

The *<area-id>* parameter shows information for the specified area.

The *<num>* parameter displays the entry that corresponds to the entry number you enter. The entry number identifies the entry position in the area table.

This display shows the following information.

TABLE 213 CLI display of OSPF area information

Field	Definition
Indx	The row number of the entry in the router OSPF area table.
Area	The area number.
Type	The area type, which can be one of the following: <ul style="list-style-type: none"> nssa normal stub
Cost	The area cost.
SPFR	The SPFR value.
ABR	The ABR number.
ASBR	The ABSR number.
LSA	The LSA number.
Chksum(Hex)	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The Layer 3 Switch uses the checksum to verify that the packet is not corrupted.

Displaying OSPF neighbor information

To display OSPF neighbor information, enter the following command at any CLI level.

```
Brocade#show ip ospf neighbor
Port Address      Pri State      Neigh Address  Neigh ID
8   212.76.7.251    1   full        212.76.7.200   173.35.1.220
```

To display detailed OSPF neighbor information, enter the following command at any CLI level.

Displaying OSPF information

```
Brocade#show ip ospf neighbor detail
```

Port	Address	Pri	State	Neigh Address	Neigh ID	Ev	Op	Cnt
9/1	20.2.0.2	1	FULL/DR	20.2.0.1	2.2.2.2	6	2	0
Second-to-dead:39								
10/1	20.3.0.2	1	FULL/BDR	20.3.0.1	3.3.3.3	5	2	0
Second-to-dead:36								
1/1-1/8	23.5.0.1	1	FULL/DR	23.5.0.2	16.16.16.16	6	2	0
Second-to-dead:33								
2/1-2/2	23.2.0.1	1	FULL/DR	23.2.0.2	15.15.15.15	6	2	0
Second-to-dead:33								

Syntax: `show ip ospf neighbor [router-id <ip-addr>] | [<num>] | [detail]`

The **router-id** <ip-addr> parameter displays only the neighbor entries for the specified router.

The <num> parameter displays only the entry in the specified index position in the neighbor table. For example, if you enter “1”, only the first entry in the table is displayed.

The **detail** parameter displays detailed information about the neighbor routers.

These displays show the following information.

TABLE 214 CLI display of OSPF neighbor information

Field	Description
Port	The port through which the Layer 3 Switch is connected to the neighbor. The port on which an OSPF point-to-point link is configured.
Address	The IP address of this Layer 3 Switch interface with the neighbor.
Pri	The OSPF priority of the neighbor: <ul style="list-style-type: none">• For multi-access networks, the priority is used during election of the Designated Router (DR) and Backup designated Router (BDR).• For point-to-point links, this field shows one of the following values:• 1 = point-to-point link• 3 = point-to-point link with assigned subnet

TABLE 214 CLI display of OSPF neighbor information (Continued)

Field	Description
State	<p>The state of the conversation between the Layer 3 Switch and the neighbor. This field can have one of the following values:</p> <ul style="list-style-type: none"> • Down – The initial state of a neighbor conversation. This value indicates that there has been no recent information received from the neighbor. • Attempt – This state is only valid for neighbors attached to non-broadcast networks. It indicates that no recent information has been received from the neighbor. • Init – A Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor. (The router itself did not appear in the neighbor's Hello packet.) All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface. • 2-Way – Communication between the two routers is bidirectional. This is the most advanced state before beginning adjacency establishment. The Designated Router and Backup Designated Router are selected from the set of neighbors in the 2-Way state or greater. • ExStart – The first step in creating an adjacency between the two neighboring routers. The goal of this step is to decide which router is the master, and to decide upon the initial Database Description (DD) sequence number. Neighbor conversations in this state or greater are called adjacencies. • Exchange – The router is describing its entire link state database by sending Database Description packets to the neighbor. Each Database Description packet has a DD sequence number, and is explicitly acknowledged. Only one Database Description packet can be outstanding at any time. In this state, Link State Request packets can also be sent asking for the neighbor's more recent advertisements. All adjacencies in Exchange state or greater are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading – Link State Request packets are sent to the neighbor asking for the more recent advertisements that have been discovered (but not yet received) in the Exchange state. • Full – The neighboring routers are fully adjacent. These adjacencies will now appear in router links and network link advertisements.
Neigh Address	<p>The IP address of the neighbor:</p> <p>For point-to-point links, the value is as follows:</p> <ul style="list-style-type: none"> • If the Pri field is "1", this value is the IP address of the neighbor router interface. • If the Pri field is "3", this is the subnet IP address of the neighbor router interface.
Neigh ID	The neighbor router ID.
Ev	The number of times the neighbor state changed.
Opt	The sum of the option bits in the Options field of the Hello packet. This information is used by Brocade technical support. Refer to Section A.2 in RFC 2178 for information about the Options field in Hello packets.
Cnt	The number of LSAs that were retransmitted.
Second-to-dead	The amount of time the Brocade device will wait for a HELLO message from each OSPF neighbor before assuming the neighbor is dead.

Displaying OSPF interface information

To display OSPF interface information, enter the **show ip ospf interface** command at any CLI level.

```
Brocade#show ip ospf interface 192.168.1.1
```

```
Ethernet 2/1,OSPF enabled
  IP Address 192.168.1.1, Area 0
  OSPF state ptr2ptr, Pri 1, Cost 1, Options 2, Type pt-2-pt Events 1
  Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
  DR:  Router ID 0.0.0.0          Interface Address 0.0.0.0
  BDR: Router ID 0.0.0.0          Interface Address 0.0.0.0
  Neighbor Count = 0, Adjacent Neighbor Count= 1
  Neighbor: 2.2.2.2
  Authentication-Key:None
  MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

Syntax: **show ip ospf interface** [*<ip-addr>*]

The *<ip-addr>* parameter displays the OSPF interface information for the specified IP address.

The following table defines the highlighted fields shown in the above example output of the **show ip ospf interface** command.

TABLE 215 Output of the **show ip ospf interface** command

Field	Definition
IP Address	The IP address of the interface.
OSPF state	ptr2ptr (point to point)
Pri	The link ID as defined in the router-LSA. This value can be one of the following: <ul style="list-style-type: none"> 1 = point-to-point link 3 = point-to-point link with an assigned subnet
Cost	The configured output cost for the interface.
Options	OSPF Options (Bit7 - Bit0): <ul style="list-style-type: none"> unused:1 opaque:1 summary:1 dont_propagate:1 nssa:1 multicast:1 externals:1 tos:1
Type	The area type, which can be one of the following: <ul style="list-style-type: none"> Broadcast = 0x01 NBMA = 0x02 Point to Point = 0x03 Virtual Link = 0x04 Point to Multipoint = 0x05

TABLE 215 Output of the **show ip ospf interface** command (Continued)

Field	Definition
Events	OSPF Interface Event: <ul style="list-style-type: none"> • Interface_Up = 0x00 • Wait_Timer = 0x01 • Backup_Seen = 0x02 • Neighbor_Change = 0x03 • Loop_Indication = 0x04 • Unloop_Indication = 0x05 • Interface_Down = 0x06 • Interface_Passive = 0x07
Adjacent Neighbor Count	The number of adjacent neighbor routers.
Neighbor:	The neighbor router ID.

Displaying OSPF route information

To display OSPF route information for the router, enter the **show ip ospf routes** command at any CLI level.

```
Brocade#show ip ospf routes
Index Destination      Mask                Path_Cost Type2_Cost Path_Type
1      212.95.7.0           255.255.255.0      1          0          Intra
      Adv_Router      Link_State          Dest_Type State      Tag      Flags
      173.35.1.220      212.95.7.251      Network Valid      00000000 7000
      Paths Out_Port Next_Hop            Type      Arp_Index State
      1      5/6      209.95.7.250      OSPF      8          84 00

Index Destination      Mask                Path_Cost Type2_Cost Path_Type
2      11.3.63.0           255.255.255.0      11         0          Inter
      Adv_Router      Link_State          Dest_Type State      Tag      Flags
      209.95.7.250      11.3.63.0          Network Valid      00000000 0000
      Paths Out_Port Next_Hop            Type      Arp_Index State
      1      5/6      209.95.7.250      OSPF      8          84 00
```

Syntax: **show ip ospf routes** [*<ip-addr>*]

The *<ip-addr>* parameter specifies a destination IP address. If you use this parameter, only the route entries for that destination are shown.

This display shows the following information.

TABLE 216 CLI Display of OSPF route information

Field	Definition
Index	The row number of the entry in the router OSPF route table.
Destination	The IP address of the route's destination.
Mask	The network mask for the route.
Path_Cost	The cost of this route path. (A route can have multiple paths. Each path represents a different exit port for the Layer 3 Switch.)
Type2_Cost	The type 2 cost of this path.

TABLE 216 CLI Display of OSPF route information (Continued)

Field	Definition
Path_Type	The type of path, which can be one of the following: <ul style="list-style-type: none"> • Inter – The path to the destination passes into another area. • Intra – The path to the destination is entirely within the local area. • External1 – The path to the destination is a type 1 external route. • External2 – The path to the destination is a type 2 external route.
Adv_Router	The OSPF router that advertised the route to this Brocade Layer 3 Switch.
Link-State	The link state from which the route was calculated.
Dest_Type	The destination type, which can be one of the following: <ul style="list-style-type: none"> • ABR – Area Border Router • ASBR – Autonomous System Boundary Router • Network – the network
State	The route state, which can be one of the following: <ul style="list-style-type: none"> • Changed • Invalid • Valid This information is used by Brocade technical support.
Tag	The external route tag.
Flags	State information for the route entry. This information is used by Brocade technical support.
Paths	The number of paths to the destination.
Out_Port	The router port through which the Layer 3 Switch reaches the next hop for this route path.
Next_Hop	The IP address of the next-hop router for this path.
Type	The route type, which can be one of the following: <ul style="list-style-type: none"> • OSPF • Static Replaced by OSPF
Arp_Index	The index position in the ARP table of the ARP entry for this path's IP address.
State	State information for the path. This information is used by Brocade technical support.

Displaying the routes that have been redistributed into OSPF

You can display the routes that have been redistributed into OSPF. To display the redistributed routes, enter the **show ip ospf redistribute route** command at any level of the CLI.

```
Brocade#show ip ospf redistribute route
 4.3.0.0 255.255.0.0 static
 3.1.0.0 255.255.0.0 static
10.11.61.0 255.255.255.0 connected
 4.1.0.0 255.255.0.0 static
```

In this example, four routes have been redistributed. Three of the routes were redistributed from static IP routes and one route was redistributed from a directly connected IP route.

Syntax: **show ip ospf redistribute route** [*<ip-addr>* *<ip-mask>*]

The `<ip-addr> <ip-mask>` parameter specifies a network prefix and network mask. Here is an example.

```
Brocade#show ip ospf redistribute route 3.1.0.0 255.255.0.0
3.1.0.0 255.255.0.0 static
```

Displaying OSPF external link state information

To display external link state information, enter the **show ip ospf database external-link-state** command at any CLI level.

```
Brocade#show ip ospf database external-link-state
```

Index	Aging	LS ID	Router	Netmask	Metric	Flag
1	1794	1.168.64.0	192.85.0.3	fffffe000	000003e8	b500 0.0.0.0
2	1794	3.215.0.0	192.85.0.3	fffff0000	000003e8	b500 0.0.0.0
3	1794	1.27.250.0	192.85.0.3	ffffffe00	000003e8	b500 0.0.0.0
4	1794	1.24.23.0	192.85.0.3	ffffff000	000003e8	b500 0.0.0.0
5	1794	1.21.52.0	192.85.0.3	ffffff000	000003e8	b500 0.0.0.0
6	1794	1.18.81.0	192.85.0.3	ffffff000	000003e8	b500 0.0.0.0
7	1794	1.15.110.0	192.85.0.3	ffffff000	000003e8	b500 0.0.0.0
8	1794	1.12.139.0	192.85.0.3	ffffff000	000003e8	b500 0.0.0.0
9	1794	1.9.168.0	192.85.0.3	ffffff000	000003e8	b500 0.0.0.0

Syntax: **show ip ospf database external-link-state** [**advertise** *<num>*] | [**extensive**] | [**link-state-id** *<ip-addr>*] | [**router-id** *<ip-addr>*] | [**sequence-number** *<num(Hex)>*] | [**status** *<num>*]

The **advertise** *<num>* parameter displays the hexadecimal data in the specified LSA packet. The *<num>* parameter identifies the LSA packet by its position in the router External LSA table. To determine an LSA packet position in the table, enter the **show ip ospf external-link-state** command to display the table. Refer to [“Displaying the data in an LSA”](#) on page 1283 for an example.

The **extensive** option displays the LSAs in decrypted format.

NOTE

You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** *<ip-addr>* parameter displays the External LSAs for the LSA source specified by *<IP-addr>*.

The **router-id** *<ip-addr>* parameter shows the External LSAs for the specified OSPF router.

The **sequence-number** *<num(Hex)>* parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

The **status** *<num>* option shows status information.

This OSPF external link state display shows the following information.

TABLE 217 CLI display of OSPF external link state information

Field	Definition
Area ID	The OSPF area the router is in.
Aging	The age of the LSA, in seconds.
LS ID	The ID of the link-state advertisement from which the Layer 3 Switch learned this route.
Router	The router IP address.
Seq(hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps it with a sequence number to enable the Layer 3 Switch and other OSPF routers to determine which LSA for a given route is the most recent.
Chksum	A checksum for the LSA packet, which is based on all the fields in the packet except the age field. The Layer 3 Switch uses the checksum to verify that the packet is not corrupted.
Type	The route type, which is always EXTR (external).

Displaying OSPF link state information

To display link state information, enter the **show ip ospf database link-state** command at any CLI level.

```
Brocade#show ip ospf database link-state
```

Syntax: **show ip ospf database link-state** [**advertise** <num>] | [**asbr**] | [**extensive**] | [**link-state-id** <ip-addr>] | [**network**] | [**nssa**] | [**opaque-area**] | [**router**] | [**router-id** <ip-addr>] | [**sequence-number** <num(Hex)>] | [**status** <num>] | [**summary**]

The **advertise** <num> parameter displays the hexadecimal data in the specified LSA packet. The <num> parameter identifies the LSA packet by its position in the router External LSA table. To determine an LSA packet position in the table, enter the **show ip ospf external-link-state** command to display the table. Refer to “[Displaying the data in an LSA](#)” on page 1283 for an example.

The **asbr** option shows ASBR information.

The **extensive** option displays the LSAs in decrypted format.

NOTE

You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** <ip-addr> parameter displays the External LSAs for the LSA source specified by <IP-addr>.

The **network** option shows network information.

The **nssa** option shows network information.

The **opaque-area** option shows information for opaque areas.

The **router-id** <ip-addr> parameter shows the External LSAs for the specified OSPF router.

The **sequence-number** <num(Hex)> parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

The **status** *<num>* option shows status information.

The **summary** option shows summary information.

Displaying the data in an LSA

You can use the CLI to display the data the Layer 3 Switch received in a specific External LSA packet or other type of LSA packet. For example, to display the LSA data in entry 3 in the External LSA table, enter the following command.

```
Brocade#show ip ospf database external-link-state advertise 3
Index Aging  LS ID          Router          Netmask  Metric   Flag
3      619    1.27.250.0      192.85.0.3     fffffff0 000003e8 b500 0.0.0.0
LSA Header:  age: 619, options: 0x02, seq-nbr: 0x80000003, length: 36
NetworkMask: 255.255.254.0
TOS 0:  metric_type: 1, metric: 1000
        forwarding_address: 0.0.0.0
        external_route_tag: 0
```

Syntax: `show ip ospf database external-link-state [advertise <num>] | [link-state-id <ip-addr>] | [router-id <ip-addr>] | [sequence-number <num(Hex)>] | [status <num>]`

To determine an external LSA or other type of LSA index number, enter one of the following commands to display the appropriate LSA table:

- **show ip ospf database link-state advertise *<num>*** – This command displays the data in the packet for the specified LSA.
- **show ip ospf database external-link-state advertise *<num>*** – This command displays the data in the packet for the specified external LSA.

For example, to determine an external LSA index number, enter the **show ip ospf external-link-state** command.

```
Brocade#show ip ospf external-link-state

Index Aging  LS ID          Router          Netmask  Metric   Flag
1      1809    1.18.81.0      103.103.103.6   fffffff0 000003e8 b500 0.0.0.0
2       8     1.27.250.0     103.103.103.6   fffffff0 000003e8 b500 0.0.0.0
3       8     3.215.0.0      103.103.103.6   ffff0000 000003e8 b500 0.0.0.0
4       18     1.33.192.0     102.102.102.6   fffffc00 000003e8 b500 0.0.0.0
5      959     1.9.168.0      102.102.102.6   fffffff0 00002710 b500 0.0.0.0
6     1807     1.3.226.0      192.85.0.3      fffffff0 000003e8 b500 0.0.0.0
7     1809     1.6.197.0      192.85.3.3      fffffff0 000003e8 b500 0.0.0.0
```

Displaying OSPF virtual neighbor information

To display OSPF virtual neighbor information, enter the **show ip ospf virtual-neighbor** command at any CLI level.

```
Brocade#show ip ospf virtual-neighbor
```

Syntax: `show ip ospf virtual-neighbor [<num>]`

The *<num>* parameter displays the table beginning at the specified entry number.

Displaying OSPF virtual link information

To display OSPF virtual link information, enter the **show ip ospf virtual-link** command at any CLI level.

```
Brocade#show ip ospf virtual-link
```

Syntax: **show ip ospf virtual-link** [*<num>*]

The *<num>* parameter displays the table beginning at the specified entry number.

Displaying OSPF ABR and ASBR information

To display OSPF ABR and ASBR information, enter the **show ip ospf border-routers** command at any CLI level.

```
Brocade#show ip ospf border-routers
```

Syntax: **show ip ospf border-routers** [*<ip-addr>*]

The *<ip-addr>* parameter displays the ABR and ASBR entries for the specified IP address.

Displaying OSPF trap status

All traps are enabled by default when you enable OSPF. To disable or re-enable an OSPF trap, refer to [“Modifying OSPF traps generated”](#) on page 1266.

To display the state of each OSPF trap, enter the **show ip ospf trap** command at any CLI level.

```
Brocade#show ip ospf trap
Interface State Change Trap:           Enabled
Virtual Interface State Change Trap:    Enabled
Neighbor State Change Trap:            Enabled
Virtual Neighbor State Change Trap:     Enabled
Interface Configuration Error Trap:     Enabled
Virtual Interface Configuration Error Trap: Enabled
Interface Authentication Failure Trap:  Enabled
Virtual Interface Authentication Failure Trap: Enabled
Interface Receive Bad Packet Trap:      Enabled
Virtual Interface Receive Bad Packet Trap: Enabled
Interface Retransmit Packet Trap:       Enabled
Virtual Interface Retransmit Packet Trap: Enabled
Originate LSA Trap:                    Enabled
Originate MaxAge LSA Trap:             Enabled
Link State Database Overflow Trap:      Enabled
Link State Database Approaching Overflow Trap: Enabled
```

Syntax: **show ip ospf trap**

Displaying OSPF graceful restart information

To display OSPF graceful restart information for OSPF neighbors, use the **show ip ospf neighbors** command.

```
Brocade#show ip ospf neighbors
Port Address Pri State Neigh Address Neigh ID Ev Opt Cnt
2/7 50.50.50.10 0 FULL/OTHER 50.50.50.1 10.10.10.30 21 66 0
< in graceful restart state, helping 1, timer 60 sec >
```

Syntax: show ip ospf neighbor

Use the following command to display Type 9 grace LSAs on a Brocade Layer 3 switch.

```
Brocade#show ip ospf database grace-link-state
Graceful Link States
Area Interface Adv Rtr Age Seq(Hex) Prd Rsn Nbr Intf IP
0 eth 1/2 2.2.2.2 7 80000001 60 SW 6.1.1.2
```

Syntax: show ip ospf database grace-link-state

[Table 218](#) defines the fields in the show output.

TABLE 218 CLI display of OSPF database grace LSA information

Field	Definition
Area	The OSPF area that the interface configured for OSPF graceful restart is in.
Interface	The interface that is configured for OSPF graceful restart.
Adv Rtr	The ID of the advertised route.
Age	The age of the LSA in seconds.
Seq (Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the FastIron and other OSPF routers to determine the most recent LSA for a given route.
Prd	The grace period. The number of seconds that the neighbor routers should continue to advertise the router as fully adjacent, regardless of the state of database synchronization between the router and its neighbors. Since this time period begins when the grace LSA LS age is equal to 0, the grace period terminates when either the LS age of the grace LSA exceeds the value of a grace period or the grace LSA is flushed.
Rsn	The reason for the graceful restart. Possible values: <ul style="list-style-type: none"> • UK – Unknown • RS – Software restart • UP – Software upgrade or reload • SW – Switch to redundant control processor
Nbr Intf IP	The IP address of the OSPF graceful restart neighbor.

Displaying OSPF information

OSPF version 3 (IPv6)

Table 219 lists the individual Brocade FastIron switches and the Open Shortest Path First (OSPF) version 3 (IPv6) features they support. These features are supported with premium IPv6 devices running the full Layer 3 software image.

TABLE 219 Supported OSPF V3 features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
OSPF V3	Yes	No	Yes	Yes	No
Assigning OSPF V3 areas	Yes	No	Yes	Yes	No
Assigning interfaces to an area	Yes	No	Yes	Yes	No
Virtual links	Yes	No	Yes	Yes	No
Changing the reference bandwidth	Yes	No	Yes	Yes	No
Redistributing routes into OSPF V3	Yes	No	Yes	Yes	No
Filtering OSPF V3 routes	Yes	No	Yes	Yes	No
Configuring default route origination	Yes	No	Yes	Yes	No
Modifying SPF timers	Yes	No	Yes	Yes	No
Modifying administrative distance	Yes	No	Yes	Yes	No
OSPF V3 LSA pacing interval	Yes	No	Yes	Yes	No
Modifying the exit overflow interval	Yes	No	Yes	Yes	No
Modifying the external link state database limit	Yes	No	Yes	Yes	No
Modifying OSPF V3 interface defaults	Yes	No	Yes	Yes	No
Event logging	Yes	No	Yes	Yes	No
IPSec for OSPFv3	Yes	No	Yes	Yes	No

OSPF (IPv6) overview

Open Shortest Path First (OSPF) is a link-state routing protocol. OSPF uses link-state advertisements (LSAs) to update neighboring routers about its interfaces and information on those interfaces. The switch floods LSAs to all neighboring routers to update them about the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router:

- The differences between OSPF Version 2 (OSPF V2) and OSPF Version 3 (OSPF V3).
- The link state advertisement types for OSPF Version 3.

- How to configure OSPF Version 3.
- How to display OSPF Version 3 information and statistics.

NOTE

The terms *Layer 3 Switch* and *router* are used interchangeably in this chapter and mean the same thing.

Differences between OSPF V2 and OSPF V3

IPv6 supports OSPF V3 functions similarly to OSPF V2 (the current version that IPv4 supports), except for the following enhancements:

- Support for IPv6 addresses and prefixes.
- While OSPF V2 runs per IP subnet, OSPF V3 runs per link. In general, you can configure several IPv6 addresses on a router interface, but OSPF V3 forms one adjacency per interface only, using the interface associated link-local address as the source for OSPF protocol packets. On virtual links, OSPF V3 uses the global IP address as the source.
- You can run one instance of OSPF Version 2 and one instance of OSPF V3 concurrently on a link.
- Support for IPv6 link state advertisements (LSAs).

In addition, Brocade implements some new commands that are specific to OSPF V3. This chapter describes the commands that are specific to OSPF V3.

NOTE

Although OSPF Versions 2 and 3 function similarly to each other, Brocade has implemented the user interface for each version independently of each other. Therefore, any configuration of OSPF Version 2 features will not affect the configuration of OSPF V3 features and vice versa.

NOTE

You are required to configure a router ID when running only IPv6 routing protocols.

Link state advertisement types for OSPF V3

OSPF V3 supports the following types of LSAs:

- Router LSAs (Type 1)
- Network LSAs (Type 2)
- Interarea-prefix LSAs for ABRs (Type 3)
- Interarea-router LSAs for ASBRs (Type 4)
- Autonomous system external LSAs (Type 5)
- Link LSAs (Type 8)
- Intra-area prefix LSAs (Type 9)

For more information about these LSAs, see RFC 2740.

OSPF V3 configuration

To configure OSPF V3, you must perform the following tasks:

1. [“Enabling OSPF V3”](#) on page 1289
2. [“Assigning OSPF V3 areas”](#) on page 1290
3. [“Assigning interfaces to an area”](#) on page 1291

The following configuration tasks are optional:

- Configure a virtual link between an ABR without a physical connection to a backbone area and the Brocade device in the same area with a physical connection to the backbone area.
- Change the reference bandwidth for the cost on OSPF V3 interfaces.
- Configure the redistribution of routes into OSPF V3.
- Configure default route origination.
- Modify the shortest path first (SPF) timers.
- Modify the administrative distances for OSPF V3 routes.
- Configure the OSPF V3 LSA pacing interval
- Modify how often the Brocade device checks on the elimination of the database overflow condition.
- Modify the external link state database limit.
- Modify the default values of OSPF V3 parameters for router interfaces.
- Disable or re-enable OSPF V3 event logging.

Enabling OSPF V3

Before enabling the Brocade device to run OSPF V3, you must do the following:

- Enable the forwarding of IPv6 traffic on the Brocade device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface over which you plan to enable OSPF V3. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

By default, OSPF V3 is disabled. To enable OSPF V3, you must enable it globally.

To enable OSPF V3 globally, enter the **ipv6 router ospf** command.

```
Brocade(config-ospf-router)#ipv6 router ospf
Brocade(config-ospf6-router)#
```

After you enter this command, the Brocade device enters the IPv6 OSPF configuration level, where you can access several commands that allow you to configure OSPF V3.

Syntax: **[no] ipv6 router ospf**

To disable OSPF V3, enter the **no** form of this command. If you disable OSPF V3, the Brocade device removes all the configuration information for the disabled protocol from the running-config. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

The CLI displays a warning message such as the following.

```
Brocade(config-ospf6-router)#no ipv6 router ospf
ipv6 router ospf mode now disabled. All ospf config data will be lost when writing
to flash!
```

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (for example, **ipv6 router ospf**). If you have already saved the configuration to the startup-config file and reloaded the software, the information is gone. If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup-config file containing the protocol configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

Assigning OSPF V3 areas

After OSPF V3 is enabled, you can assign OSPF V3 areas. You can assign an IPv4 address or a number as the **area ID** for each area. The area ID is representative of all IPv6 addresses (subnets) on a router interface. Each router interface can support one area.

An area can be **normal** or a **stub**:

- **Normal** – OSPF routers within a normal area can send and receive External Link State Advertisements (LSAs).
- **Stub** – OSPF routers within a stub area cannot send or receive External LSAs. In addition, OSPF routers in a stub area must use a default route to the area Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.

For example, to set up OSPF V3 areas 0.0.0.0, 200.5.0.0, 192.5.1.0, and 195.5.0.0, enter the following commands.

```
Brocade(config-ospf6-router)#area 0.0.0.0
Brocade(config-ospf6-router)#area 200.5.0.0
Brocade(config-ospf6-router)#area 192.5.1.0
Brocade(config-ospf6-router)#area 195.5.0.0
```

Syntax: [no] **area** <number> | <ipv4-address>

The <number> | <ipv4-address> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 – 18.

NOTE

You can assign one area on a router interface.

Assigning a totally stubby area

By default, the Brocade device sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of LSAs sent into a stub area by configuring the Brocade device to stop sending summary LSAs into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs into a stub area, but the Brocade device still accepts summary LSAs from OSPF neighbors and floods them to other areas. The Brocade device can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the router flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE

This feature applies only when the Brocade device is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

For example, to disable summary LSAs for stub area 40 and specify an additional metric of 99, enter the following command.

```
Brocade(config-ospf6-router)#area 40 stub 99 no-summary
```

Syntax: **area** <number> | <ipv4-address> **stub** <metric> [**no-summary**]

The <number> | <ipv4-address> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 – 18.

The **stub** <metric> parameter specifies an additional cost for using a route to or from this area and can be from 1 – 16777215. There is no default. Normal areas do not use the cost parameter.

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area.

Assigning interfaces to an area

After you define OSPF V3 areas, you must assign router interfaces to the areas. All router interfaces must be assigned to one of the defined areas on an OSPF router. When an interface is assigned to an area, all corresponding subnets on that interface are automatically included in the assignment.

For example, to assign Ethernet interface 3/1 to area 192.5.0.0, enter the following commands.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e100-3/1)#ipv6 ospf area 192.5.0.0
```

Syntax: [**no**] **ipv6 ospf area** <number> | <ipv4-address>

The <number> | <ipv4-address> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 – 18.

To remove the interface from the specified area, use the **no** form of this command.

Configuring virtual links

All ABRs must have either a direct or indirect link to an OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to a backbone area, you can configure a virtual link from the ABR to another router within the same area that has a physical connection to the backbone area.

The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection) and the ABR requiring a logical connection to the backbone.

Two parameters must be defined for all virtual links—transit area ID and neighbor router:

- The transit area ID represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- When assigned from the router interface requiring a logical connection, the neighbor router field is the router ID (IPv4 address) of the router that is physically connected to the backbone. When assigned from the router interface with the physical connection, the neighbor router is the router ID (IPv4) address of the router requiring a logical connection to the backbone.

NOTE

By default, the Brocade router ID is the IPv4 address configured on the lowest numbered loopback interface. If the Brocade device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.

NOTE

When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

For example, imagine that ABR1 in areas 1 and 2 is cut off from the backbone area (area 0). To provide backbone access to ABR1, you can add a virtual link between ABR1 and ABR2 in area 1 using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on ABR1, enter the following command on ABR1.

```
Brocade(config-ospf6-router)#area 1 virtual-link 209.157.22.1
```

To define the virtual link on ABR2, enter the following command on ABR2.

```
Brocade(config-ospf6-router)#area 1 virtual-link 10.0.0.1
```

Syntax: **area** <number> | <ipv4-address> **virtual-link** <router-id>

The **area** <number> | <ipv4-address> parameter specifies the transit area.

The <router-id> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a router, enter the **show ip** command.

Assigning a virtual link source address

When routers at both ends of a virtual link need to communicate with one another, the source address included in the packets must be a global IPv6 address. Therefore, you must determine the global IPv6 address to be used by the routers for communication across the virtual link. You can specify that a router uses the IPv6 global address assigned to one of its interfaces.

For example, to specify the global IPv6 address assigned to Ethernet interface 3/1 on ABR1 as the source address for the virtual link on ABR1, enter the following command on ABR1.

```
Brocade(config-ospf6-router)#virtual-link-if-address interface ethernet 3/1
```

To specify the global IPv6 address assigned to tunnel interface 1 on ABR2 as the source address for the virtual link on ABR2, enter the following command on ABR2.

```
Brocade(config-ospf6-router)#virtual-link-if-address interface tunnel 1
```

Syntax: **virtual-link-if-address interface ethernet** <port> | **loopback** <number> | **tunnel** <number> | **ve** <number>

The **ethernet** | **loopback** | **tunnel** | **ve** parameter specifies the interface from which the router derives the source IPv6 address for communication across the virtual link. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the respective interface.

To delete the source address for the virtual link, use the **no** form of this command.

Modifying virtual link parameters

You can modify the following virtual link parameters:

- **Dead-interval:** The number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router is down. The range is 1 – 65535 seconds. The default is 40 seconds.
- **Hello-interval:** The length of time between the transmission of hello packets. The range is 1 – 65535 seconds. The default is 10 seconds.
- **Retransmit-interval:** The interval between the re-transmission of link state advertisements to router adjacencies for this interface. The range is 0 – 3600 seconds. The default is 5 seconds.
- **Transmit-delay:** The period of time it takes to transmit Link State Update packets on the interface. The range is 0 – 3600 seconds. The default is 1 second.

NOTE

The values of the **dead-interval** and **hello-interval** parameters must be the same at both ends of a virtual link. Therefore, if you modify the values of these parameters at one end of a virtual link, you must remember to make the same modifications on the other end of the link.

The values of the other virtual link parameters do not require synchronization.

For example, to change the dead interval to 60 seconds on the virtual links defined on ABR1 and ABR2, enter the following command on ABR1.

```
Brocade(config-ospf6-router)#area 1 virtual-link 209.157.22.1
dead-interval 60
```

Enter the following command on ABR2.

```
Brocade(config-ospf6-router)#area 1 virtual-link 10.0.0.1 dead-interval 60
```

Syntax: **area** <number> | <ipv4-address> **virtual-link** <router-id> [**dead-interval** <seconds> | **hello-interval** <seconds> | **retransmit-interval** <seconds> | **transmit-delay** <seconds>]

The **area** <number> | <ipv4-address> parameter specifies the transit area.

The <router-id> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a router, enter the **show ip** command.

The **dead-interval**, **hello-interval**, **retransmit-interval**, and **transmit-delay** parameters are discussed earlier in this section.

Changing the reference bandwidth for the cost on OSPF V3 interfaces

Each interface on which OSPF V3 is enabled has a cost associated with it. The Brocade device advertises its interfaces and their costs to OSPF V3 neighbors. For example, if an interface has an OSPF cost of ten, the Brocade device advertises the interface with a cost of ten to other OSPF routers.

By default, an interface OSPF cost is based on the port speed of the interface. The software uses the following formula to calculate the cost.

Cost = reference-bandwidth/interface-speed

By default, the reference bandwidth is 100 Mbps. If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port cost = $100/10 = 10$
- 100 Mbps port cost = $100/100 = 1$
- 1000 Mbps port cost = $100/1000 = 0.10$, which is rounded up to 1
- 155 Mbps port cost = $100/155 = 0.65$, which is rounded up to 1
- 622 Mbps port cost = $100/622 = 0.16$, which is rounded up to 1
- 2488 Mbps port cost = $100/2488 = 0.04$, which is rounded up to 1

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- **Trunk group** – The combined bandwidth of all the ports.
- **Virtual (Ethernet) interface** – The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

You can change the default reference bandwidth from 100 Mbps to a value from 1 – 4294967 Mbps.

If a change to the reference bandwidth results in a cost change to an interface, the Brocade device sends a link state update to update the costs of interfaces advertised by the Brocade device.

NOTE

If you specify the cost for an individual interface, the cost you specify overrides the cost calculated by the software.

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 0.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.

For example, to change the reference bandwidth to 500, enter the following command.

```
Brocade(config-ospf6-router)#auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port cost = $500/10 = 50$
- 100 Mbps port cost = $500/100 = 5$
- 1000 Mbps port cost = $500/1000 = 0.5$, which is rounded up to 1

- 155 Mbps port cost = $500/155 = 3.23$, which is rounded up to 4
- 622 Mbps port cost = $500/622 = 0.80$, which is rounded up to 1
- 2488 Mbps port cost = $500/2488 = 0.20$, which is rounded up to 1

The costs for 10 Mbps, 100 Mbps, and 155 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: `[no] auto-cost reference-bandwidth <number>`

The `<number>` parameter specifies the reference bandwidth and can be a value from 1 – 4294967. The default is 100.

To restore the reference bandwidth to its default value and thus restore the default costs of interfaces to their default values, enter the **no** form of this command.

Redistributing routes into OSPF V3

In addition to specifying which routes are redistributed into OSPF V3, you can configure the following aspects related to route redistribution:

- Default metric
- Metric type
- Advertisement of an external aggregate route

Configuring route redistribution into OSPF V3

You can configure the Brocade device to redistribute routes from the following sources into OSPF V3:

- IPv6 static routes
- Directly connected IPv6 networks
- RIPng

You can redistribute routes in the following ways:

- By route types, for example, the Brocade device redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the Brocade device redistributes specified IPv6 static and RIPng routes only.

For example, to configure the redistribution of all IPv6 static RIPng routes, enter the following commands.

```
Brocade(config-ospf6-router)#redistribute static
Brocade(config-ospf6-router)#redistribute rip
```

Syntax: `no] redistribute bgp | connected | rip | static [metric <number> | metric-type <type>]`

The **connected** | **rip** | **static** keywords specify the route source.

The **metric <number>** parameter specifies the metric used for the redistributed route. If a value is not specified for this option, and the value for the **default-metric** command is set to 0, its default metric, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed. For information about the **default-metric** command, refer to [“Modifying default metric for routes redistributed into OSPF version 3”](#) on page 1297

The **metric-type** <type> parameter specifies an OSPF metric type for the redistributed route. You can specify external type 1 or external type 2. If a value is not specified for this option, the Brocade device uses the value specified by the **metric-type** command. For information about modifying the default metric type using the **metric-type** command, refer to [“Modifying default metric for routes redistributed into OSPF version 3”](#) on page 1297

For example, to configure a route map and use it for redistribution of routes into OSPF V3, enter commands such as the following.

```
Brocade(config)#ipv6 route 2001:1::/32 4823:eoff:343e::23
Brocade(config)#ipv6 route 2001:2::/32 4823:eoff:343e::23
Brocade(config)#ipv6 route 2001:3::/32 4823:eoff:343e::23 metric 5
Brocade(config)#route-map abc permit 1
Brocade(config-route-map abc)#match metric 5
Brocade(config-route-map abc)#set metric 8
Brocade(config-route-map abc)#ipv6 router ospf
Brocade(config-ospf6-router)#redistribute static route-map abc
```

The commands in this example configure some static IPv6 routes and a route map, and use the route map for redistributing the static IPv6 routes into OSPF V3.

The **ipv6 route** commands configure the static IPv6 routes.

The **route-map** command begins configuration of a route map called “abc”. The number indicates the route map entry (called the “instance”) you are configuring. A route map can contain multiple entries. The software compares packets to the route map entries in ascending numerical order and stops the comparison once a match is found.

NOTE

The default action rule for route-map is to deny all routes that are not explicitly permitted. Refer to [“Configuring an OSPF V3 distribution list using a route map that uses a prefix list”](#) on page 1301.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribute** command configures the redistribution of static IPv6 routes into OSPF V3, and uses route map “abc” to control the routes that are redistributed. In this example, the route map allows a static IPv6 route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route redistribution table.

Syntax: [no] redistribute bgp | connected | isis | rip | static [route-map <map-name>]

The **bgp** | **connected** | **isis** | **rip** | **static** keywords specify the route source.

The **route-map** <map-name> parameter specifies the route map name. The following match parameters are valid for OSPF V3 redistribution:

- **match metric** <number>

The following set parameters are valid for OSPF redistribution:

- **set metric** [+ | -] <number> | none
- **set metric-type** type-1 | type-2

NOTE

You must configure the route map before you configure a redistribution filter that uses the route map.

NOTE

When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

NOTE

For an external route that is redistributed into OSPF V3 through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map or the **default-metric <num>** command. For a route redistributed without using a route map, the metric is set by the metric parameter if set or the **default-metric <num>** command if the metric parameter is not set.

Modifying default metric for routes redistributed into OSPF version 3

The default metric is a global parameter that specifies the cost applied by default to routes redistributed into OSPF V3. The default value is 0.

If the **metric** parameter for the **redistribute** command is not set and the **default-metric** command is set to 0, its default value, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed. For information about the **redistribute** command, refer to [“Configuring route redistribution into OSPF V3”](#) on page 1295.

NOTE

You also can define the cost on individual interfaces. The interface cost overrides the default cost. For information about defining the cost on individual interfaces, refer to [“Modifying OSPF V3 interface defaults”](#) on page 1306 and [“Changing the reference bandwidth for the cost on OSPF V3 interfaces”](#) on page 1294.

To assign a default metric of 4 to all routes imported into OSPF V3, enter the **default-metric** command.

```
Brocade(config-ospf6-router)#default-metric 4
```

Syntax: **no** | **default-metric** <number>

You can specify a value from 0 – 65535. The default is 0.

To restore the default metric to the default value, use the **no** form of this command.

Modifying metric type for routes redistributed into OSPF version 3

The Brocade device uses the **metric-type** parameter by default for all routes redistributed into OSPF V3 unless you specify a different metric type for individual routes using the **redistribute** command. (For more information about using the **redistribute** command, refer to [“Redistributing routes into OSPF V3”](#) on page 1295).

A type 1 route specifies a small metric (two bytes), while a type 2 route specifies a big metric (three bytes). The default value is type 2.

To modify the default value of type 2 to type 1, enter the **metric-type** command.

```
Brocade(config-ospf6-router)#metric-type type1
```

Syntax: **no** | **metric-type** type1 | type2

To restore the metric type to the default value, use the **no** form of this command.

External route summarization

When the Brocade device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified IPv6 address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the Brocade device, no action is taken if the device has already advertised the aggregate route; otherwise, the device advertises the aggregate route. If an imported route that falls within a configured address range is removed by the device, no action is taken if there are other imported routes that fall within the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The Brocade device sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external link state database overflow (LSDB) condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the Brocade device exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE

If you use redistribution filters in addition to address ranges, the Brocade device applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE

If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

NOTE

This option affects only imported, type 5 external routes. A single type 5 LSA is generated and flooded throughout the AS for multiple external routes.

Configuring external route summarization

To configure the summary address 2201::/24 for routes redistributed into OSPF V3, enter the following command.

```
Brocade(config-ospf6-router)#summary-address 2201::/24
```

In this example, the summary prefix 2201::/24 includes addresses 2201::/1 through 2201::/24. Only the address FEC0::/24 is advertised in an external link-state advertisement.

Syntax: `summary-address <ipv6-prefix>/<prefix-length>`

You must specify the `<ipv6-prefix>` parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the `<prefix-length>` parameter as a decimal value. A slash mark (/) must follow the `<ipv6-prefix>` parameter and precede the `<prefix-length>` parameter.

Filtering OSPF V3 routes

You can filter the routes to be placed in the OSPF V3 route table by configuring distribution lists. OSPF V3 distribution lists can be applied globally or to an interface.

The functionality of OSPF V3 distribution lists is similar to that of OSPFv2 distribution lists. However, unlike OSPFv2 distribution lists, which filter routes based on criteria specified in an Access Control List (ACL), OSPF V3 distribution lists can filter routes using information specified in an IPv6 prefix list or a route map.

Configuration examples for filtering OSPF v3 routes

The following sections show examples of filtering OSPF V3 routes using prefix lists globally and for a specific interface, as well as filtering OSPF V3 routes using a route map.

You can configure the device to use all three types of filtering. When you do this, filtering using route maps has higher priority over filtering using global prefix lists. Filtering using prefix lists for a specific interface has lower priority than the other two filtering methods.

The example in this section assume the following routes are in the OSPF V3 route table.

```
Brocade#show ipv6 ospf route
```

```
Current Route count: 5
  Intra: 3 Inter: 0 External: 2 (Type1 0/Type2 2)
  Equal-cost multi-path: 0
  Destination          Options   Area          Cost Type2 Cost
  Next Hop Router      Outgoing Interface
*IA 3001::/64          -----  0.0.0.1        0  0
  ::                   ve 10
*E2 3010::/64          -----  0.0.0.0        10 0
  fe80::2e0:52ff:fe00:10 ve 10
*IA 3015::/64          V6E---R-- 0.0.0.0        11 0
  fe80::2e0:52ff:fe00:10 ve 10
*IA 3020::/64          -----  0.0.0.0        10 0
  ::                   ve 11
*E2 6001:5000::/64     -----  0.0.0.0        10 0
  fe80::2e0:52ff:fe00:10 ve 10
```

Configuring an OSPF V3 distribution list using an IPv6 prefix list as input

The following example illustrates how to use an IPv6 prefix list is used to filter OSPF V3 routes.

To specify an IPv6 prefix list called filterOspfRoutes that denies route 3010::/64, enter the following commands.

```
Brocade(config)#ipv6 prefix-list filterOspfRoutes seq 5 deny 3010::/64
Brocade(config)#ipv6 prefix-list filterOspfRoutes seq 7 permit ::/0 ge 1 le 128
```

Syntax: `ipv6 prefix-list <name> [seq <seq-value>] [description <string>] deny | permit <ipv6-addr>/<mask-bits> [ge <ge-value>] [le <le-value>]`

To configure a distribution list that applies the filterOspfRoutes prefix list globally.

```
Brocade(config)#ipv6 router ospf
Brocade(config-ospf6-router)#distribute-list prefix-list filterOspfRoutes in
```

Syntax: `[no] distribute-list prefix-list <name> in [<interface>]`

OSPF V3 configuration

After this distribution list is configured, route 3010::/64 would be omitted from the OSPF V3 route table.

```
Brocade#show ipv6 ospf route
```

```
Current Route count: 4
  Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
  Equal-cost multi-path: 0
  Destination          Options   Area          Cost Type2 Cost
  Next Hop Router      Outgoing Interface
*IA 3001::/64          -----  0.0.0.1         0  0
   ::                  ve 10
*IA 3015::/64          V6E---R-- 0.0.0.0        11  0
   fe80::2e0:52ff:fe00:10 ve 10
*IA 3020::/64          -----  0.0.0.0         10  0
   ::                  ve 11
*E2 6001:5000::/64     -----  0.0.0.0         10  0
   fe80::2e0:52ff:fe00:10 ve 10
```

The following commands specify an IPv6 prefix list called filterOspfRoutesVe that denies route 3015::/64.

```
Brocade(config)#ipv6 prefix-list filterOspfRoutesVe seq 5 deny 3015::/64
Brocade(config)#ipv6 prefix-list filterOspfRoutesVe seq 10 permit ::/0 ge 1 le 128
```

The following commands configure a distribution list that applies the filterOspfRoutesVe prefix list to routes pointing to virtual interface 10.

```
Brocade(config)#ipv6 router ospf
Brocade(config-ospf6-router)#distribute-list prefix-list filterOspfRoutes in ve
10
```

After this distribution list is configured, route 3015::/64, pointing to virtual interface 10, would be omitted from the OSPF V3 route table.

```
Brocade#show ipv6 ospf route
```

```
Current Route count: 4
  Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
  Equal-cost multi-path: 0
  Destination          Options   Area          Cost Type2 Cost
  Next Hop Router      Outgoing Interface
*IA 3001::/64          -----  0.0.0.1         0  0
   ::                  ve 10
*E2 3010::/64          -----  0.0.0.0         10  0
   fe80::2e0:52ff:fe00:10 ve 10
*IA 3020::/64          -----  0.0.0.0         10  0
   ::                  ve 11
*E2 6001:5000::/64     -----  0.0.0.0         10  0
   fe80::2e0:52ff:fe00:10 ve 10
```

Configuring an OSPF V3 distribution list using a route map as input

The following commands configure a route map that matches internal routes.

```
Brocade(config)#route-map allowInternalRoutes permit 10
Brocade(config-route-map allowInternalRoutes)#match route-type internal
```

Refer to “Policy-Based Routing” for information on configuring route maps.

The following commands configure a distribution list that applies the allowInternalRoutes route map globally to OSPF V3 routes.

```
Brocade(config)#ipv6 router ospf
Brocade(config-ospf6-router)#distribute-list route-map allowinternalroutes in
```

Syntax: [no] distribute-list route-map <name> in

After this distribution list is configured, the internal routes would be included, and the external routes would be omitted from the OSPF V3 route table.

```
Brocade#show ipv6 ospf route
```

```
Current Route count: 3
  Intra: 3 Inter: 0 External: 0 (Type1 0/Type2 0)
  Equal-cost multi-path: 0
  Destination          Options   Area          Cost Type2 Cost
  Next Hop Router      Outgoing Interface
-----
*IA 3001::/64          -----  0.0.0.1          0  0
  ::                   ve 10
*IA 3015::/64          V6E---R--  0.0.0.0          11 0
  fe80::2e0:52ff:fe00:10 ve 10
*IA 3020::/64          -----  0.0.0.0          10 0
  ::                   ve 11
```

Configuring an OSPF V3 distribution list using a route map that uses a prefix list

When you configure route redistribution into OSPF V3 using a route map that uses a prefix list, the device supports both **permit** and **deny** statements in the route map and **permit** statements only in the prefix list. Therefore, the action to permit or deny is determined by the route map, and the conditions for the action are contained in the prefix list. The following shows an example configuration.

```
Brocade(config)#route-map v64 deny 10
Brocade(config-route-map v64)#match ipv6 next-hop prefix-list ospf-filter5
Brocade(config-route-map v64)#route-map v64 deny 11
Brocade(config-route-map v64)#match ipv6 address prefix-list ospf-filter2
Brocade(config-route-map v64)#route-map v64 permit 12
Brocade(config-route-map v64)#exit
Brocade(config)#ipv6 prefix-list ospf-filter2 seq 15 permit 2001:aa:2001:102::/64
ge 65 le 96
Brocade(config)#ipv6 prefix-list ospf-filter5 seq 15 permit
fe80::2e0:52ff:fe00:100/128
```

In this example the prefix lists, **ospf-filter2** and **ospf-filter5**, contain a range of IPv6 routes and one host route to be denied, and the route map **v64** defines the deny action.

NOTE

The default action rule for **route-map** is to deny all routes that are not explicitly permitted. If you configure a “deny” route map but want to permit other routes that do not match the rule, configure an “empty” permit route map. For example.

```
Brocade(config)#route-map abc deny 10
Brocade(config-route-map abc)#match metric 20
Brocade(config-route-map abc)#route-map abc permit 20
```

Without the last line in the above example, all routes would be denied.

Default route origination

When the Brocade device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPF V3 routing domain. This feature is called “default route origination” or “default information origination.”

By default, the Brocade device does not advertise the default route into the OSPF V3 domain. If you want the device to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the device advertises a type 5 default route that is flooded throughout the AS (except stub areas).

The device advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor.

NOTE

The Brocade device does not advertise the OSPF default route, regardless of other configuration parameters, unless you explicitly enable default route origination.

If default route origination is enabled and you disable it, the default route originated by the device is flushed. Default routes generated by other OSPF routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

Configuring a default route origination

To create and advertise a default route with a metric of 2 and as a type 1 external route, enter the following command.

```
Brocade(config-ospf6-router)#default-information-originate always metric 2
metric-type type1
```

Syntax: [no] **default-information-originate** [always] [metric <value>] [metric-type <type>]

The **always** keyword originates a default route regardless of whether the device has learned a default route. This option is disabled by default.

The **metric <value>** parameter specifies a metric for the default route. If this option is not used, the value of the **default-metric** command is used for the route. For information about this command, refer to [“Modifying default metric for routes redistributed into OSPF version 3”](#) on page 1297

The **metric-type** <type> parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The <type> can be one of the following:

- 1 – Type 1 external route
- 2 – Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE

If you specify a metric and metric type, the values you specify are used even if you do not use the always option.

To disable default route origination, enter the **no** form of the command.

Shortest path first timers

The Brocade device uses the following timers when calculating the shortest path for OSPF V3 routes:

- **SPF delay** – When the Brocade device receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits 5 seconds. You can configure the SPF delay to a value from 0 – 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- **SPF hold time** – The Brocade device waits a specific amount of time between consecutive SPF calculations. By default, the device waits 10 seconds. You can configure the SPF hold time to a value from 0 – 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the SPF delay and hold time to lower values to cause the device to change to alternate paths more quickly if a route fails. Note that lower values for these parameters require more CPU processing time.

You can change one or both of the timers.

NOTE

If you want to change only one of the timers, for example, the SPF delay timer, you must specify the new value for this timer as well as the current value of the SPF hold timer, which you want to retain. The Brocade device does not accept only one timer value.

Modifying shortest path first timers

To change the SPF delay to 10 seconds and the SPF hold to 20 seconds, enter the following command.

```
Brocade(config-ospf6-router)#timers spf 10 20
```

Syntax: **timers spf** <delay> <hold-time>

For the <delay> and <hold-time> parameters, specify a value from 0 – 65535 seconds.

To set the timers back to their default values, enter the **no** version of this command.

Administrative distance

The Brocade device can learn about networks from various protocols, including IPv6, RIPng, and OSPF V3. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. By default, the administrative distance for OSPF V3 routes is 110.

The device selects one route over another based on the source of the route information. To do so, the device can use the administrative distances assigned to the sources. You can influence the device decision by changing the default administrative distance for OSPF V3 routes.

Configuring administrative distance based on route type

You can configure a unique administrative distance for each type of OSPF V3 route. For example, you can use this feature to influence the Brocade device to prefer a static route over an OSPF inter-area route and to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the device has multiple routes to the same network from different protocols. The device prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following OSPF V3 route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all of these OSPF V3 route types is 110.

NOTE

This feature does not influence the choice of routes within OSPF V3. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route distance is greater than the inter-area route distance.

For example, to change the default administrative distances for intra-area routes to 80, inter-area routes to 90, and external routes to 100, enter the following commands.

```
Brocade(config-ospf6-router)#distance intra-area 80
Brocade(config-ospf6-router)#distance inter-area 90
Brocade(config-ospf6-router)#distance external 100
```

Syntax: `distance external | inter-area | intra-area <distance>`

The **external | inter-area | intra-area** keywords specify the route type for which you are changing the default administrative distance.

The **<distance>** parameter specifies the new distance for the specified route type. You can specify a value from 1 – 255.

To reset the administrative distance of a route type to its system default, enter the **no** form of this command.

Configuring the OSPF V3 LSA pacing interval

The Brocade device paces OSPF V3 LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA refresh timer expires. The accumulated LSAs constitute a group, which the Brocade device refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the Brocade device refreshes an accumulated group of LSAs, is configurable to a range from 10 – 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the Brocade device refreshes the group of accumulated LSAs and sends the group together in the same packets.

The pacing interval is inversely proportional to the number of LSAs the Brocade device is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 – 100 LSAs), increasing the pacing interval to 10 – 20 minutes might enhance performance only slightly.

To change the OSPF V3 LSA pacing interval to two minutes (120 seconds), enter the following command.

```
Brocade(config)#ipv6 router ospf
Brocade(config-ospf6-router)#timers lsa-group-pacing 120
```

Syntax: [no] **timers lsa-group-pacing** <seconds>

The <seconds> parameter specifies the number of seconds and can be from 10 – 1800 (30 minutes). The default is 240 seconds (four minutes).

To restore the pacing interval to its default value, use the **no** form of the command.

Modifying exit overflow interval

If a database overflow condition occurs on the Brocade device, the device eliminates the condition by removing entries that originated on the device. The exit overflow interval allows you to set how often a device checks to see if the overflow condition has been eliminated. The default value is 0. If the configured value of the database overflow interval is 0, then the device never leaves the database overflow condition.

For example, to modify the exit overflow interval to 60 seconds, enter the following command.

```
Brocade(config-ospf6-router)#database-overflow-interval 60
```

Syntax: [no] **auto-cost reference-bandwidth** <number>

The <seconds> parameter can be a value from 0 – 86400 seconds (24 hours).

To reset the exit overflow interval to its system default, enter the **no** form of this command.

Modifying external link state database limit

By default, the link state database can hold a maximum of 2000 entries for external (type 5) LSAs. You can change the maximum number of entries from 500 – 8000. After changing this limit, make sure to save the running-config file and reload the software. The change does not take effect until you reload or reboot the software.

For example, to change the maximum number entries from the default of 2000 to 3000, enter the following command.

```
Brocade(config-ospf6-router)#external-lsdb-limit 3000
```

Syntax: `ipv6 ospf area <number> | <ipv4-address>`

The `<entries>` parameter can be a numerical value from 500 – 8000 seconds.

To reset the maximum number of entries to its system default, enter the **no** form of this command.

Modifying OSPF V3 interface defaults

OSPF V3 has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

You can modify the default values for the following OSPF interface parameters:

- **Cost:** Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps and 1000 Mbps (1 Gbps) links. The command syntax is `ipv6 ospf cost <number>`. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for both 100 Mbps and 1000 Mbps links is 1, because the speed of 1000 Mbps was not in use at the time the OSPF cost formula was devised.
- **Dead-interval:** Indicates the number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The command syntax is `ipv6 ospf dead-interval <seconds>`. The value can be from 1 – 2147483647 seconds. The default is 40 seconds.
- **Hello-interval:** Represents the length of time between the transmission of hello packets. The command syntax is `ipv6 ospf hello-interval <seconds>`. The value can be from 1 – 65535 seconds. The default is 10 seconds.
- **Instance:** Indicates the number of OSPF V3 instances running on an interface. The command syntax is `ipv6 ospf instance <number>`. The value can be from 0 – 255. The default is 1.
- **MTU-ignore:** Allows you to disable a check that verifies the same MTU is used on an interface shared by neighbors. The command syntax is `ipv6 ospf mtu-ignore`. By default, the mismatch detection is enabled.
- **Network:** Allows you to configure the OSPF network type. The command syntax is `ipv6 ospf network [point-to-multipoint]`. The default setting of the parameter depends on the network type.
- **Passive:** When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. This option affects all IPv6 subnets configured on the interface. The command syntax is `ipv6 ospf passive`. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network.
- **Priority:** Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The command syntax is `ipv6 ospf priority <number>`. The value can be from 0 – 255. The default is 1. If you set the priority to 0, the router does not participate in DR and BDR election.
- **Retransmit-interval:** The time between retransmissions of LSAs to adjacent routers for an interface. The command syntax is `ipv6 ospf retransmit-interval <seconds>`. The value can be from 0 – 3600 seconds. The default is 5 seconds.

- **Transmit-delay:** The time it takes to transmit Link State Update packets on this interface. The command syntax is **ipv6 ospf transmit-delay <seconds>**. The value can be from 0 – 3600 seconds. The default is 1 second.

Disabling or re-enabling event logging

OSPF V3 does not currently support the generation of SNMP traps. Instead, you can disable or re-enable the logging of OSPF V3-related events such as neighbor state changes and database overflow conditions. By default, the Brocade device logs these events.

To disable the logging of events, enter the following command.

```
Brocade(config-ospf6-router)#no log-status-change
```

Syntax: [no] log-status-change

To re-enable the logging of events, enter the following command.

```
Brocade(config-ospf6-router)#log-status-change
```

IPSec for OSPFv3

This section describes the implementation of Internet Protocol Security (IPSec) for securing OSPFv3 traffic. For background information and configuration steps, refer to [“IPsec for OSPFv3 configuration”](#) on page 1308.

IPsec is available for OSPFv3 traffic only and only for packets that are “for-us.” A for-us packet is addressed to one of the IPv6 addresses on the device or to an IPv6 multicast address. Packets that are just forwarded by the line card do not receive IPsec scrutiny.

Brocade devices support the following components of IPsec for IPv6-addressed packets:

- Authentication through Encapsulating Security Payload (ESP) in transport mode
- HMAC-SHA1-96 as the authentication algorithm
- Manual configuration of keys
- Configurable rollover timer

IPsec can be enabled on the following logical entities:

- Interface
- Area
- Virtual link

With respect to traffic classes, this implementation of IPsec uses a single security association (SA) between the source and destination to support all traffic classes and so does not differentiate between the different classes of traffic that the DSCP bits define.

Instructions for configuring IPsec on these entities appear in [“IPsec for OSPFv3 configuration”](#) on page 1308.

IPsec on a virtual link is a global configuration. Interface and area IPsec configurations are more granular.

Among the entities that can have IPsec protection, the interfaces and areas can overlap. The interface IPsec configuration takes precedence over the area IPsec configuration when an area and an interface within that area use IPsec. Therefore, if you configure IPsec for an interface and an area configuration also exists that includes this interface, the interface's IPsec configuration is used by that interface. However, if you disable IPsec on an interface, IPsec is disabled on the interface even if the interface has its own, specific authentication. Refer to [“Disabling IPsec on an interface”](#) on page 1313.

For IPsec, the system generates two types of databases. The *security association database* (SAD) contains a security association for each interface or one global database for a virtual link. Even if IPsec is configured for an area, each interface that uses the area's IPsec still has its own security association in the SAD. Each SA in the SAD is a generated entry that is based on your specifications of an authentication protocol (ESP in the current release), destination address, and a security policy index (SPI). The SPI number is user-specified according to the network plan. Consideration for the SPI values to specify must apply to the whole network.

The system-generated security policy databases (SPDs) contain the security policies against which the system checks the for-us packets. For each for-us packet that has an ESP header, the applicable security policy in the security policy database (SPD) is checked to see if this packet complies with the policy. The IPsec task drops the non-compliant packets. Compliant packets continue on to the OSPFv3 module.

IPsec for OSPFv3 configuration

This section describes how to configure IPsec for an interface, area, and virtual link. It also describes how to change the key rollover timer if necessary and how to disable IPsec on a particular interface for special purposes.

By default, OSPFv3 IPsec authentication is disabled. The following IPsec parameters are configurable:

- ESP security protocol
- Authentication
- HMAC-SHA1-96 authentication algorithm
- Security parameter index (SPI)
- A 40-character key using hexadecimal characters
- An option for not encrypting the keyword when it appears in **show** command output
- Key rollover timer

NOTE

In the current release, certain keyword parameters must be entered even though only one keyword choice is possible for that parameter. For example, the only authentication algorithm in the current release is HMAC-SHA1-96, but you must nevertheless enter the keyword for this algorithm. Also, ESP currently is the only authentication protocol, but you must still enter the **esp** keyword. This section describes all keywords.

General considerations when configuring IPsec for OSPFv3

The IPsec component generates security associations and security policies based on certain user-specified parameters. The parameters are described with the syntax of each command in this section and also pointed out in the section with the **show** command examples, [“IPsec examples”](#) on page 1334. User-specified parameters and their relation to system-generated values are as follows:

- **Security association:** based on your entries for *security policy index* (SPI), *destination address*, and *security protocol* (currently ESP), the system creates a security association for each interface or virtual link.
- **Security policy database:** based on your entries for SPI, *source address*, *destination addresses*, and *security protocol*, the system creates a security policy database for each interface or virtual link.
- You can configure the same SPI and key on multiple interfaces and areas, but they still have unique IPsec configurations because the SA and policies are added to each separate security policy database (SPD) that is associated with a particular interface. If you configure an SA with the same SPI in multiple places, the rest of the parameters associated with the SA—such as key, crypto algorithm, and security protocol, and so on—must match. If the system detects a mismatch, it displays an error message.
- IPsec authentication for OSPFv3 requires the use of multiple SPDs, one for each interface. A virtual link has a separate, global SPD. The authentication configuration on a virtual link must be different from the authentication configuration for an area or interface, as required by RFC4552. The interface number is used to generate a non-zero security policy database identifier (SPDID), but for the global SPD for a virtual link, the system-generated SPDID is always zero. As a hypothetical example, the SPD for interface eth 1/1 might have the system-generated SPDID of 1, and so on.
- If you change an existing key, you must also specify a different SPI value. For example, in an interface context where you intend to change a key, you must type a different SPI value—which occurs before the key parameter on the command line—before you type the new key. The example in [“IPsec for OSPFv3 configuration”](#) illustrates this requirement.
- The old key is active for twice the current configured key-rollover-interval for the inbound direction. In the outbound direction, the old key remains active for a duration equal to the key-rollover-interval. If the key-rollover-interval is set to 0, the new key immediately takes effect for both directions. For a description of the key-rollover-interval, refer to the [“Changing the key rollover timer”](#) on page 1313 section.

Interface and area IPsec considerations

This section describes the precedence of interface and area IPsec configurations.

If you configure an interface IPsec by using the **ipv6 ospf authentication** command in the context of a specific interface, that interface's IPsec configuration overrides the area configuration of IPsec.

If you configure IPsec for an area, all interfaces that utilize the area-wide IPsec (where interface-specific IPsec is not configured) nevertheless receive an SPD entry (and SPDID number) that is unique for the interface.

The area-wide SPI that you specify is a constant for all interfaces in the area that use the area IPsec, but the use of different interfaces results in an SPDID and an SA that are unique to each interface. (Recall from “IPsec for OSPFv3” on page 1307 that the security policy database depends partly on the source IP address, so a unique SPD for each interface results.)

Considerations for IPsec on virtual links

The IPsec configuration for a virtual link is global, so only one security association database and one security policy database exist for virtual links if you choose to configure IPsec for virtual links.

The virtual link IPsec SAs and policies are added to all interfaces of the transit area for the outbound direction. For the inbound direction, IPsec SAs and policies for virtual links are added to the global database.

NOTE

The security association (SA), security protocol index (SPI), security protocol database (SPD), and key have mutual dependencies, as the subsections that follow describe.

Specifying the key rollover timer

Configuration changes for authentication takes effect in a controlled manner through the key rollover procedure as specified in RFC 4552, Section 10.1. The key rollover timer controls the timing of the configuration changeover. The key rollover timer can be configured in the IPv6 router OSPF context, as the following example illustrates.

```
Brocade(config-ospf6-router)#key-rollover-interval 200
```

Syntax: `key-rollover-interval <time>`

The range for the key-rollover-interval is 0 – 14400 seconds. The default is 300 seconds.

Configuring IPsec on a interface

For IPsec to work, the IPsec configuration must be the same on all the routers to which an interface connects.

For multicast, IPsec does not need or use a specific destination address—the destination address is “do not care,” and this status is reflected by the lone pair of colons (::) for destination address in the **show** command output.

To configure IPsec on an interface, proceed as in the following example.

NOTE

The IPsec configuration for an interface applies to the inbound and outbound directions. Also, the same authentication parameters must be used by all routers on the network to which the interface is connected, as described in section 7 of RFC 4552.

```
Brocade(config-if-e10000-1/2)#ipv6 ospf auth ipsec spi 429496795 esp sha1
abcdef12345678900987654321fedcba12345678
```

Syntax: `[no] ipv6 ospf authentication ipsec spi <spinum> esp sha1 [no-encrypt] <key>`

The **no** form of this command deletes IPsec from the interface.

The **ipv6** command is available in the configuration interface context for a specific interface.

The **ospf** keyword identifies OSPFv3 as the protocol to receive IPsec security.

The **authentication** keyword enables authentication.

The **ipsec** keyword specifies IPsec as the authentication protocol.

The **spi** keyword and the *<spinum>* variable specify the security parameter that points to the security association. The near-end and far-end values for spinum must be the same. The range for *<spinum>* is decimal 256 – 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that when you display the IPsec configuration, the key is displayed in its unencrypted form and also saved as unencrypted.

The *<key>* variable must be 40 hexadecimal characters. To change an existing key, you must also specify a different SPI value. You cannot just change the key without also specifying a different SPI, too. For example, in an interface context where you intend to change a key, you must type a different SPI value—which occurs before the key parameter on the command line—before you type the new key. The example in [“IPsec for OSPFv3 configuration”](#) illustrates this requirement.

If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- **encrypt** = the key string uses proprietary simple cryptographic 2-way algorithm.
- **encryptb64** = the key string uses proprietary base64 cryptographic 2-way algorithm.

This example results in the configuration shown in the screen output that follows. Note that because the optional **no-encrypt** keyword was omitted, the display of the key has the encrypted form by default.

```
interface ethernet 1/1/2
  enable
  ip address 40.3.3.1/8
  ipv6 address 40:3:3::1/64
  ipv6 ospf area 1
  ipv6 ospf authentication ipsec spi 429496795 esp sha1 encryptb64
  $ITJkQG5HWnw4M09tWVd
```

Configuring IPsec for an area

This application of the **area** command (for IPsec) applies to all of the interfaces that belong to an area unless an interface has its own IPsec configuration. (As described in [“Disabling IPsec on an interface”](#) on page 1313, the interface IPsec can be operationally disabled if necessary.) To configure IPsec for an area in the IPv6 router OSPF context, proceed as in the following example.

```
Brocade(config-ospf6-router)#area 2 auth ipsec spi 400 esp sha1
abcef12345678901234fedcba098765432109876
```

Syntax: **area** *<area-id>* **authentication ipsec spi** *<spinum>* **esp sha1** [**no-encrypt**] *<key>*

The **no** form of this command deletes IPsec from the area.

The **area** command and the *<area-id>* variable specify the area for this IPsec configuration. The *<area-id>* can be an integer in the range 0 – 2,147,483,647 or have the format of an IP address.

The **authentication** keyword specifies that the function to specify for the area is packet authentication.

The **ipsec** keyword specifies that IPsec is the protocol that authenticates the packets.

The **spi** keyword and the *<spinum>* variable specify the index that points to the security association. The near-end and far-end values for spinum must be the same. The range for *<spinum>* is decimal 256 – 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that the 40-character key is not encrypted upon either its entry or its display. The key must be 40 hexadecimal characters.

If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- **encrypt** = the key string uses proprietary simple cryptographic 2-way algorithm.
- **encryptb64** = the key string uses proprietary base64 cryptographic 2-way algorithm.

The configuration in the preceding example results in the configuration for area 2 that is illustrated in the following example.

```
ipv6 router ospf
  area 0
  area 1
  area 2
  area 2 auth ipsec spi 400 esp sha1 abcef12345678901234fedcba098765432109876
```

Configuring IPsec for a virtual link

IPsec on a virtual link has a global configuration.

To configure IPsec on a virtual link, enter the IPv6 router OSPF context of the CLI and proceed as the following example illustrates. (Note the **no-encrypt** option in this example.)

```
Brocade(config-ospf6-router)#area 1 vir 2.2.2.2 auth ipsec spi 360 esp sha1
no-encrypt 1234567890098765432112345678990987654321
```

Syntax: **[no] area <area-id> virtual <nbrid> authentication ipsec spi <spinum> esp sha1**
[no-encrypt] <key>

The **no** form of this command deletes IPsec from the virtual link.

The **area** command and the *<area-id>* variable specify the area is to be configured. The *<area-id>* can be an integer in the range 0 – 2,147,483,647 or have the format of an IP address.

The **virtual** keyword indicates that this configuration applies to the virtual link identified by the subsequent variable *<nbrid>*. The variable *<nbrid>* is in dotted decimal notation of an IP address.

The **authentication** keyword specifies that the function to specify for the area is packet authentication.

The **ipsec** keyword specifies that IPsec is the protocol that authenticates the packets.

The **spi** keyword and the *<spinum>* variable specify the index that points to the security association. The near-end and far-end values for spinum must be the same. The range for *<spinum>* is decimal 256 – 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that the 40-character key is not encrypted in **show** command displays. If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- **encrypt** = the key string uses proprietary simple cryptographic 2-way algorithm.
- **encryptb64** = the key string uses proprietary base64 cryptographic 2-way algorithm.

This example results in the following configuration.

```
area 1 virtual-link 2.2.2.2
area 1 virtual-link 2.2.2.2 authentication ipsec spi 360 esp sha1 no-encrypt 12
34567890098765432112345678990987654321
```

Disabling IPsec on an interface

For the purpose of troubleshooting, you can operationally disable IPsec on an interface by using the **ipv6 ospf authentication ipsec disable** command in the CLI context of a specific interface. This command disables IPsec on the interface whether its IPsec configuration is the area's IPsec configuration or is specific to that interface. The output of the **show ipv6 ospf interface command** shows the current setting for the disable command.

To disable IPsec on an interface, go to the CLI context of the interface and proceed as in the following example.

```
Brocade(config-if-e10000-1/2)#ipv6 ospf auth ipsec disable
```

Syntax: **[no] ipv6 ospf authentication ipsec disable**

The **no** form of this command restores the area and interface-specific IPsec operation.

Changing the key rollover timer

Configuration changes for authentication takes effect in a controlled manner through the key rollover procedure as specified in RFC 4552, Section 10.1. The key rollover timer controls the timing of the configuration changeover. The key rollover timer can be configured in the IPv6 router OSPF context, as the following example illustrates.

```
Brocade(config-ospf6-router)#key-rollover-interval 200
```

Syntax: **key-rollover-interval <time>**

The range for the key-rollover-interval is 0 – 14400 seconds. The default is 300 seconds.

Clearing IPsec statistics

This section describes the **clear ipsec statistics** command for clearing statistics related to IPsec. The command resets to 0 the counters (which you can view as a part of IPsec Packet Statistics). The counters hold IPsec packet statistics and IPsec error statistics. The following example illustrates the **show ipsec statistics** output.

```
Brocade#show ipsec statistics
                        IPsec Statistics
secEspCurrentInboundSAs 1      ipsecEspTotalInboundSAs: 2
secEspCurrentOutboundSA 1      ipsecEspTotalOutboundSAs: 2
                        IPsec Packet Statistics
secEspTotalInPkts:      20      ipsecEspTotalInPktsDrop: 0
secEspTotalOutPkts:      84
                        IPsec Error Statistics
secAuthenticationErrors 0
secReplayErrors:      0      ipsecPolicyErrors:      13
secOtherReceiveErrors: 0      ipsecSendErrors:      0
secUnknownSpiErrors: 0
```

To clear the statistics, enter the **clear ipsec statistics** command as in the following example.

```
Brocade#clear ipsec statistics
```

Syntax: **clear ipsec statistics**

This command takes no parameters.

Displaying OSPF V3 Information

You can display the information for the following OSPF V3 parameters:

- Areas
- Link state databases
- Interfaces
- Memory usage
- Neighbors
- Redistributed routes
- Routes
- SPF
- Virtual links
- Virtual neighbors
- IPsec

Displaying OSPF V3 area information

To display global OSPF V3 area information for the Brocade device, enter the following command at any CLI level.

```
Brocade#show ipv6 ospf area
Area 0:
  Interface attached to this area: loopback 2 ethe 3/2 tunnel 2
  Number of Area scoped LSAs is 6
  Statistics of Area 0:
    SPF algorithm executed 16 times
    SPF last updated: 335256 sec ago
    Current SPF node count: 3
      Router: 2 Network: 1
    Maximum of Hop count to nodes: 2
...
```

Syntax: `show ipv6 ospf area [<area id>]`

You can specify the <area-id> parameter in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

The <area-id> parameter restricts the display to the specified OSPF area.

This display shows the following information.

TABLE 220 OSPF V3 area information fields

Field	Description
Area	The area number.
Interface attached to this area	The router interfaces attached to the area.
Number of Area scoped LSAs	Number of LSAs with a scope of the specified area.
SPF algorithm executed	The number of times the OSPF Shortest Path First (SPF) algorithm is executed within the area.
SPF last updated	The interval in seconds that the SPF algorithm was last executed within the area.
Current SPF node count	The current number of SPF nodes in the area.
Router	Number of router LSAs in the area.
Network	Number of network LSAs in the area.
Indx	The row number of the entry in the router OSPF area table.
Area	The area number.
Maximum hop count to nodes.	The maximum number of hop counts to an SPF node within the area.

Displaying OSPF V3 database information

You can display a summary of the link state database or detailed information about a specified LSA type.

To display a summary of a device link state database, enter the **show ipv6 ospf database** command at any CLI level.

```
Brocade#show ipv6 ospf database
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
0                Link 000001e6 223.223.223.223 800000ab 1547 8955 68
0                Link 000000d8 1.1.1.1         800000aa 1295 0639 68
0                Link 00000185 223.223.223.223 800000ab 1481 7e6b 56
0                Iap  00000077 223.223.223.223 800000aa 1404 966a 56
0                Rtr  00000124 223.223.223.223 800000b0 1397 912c 40
0                Net  00000016 223.223.223.223 800000aa 1388 1b09 32
0                Iap  000001d1 223.223.223.223 800000bd 1379 a072 72
0                Iap  000000c3 1.1.1.1         800000ae 1325 e021 52
0                Rtr  00000170 1.1.1.1         800000ad 1280 af8e 40
N/A              Extn 00000062 223.223.223.223 800000ae 1409 0ca7 32
N/A              Extn 0000021d 223.223.223.223 800000a8 1319 441e 32
```

Syntax: **show ipv6 ospf database** [**advrtr** <ipv4-address> | **as-external** | **extensive** | **inter-prefix** | **inter-router** | **intra-prefix** | **link** | **link-id** <number> | **network** | **router** [**scope** <area-id> | **as** | **link**]]

The **advrtr** <ipv4-address> parameter displays detailed information about the LSAs for a specified advertising router only.

The **as-external** keyword displays detailed information about the AS externals LSAs only.

The **extensive** keyword displays detailed information about all LSAs in the database.

The **inter-prefix** keyword displays detailed information about the inter-area prefix LSAs only.

The **inter-router** keyword displays detailed information about the inter-area router LSAs only.

The **intra-prefix** keyword displays detailed information about the intra-area prefix LSAs only.

The **link** keyword displays detailed information about the link LSAs only.

The **link-id** <number> parameter displays detailed information about the specified link LSAs only.

The **network** <number> displays detailed information about the network LSAs only.

The **router** <number> displays detailed information about the router LSAs only.

The **scope** <area-id> parameter displays detailed information about the LSAs for a specified area, AS, or link.

This display shows the following information.

TABLE 221 OSPF V3 database summary fields

Field	Description
Area ID	The OSPF area in which the Brocade device resides.
Type	Type of LSA. LSA types can be the following: <ul style="list-style-type: none"> • Rtr – Router LSAs (Type 1). • Net – Network LSAs (Type 2). • Inap – Inter-area prefix LSAs for ABRs (Type 3). • Inar – Inter-area router LSAs for ASBRs (Type 4). • Extn – AS external LSAs (Type 5). • Link – Link LSAs (Type 8). • lap – Intra-area prefix LSAs (Type 9).
LS ID	The ID of the LSA, in hexadecimal, from which the device learned this route.
Adv Rtr	The device that advertised the route.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps it with a sequence number to enable the Brocade device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA, in seconds.
Chksum	A checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The Brocade device uses the checksum to verify that the packet is not corrupted.
Len	The length, in bytes, of the LSA.

For example, to display detailed information about all LSAs in the database, enter the **show ipv6 ospf database extensive** command at any CLI level.

Displaying OSPF V3 Information

```
Brocade#show ipv6 ospf database extensive
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
0                Link 00000031 1.1.1.1          80000001 35   6db9   56
    Router Priority: 1
    Options: V6E---R--
    LinkLocal Address: fe80::1
    Number of Prefix: 1
    Prefix Options:
    Prefix: 3002::/64
...
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
0                Iap 00000159 223.223.223.223 800000ab 357   946b   56
    Number of Prefix: 2
    Referenced LS Type: Network
    Referenced LS ID: 00000159
    Referenced Advertising Router: 223.223.223.223
    Prefix Options: Metric: 0
    Prefix: 2000:4::/64
    Prefix Options: Metric: 0
    Prefix: 2002:c0a8:46a::/64
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
0                Rtr 00000039 223.223.223.223 800000b1 355   8f2d   40
    Capability Bits: --E-
    Options: V6E---R--
    Type: Transit Metric: 1
    Interface ID: 00000058 Neighbor Interface ID: 00000058
    Neighbor Router ID: 223.223.223.223
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
0                Net 000001f4 223.223.223.223 800000ab 346   190a   32
    Options: V6E---R--
    Attached Router: 223.223.223.223
    Attached Router: 1.1.1.1
...
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
N/A              Extn 000001df 223.223.223.223 800000af 368   0aa8   32
    Bits: E
    Metric: 00000001
    Prefix Options:
    Referenced LSType: 0
    Prefix: 2002::/16
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
1                Inap 0000011d 10.1.1.188       80000001 124   25de   36
    Metric: 2
    Prefix Options:
    Prefix: 2000:2:2::/64
Area ID          Type LS ID      Adv Rtr          Seq(Hex) Age  Cksum  Len
0                Inar 0000005b 10.1.1.198       80000001 990   dbad   32
    Options: V6E---R--
    Metric: 1
    Destination Router ID:10.1.1.188
```

NOTE

Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

The fields that display depend upon the LSA type as shown in the following table.

TABLE 222 OSPF V3 detailed database information fields

Field	Description
Router LSA (Type 1) (Rtr) fields	
Capability Bits	<p>A bit that indicates the capability of the Brocade device. The bit can be set to one of the following:</p> <ul style="list-style-type: none"> • B – The device is an area border router. • E – The device is an AS boundary router. • V – The device is a virtual link endpoint. • W – The device is a wildcard multicast receiver.
Options	<p>A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following:</p> <p>V6 – The device should be included in IPv6 routing calculations.</p> <p>E – The device floods AS-external-LSAs as described in RFC 2740.</p> <p>MC – The device forwards multicast packets as described in RFC 1586.</p> <p>N – The device handles type 7 LSAs as described in RFC 1584.</p> <p>R – The originator is an active router.</p> <p>DC – The device handles demand circuits.</p>
Type	<p>The type of interface. Possible types can be the following:</p> <ul style="list-style-type: none"> • Point-to-point – A point-to-point connection to another router. • Transit – A connection to a transit network. • Virtual link – A connection to a virtual link.
Metric	The cost of using this router interface for outbound traffic.
Interface ID	The ID assigned to the router interface.
Neighbor Interface ID	The interface ID that the neighboring router has been advertising in hello packets sent on the attached link.
Neighbor Router ID	The router ID (IPv4 address) of the neighboring router that advertised the route. (By default, the Brocade router ID is the IPv4 address configured on the lowest numbered loopback interface. If the Brocade device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.)

TABLE 222 OSPF V3 detailed database information fields (Continued)

Field	Description
Network LSA (Type 2) (Net) fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC – The device handles demand circuits.
Attached Router	The address of the neighboring router that advertised the route.
Inter-Area Prefix LSA (Type 3) (Inap) fields	
Metric	The cost of the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Prefix	The IPv6 prefix included in the LSA.
Inter-Area Router LSA (Type 4) (Inar) fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC – The device handles demand circuits.
Metric	The cost of the route.
Destination Router ID	The ID of the router described in the LSA.
AS External LSA (Type 5) (Extn) fields	
Bits	The bit can be set to one of the following: <ul style="list-style-type: none"> • E – If bit E is set, a Type 2 external metric. If bit E is zero, a Type 1 external metric. • F – A forwarding address is included in the LSA. • T – An external route tag is included in the LSA.
Metric	The cost of this route, which depends on bit E.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Referenced LS Type	If non-zero, an LSA with this LS type is associated with the LSA.
Prefix	The IPv6 prefix included in the LSA.
Link LSA (Type 8) (Link) fields	
Router Priority	The router priority of the interface attaching the originating router to the link.
Options	The set of options bits that the router would like set in the network LSA that will be originated for the link.
Link Local Address	The originating router link-local interface address on the link.
Number of Prefix	The number of IPv6 address prefixes contained in the LSA.

TABLE 222 OSPF V3 detailed database information fields (Continued)

Field	Description
Prefix Options	An 8-bit field of capabilities that serve as input to various routing calculations: <ul style="list-style-type: none"> • NU – The prefix is excluded from IPv6 unicast calculations. • LA – The prefix is an IPv6 interface address of the advertising router. • MC – The prefix is included in IPv6 multicast routing calculations.
Prefix	The IPv6 prefix included in the LSA.
Intra-Area Prefix LSAs (Type 9) (lap) fields	
Number of Prefix	The number of prefixes included in the LSA.
Referenced LS Type, Referenced LS ID	Identifies the router-LSA or network-LSA with which the IPv6 address prefixes are associated.
Referenced Advertising Router	The address of the neighboring router that advertised the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Metric	The cost of using the advertised prefix.
Prefix	The IPv6 prefix included in the LSA.
Number of Prefix	The number of prefixes included in the LSA.

Displaying OSPF V3 interface information

You can display a summary of information for all OSPF V3 interfaces or detailed information about a specified OSPF V3 interface.

To display a summary of OSPF V3 interfaces, enter the **show ipv6 ospf interface** command at any CLI level.

```
Brocade#show ipv6 ospf interface
Interface  OSPF      Status State   Area
-----
ethe 3/1   up
ethe 3/2   enabled  up      DR      0
ethe 3/4   disabled down
loopback 2 enabled  up      Loopback 0
tunnel 1   disabled down
tunnel 2   enabled  up      P2P     0
tunnel 6   up
```

Syntax: **show ipv6 ospf interface** [**ethernet** <port> | **loopback** <number> | **tunnel** <number> | **ve** <number>]

The **ethernet** | **loopback** | **tunnel** | **ve** parameter specifies the interface for which to display information. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the interface.

This display shows the following information.

TABLE 223 Summary of OSPF V3 interface information

Field	Description
Interface	The interface type, and the port number or number of the interface.
OSPF	The state of OSPF V3 on the interface. Possible states include the following: <ul style="list-style-type: none"> • Enabled. • Disabled.
Status	The status of the link. Possible status include the following: <ul style="list-style-type: none"> • Up. • Down.
State	The state of the interface. Possible states includes the following: <ul style="list-style-type: none"> • DR – The interface is functioning as the Designated Router for OSPF V3. • BDR – The interface is functioning as the Backup Designated Router for OSPF V3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Area	The OSPF area to which the interface belongs.

For example, to display detailed information about Ethernet interface 2, enter the **show ipv6 ospf interface ethernet** command at any level of the CLI.

```

Brocade#show ipv6 ospf interface ethernet 3/2
ethe 3/2 is up, type BROADCAST
  IPv6 Address:
    2002:c0a8:46a::1/64
    2000:4::106/64
  Instance ID 0, Router ID 223.223.223.223
  Area ID 0, Cost 1
  State DR, Transmit Delay 1 sec, Priority 1
  Timer intervals :
    Hello 10, Dead 40, Retransmit 5
  DR:223.223.223.223 BDR:1.1.1.1 Number of I/F scoped LSAs is 2
  DRElection:      5 times, DelayedLSAck:    523 times
  Neighbor Count = 1, Adjacent Neighbor Count= 1
  Neighbor:
    1.1.1.1 (BDR)
  Statistics of interface ethe 3/2:
    Type      tx    rx tx-byte rx-byte
    Unknown    0     0      0      0
    Hello   3149  3138 1259284 1255352
    DbDesc     7     6     416    288
    LSReq       2     2      80    152
    LSUupdate 1508   530 109508   39036
    LSAck      526 1398   19036   54568

```


This display shows the following information.

TABLE 224 Detailed OSPF V3 interface information

Field	Description
Interface status	The status of the interface. Possible status includes the following: <ul style="list-style-type: none"> Up. Down.
Type	The type of OSPF V3 circuit running on the interface. Possible types include the following: <ul style="list-style-type: none"> BROADCAST POINT TO POINT UNKNOWN
IPv6 Address	The IPv6 address(es) assigned to the interface.
Instance ID	An identifier for an instance of OSPF V3.
Router ID	The IPv4 address of the Brocade device. By default, the Brocade router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Area ID	The IPv4 address or numerical value of the area in which the interface belongs.
Cost	The overhead required to send a packet through the interface.
State	The state of the interface. Possible states include the following: <ul style="list-style-type: none"> DR – The interface is functioning as the Designated Router for OSPF V3. BDR – The interface is functioning as the Backup Designated Router for OSPF V3. Loopback – The interface is functioning as a loopback interface. P2P – The interface is functioning as a point-to-point interface. Passive – The interface is up but it does not take part in forming an adjacency. Waiting – The interface is trying to determine the identity of the BDR for the network. None – The interface does not take part in the OSPF interface state machine. Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Transmit delay	The amount of time, in seconds, it takes to transmit Link State Updates packets on the interface.
Priority	The priority used when selecting the DR and the BDR. If the priority is 0, the interface does not participate in the DR and BDR election.
Timer intervals	The interval, in seconds, of the hello-interval, dead-interval, and retransmit-interval timers.
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.
Number of I/F scoped LSAs	The number of interface LSAs scoped for a specified area, AS, or link.
DR Election	The number of times the DR election occurred.
Delayed LSA Ack	The number of the times the interface sent a delayed LSA acknowledgement.
Neighbor Count	The number of neighbors to which the interface is connected.
Adjacent Neighbor Count	The number of neighbors with which the interface has formed an active adjacency.

TABLE 224 Detailed OSPF V3 interface information (Continued)

Field	Description
Neighbor	The router ID (IPv4 address) of the neighbor. This field also identifies the neighbor as a DR or BDR, if appropriate.
Interface statistics	<p>The following statistics are provided for the interface:</p> <ul style="list-style-type: none"> Unknown – The number of Unknown packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Unknown packets. Hello – The number of Hello packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Hello packets. DbDesc – The number of Database Description packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Database Description packets. LSReq – The number of link-state requests transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. LSUpdate – The number of link-state updates transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. LSAck – The number of link-state acknowledgements transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state acknowledgements.

Displaying OSPF V3 memory usage

To display information about OSPF V3 memory usage, enter the **show ipv6 ospf memory** command at any level of the CLI.

```

Brocade#show ipv6 ospf memory
Total Static Memory Allocated : 5829 bytes
Total Dynamic Memory Allocated : 0 bytes
Memory Type           Size      Allocated  Max-alloc  Alloc-Fails
MTYPE_OSPF6_TOP        0          0           0           0
MTYPE_OSPF6_LSA_HDR    0          0           0           0
MTYPE_OSPF6_RMAP_COMPILED 0          0           0           0
MTYPE_OSPF6_OTHER      0          0           0           0
MTYPE_THREAD_MASTER    0          0           0           0
MTYPE_OSPF6_AREA       0          0           0           0
MTYPE_OSPF6_AREA_RANGE 0          0           0           0
MTYPE_OSPF6_SUMMARY_ADDRE 0          0           0           0
MTYPE_OSPF6_IF         0          0           0           0
MTYPE_OSPF6_NEIGHBOR   0          0           0           0
MTYPE_OSPF6_ROUTE_NODE 0          0           0           0
MTYPE_OSPF6_ROUTE_INFO 0          0           0           0
MTYPE_OSPF6_PREFIX     0          0           0           0
MTYPE_OSPF6_LSA        0          0           0           0
MTYPE_OSPF6_VERTEX     0          0           0           0
MTYPE_OSPF6_SPFTREE    0          0           0           0
MTYPE_OSPF6_NEXTHOP    0          0           0           0
MTYPE_OSPF6_EXTERNAL_INFO 0          0           0           0
MTYPE_THREAD           0          0           0           0

```

Syntax: show ipv6 ospf memory

This display shows the following information.

TABLE 225 OSPF V3 memory usage information

Field	Description
Total Static Memory Allocated	A summary of the amount of static memory allocated, in bytes, to OSPF V3.
Total Dynamic Memory Allocated	A summary of the amount of dynamic memory allocated, in bytes, to OSPF V3.
Memory Type	The type of memory used by OSPF V3. (This information is for use by Brocade technical support in case of a problem.)
Size	The size of a memory type.
Allocated	The amount of memory currently allocated to a memory type.
Max-alloc	The maximum amount of memory that was allocated to a memory type.
Alloc-Fails	The number of times an attempt to allocate memory to a memory type failed.

Displaying OSPF V3 neighbor information

You can display a summary of OSPF V3 neighbor information for the Brocade device or detailed information about a specified neighbor.

To display a summary of OSPF V3 neighbor information for the device, enter the **show ipv6 ospf neighbor** command at any CLI level.

```
Brocade#show ipv6 ospf neighbor
RouterID      Pri State      DR                      BDR                      Interface[State]
1.1.1.1       1 Full      223.223.223.223 1.1.1.1                ethe 3/2 [DR]
```

Syntax: **show ipv6 ospf neighbor** [**router-id** <ipv4-address>]

The **router-id** <ipv4-address> parameter displays only the neighbor entries for the specified router.

This display shows the following information.

TABLE 226 Summary of OSPF V3 neighbor information

Field	Description
Router ID	The IPv4 address of the neighbor. By default, the Brocade router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Pri	The OSPF V3 priority of the neighbor. The priority is used during election of the DR and BDR.
State	The state between the Brocade device and the neighbor. The state can be one of the following: <ul style="list-style-type: none"> • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
DR	The router ID (IPv4 address) of the DR.

TABLE 226 Summary of OSPF V3 neighbor information (Continued)

Field	Description
BDR	The router ID (IPv4 address) of the BDR.
Interface [State]	<p>The interface through which the router is connected to the neighbor. The state of the interface can be one of the following:</p> <ul style="list-style-type: none"> • DR – The interface is functioning as the Designated Router for OSPF V3. • BDR – The interface is functioning as the Backup Designated Router for OSPF V3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.

For example, to display detailed information about a neighbor with the router ID of 1.1.1.1, enter the **show ipv6 ospf neighbor router-id** command at any CLI level.

```

Brocade#show ipv6 ospf neighbor router-id 3.3.3.3
RouterID      Pri State      DR          BDR          Interface[State]
3.3.3.3       1 Full      3.3.3.3     1.1.1.1     ve 10 [BDR]
DbDesc bit for this neighbor: --s
Nbr Ifindex of this router: 1
Nbr DRDecision: DR 3.3.3.3, BDR 1.1.1.1
Last received DbDesc: opt:xxx ifmtu:0 bit:--s seqnum:0
Number of LSAs in DbDesc retransmitting: 0
Number of LSAs in SummaryList: 0
Number of LSAs in RequestList: 0
Number of LSAs in RetransList: 0
SeqnumMismatch 0 times, BadLSReq 0 times
OnewayReceived 0 times, InactivityTimer 0 times
DbDescRetrans 0 times, LSReqRetrans 0 times
LSUpdateRetrans 1 times
LSAReceived 12 times, LSUpdateReceived 6 times

```

This display shows the following information.

TABLE 227 Detailed OSPF V3 neighbor information

Field	Description
Router ID	For information about this field, refer to Table 226 on page 1325.
Pri	For information about this field, refer to Table 226 on page 1325.
State	For information about this field, refer to Table 226 on page 1325.
DR	For information about this field, refer to Table 226 on page 1325.
BDR	For information about this field, refer to Table 226 on page 1325.
Interface [State]	For information about this field, refer to Table 226 on page 1325.

TABLE 227 Detailed OSPF V3 neighbor information (Continued)

Field	Description
DbDesc bit...	The Database Description packet, which includes 3 bits of information: <ul style="list-style-type: none"> • The first bit can be “i” or “-”. “i” indicates the inet bit is set. “-” indicates the inet bit is not set. • The second bit can be “m” or “-”. “m” indicates the more bit is set. “-” indicates the more bit is not set. • The third bit can be “m” or “s”. An “m” indicates the master. An “s” indicates standby.
Index	The ID of the LSA from which the neighbor learned of the router.
DR Decision	The router ID (IPv4 address) of the neighbor elected DR and BDR.
Last Received Db Desc	The content of the last database description received from the specified neighbor.
Number of LSAs in Db Desc retransmitting	The number of LSAs that need to be retransmitted to the specified neighbor.
Number of LSAs in Summary List	The number of LSAs in the neighbor summary list.
Number of LSAs in Request List	The number of LSAs in the neighbor request list.
Number of LSAs in Retransmit List	The number of LSAs in the neighbor retransmit list.
Seqnum Mismatch	The number of times sequence number mismatches occurred.
BadLSReq	The number of times the neighbor received a bad link-state request from the Brocade device.
One way received	The number of times a hello packet, which does not mention the router, is received from the neighbor. This omission in the hello packet indicates that the communication with the neighbor is not bidirectional.
Inactivity Timer	The number of times that the neighbor inactivity timer expired.
Db Desc Retransmission	The number of times sequence number mismatches occurred.
LSReqRetrans	The number of times the neighbor retransmitted link-state requests to the Brocade device.
LSUpdateRetrans	The number of times the neighbor retransmitted link-state updates to the Brocade device.
LSA Received	The number of times the neighbor received LSAs from the Brocade device.
LS Update Received	The number of times the neighbor received link-state updates from the Brocade device.

Displaying routes redistributed into OSPF V3

You can display all IPv6 routes or a specified IPv6 route that the Brocade device has redistributed into OSPF V3.

To display all IPv6 routes that the device has redistributed into OSPF V3, enter the **show ipv6 ospf redistribute route** command at any level of the CLI.

```
Brocade#show ipv6 ospf redistribute route
Id      Prefix
snIpAsPathAccessListStringRegExpression
1       2002::/16
2       2002:1234::/32
Protocol Metric Type Metric
Static   Type-2      1
Static   Type-2      1
```

Syntax: `show ipv6 ospf redistribute route [<ipv6-prefix>]`

The `<ipv6-prefix>` parameter specifies an IPv6 network prefix. (You do not need to specify the length of the prefix.)

For example, to display redistribution information for the prefix 2002::, enter the **show ipv6 ospf redistribute route** command at any level of the CLI.

```
Brocade#show ipv6 ospf redistribute route 2002::
Id      Prefix
1       2002::/16
Protocol Metric Type Metric
Static   Type-2      1
```

These displays show the following information.

TABLE 228 OSPF V3 redistribution information

Field	Description
ID	An ID for the redistributed route.
Prefix	The IPv6 routes redistributed into OSPF V3.
Protocol	The protocol from which the route is redistributed into OSPF V3. Redistributed protocols can be the following: <ul style="list-style-type: none"> • RIP – RIPv6. • Static – IPv6 static route table. • Connected – A directly connected network.
Metric Type	The metric type used for routes redistributed into OSPF V3. The metric type can be the following: <ul style="list-style-type: none"> • Type-1 – Specifies a small metric (2 bytes). • Type-2 – Specifies a big metric (3 bytes).
Metric	The value of the default redistribution metric, which is the OSPF cost of redistributing the route into OSPF V3.

Displaying OSPF V3 route information

You can display the entire OSPF V3 route table for the Brocade device or only the route entries for a specified destination.

To display the entire OSPF V3 route table for the device, enter the **show ipv6 ospf routes** command at any level of the CLI.

```

Brocade#show ipv6 ospf routes
Current Route count: 4
  Intra: 4 Inter: 0 External: 0 (Type1 0/Type2 0)
  Equal-cost multi-path: 0
  Destination          Options   Area          Cost Type2 Cost
  Next Hop Router      Outgoing Interface
*IA 2000:4::/64        V6E---R-- 0.0.0.0        1 0
  ::                   ethe 3/2
*IA 2002:c0a8:46a::/64 V6E---R-- 0.0.0.0        1 0
  ::                   ethe 3/2
*IA 2999::1/128        ----- 0.0.0.0        0 0
  ::                   loopback 2
*IA 2999::2/128        V6E---R-- 0.0.0.0        1 0
  fe80::2e0:52ff:fe91:bb37 ethe 3/2

```

Syntax: `show ipv6 ospf routes [<ipv6-prefix>]`

The `<ipv6-prefix>` parameter specifies a destination IPv6 prefix. (You do not need to specify the length of the prefix.) If you use this parameter, only the route entries for this destination are shown.

For example, to display route information for the destination prefix 2000:4::, enter the **show ipv6 ospf routes** command at any level of the CLI.

```

Brocade#show ipv6 ospf routes 2000:4::
Destination          Options   Area          Cost Type2 Cost
Next Hop Router      Outgoing Interface
*IA 2000:4::/64      V6E---R-- 0.0.0.0        1 0
  ::                 ethe 3/2

```

These displays show the following information.

TABLE 229 OSPF V3 route information

Field	Description
Current Route Count (Displays with the entire OSPF V3 route table only)	The number of route entries currently in the OSPF V3 route table.
Intra/Inter/External (Type1/Type2) (Displays with the entire OSPF V3 route table only)	<p>The breakdown of the current route entries into the following route types:</p> <ul style="list-style-type: none"> • Inter – The number of routes that pass into another area. • Intra – The number of routes that are within the local area. • External1 – The number of type 1 external routes. • External2 – The number of type 2 external routes.
Equal-cost multi-path (Displays with the entire OSPF V3 route table only)	The number of equal-cost routes to the same destination in the OSPF V3 route table. If load sharing is enabled, the router equally distributes traffic among the routes.
Destination	The IPv6 prefixes of destination networks to which the Brocade device can forward IPv6 packets. “*IA” indicates the next router is an intra-area router.

TABLE 229 OSPF V3 route information (Continued)

Field	Description
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC – The device handles demand circuits.
Area	The area whose link state information has led to the routing table entry's collection of paths.
Cost	The type 1 cost of this route.
Type2 Cost	The type 2 cost of this route.
Next-Hop Router	The IPv6 address of the next router a packet must traverse to reach a destination.
Outgoing Interface	The router interface through which a packet must traverse to reach the next-hop router.

Displaying OSPF V3 SPF information

You can display the following OSPF V3 SPF information:

- SPF node information for a specified area.
- SPF table for a specified area.
- SPF tree for a specified area.

For example, to display information about SPF nodes in area 0, enter the **show ipv6 ospf spf node area** command at any level of the CLI.

```
Brocade#show ipv6 ospf spf node area 0
SPF node for Area 0
SPF node 223.223.223.223, cost: 0, hops: 0
  nexthops to node:
  parent nodes:
  child nodes: 223.223.223.223:88

SPF node 223.223.223.223:88, cost: 1, hops: 1
  nexthops to node:    :: ethe 3/2
  parent nodes: 223.223.223.223
  child nodes: 1.1.1.1:0

SPF node 1.1.1.1:0, cost: 1, hops: 2
  nexthops to node:    fe80::2e0:52ff:fe91:bb37 ethe 3/2
  parent nodes: 223.223.223.223:88
  child nodes:
```


Syntax: `show ipv6 ospf spf node area [<area-id>]`

The **node** keyword displays SPF node information.

The **area** *<area-id>* parameter specifies a particular area. You can specify the *<area-id>* in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

This display shows the following information.

TABLE 230 OSPF V3 SPF node information

Field	Description
SPF node	Each SPF node is identified by its router ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format <i><router-id>:<interface-id></i> .
Cost	The cost of traversing the SPF node to reach the destination.
Hops	The number of hops needed to reach the parent SPF node.
Next Hops to Node	The IPv6 address of the next hop-router or the router interface, or both, through which to access the next-hop router.
Parent Nodes	The SPF node parent nodes. A parent node is an SPF node at the highest level of the SPF tree, which is identified by its router ID.
Child Nodes	The SPF node child nodes. A child node is an SPF node at a lower level of the SPF tree, which is identified by its router ID and interface on which the node can be reached.

For example, to display the SPF table for area 0, enter the **show ipv6 ospf spf table area** command at any level of the CLI.

```
Brocade#show ipv6 ospf spf table area 0
  SPF table for Area 0
  Destination          Bits Options  Cost  Nexthop                               Interface
R 1.1.1.1              ---- V6E---R-    1  fe80::2e0:52ff:fe91:bb37             ethe 3/2
N 223.223.223.223[88]  ---- V6E---R-    1  ::                                     ethe 3/2
```

Syntax: `show ipv6 ospf spf table area <area-id>`

The **table** parameter displays the SPF table.

The **area** *<area-id>* parameter specifies a particular area. You can specify the *<area-id>* in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

This display shows the following information.

TABLE 231 OSPF V3 SPF table

Field	Description
Destination	The destination of a route, which is identified by the following: <ul style="list-style-type: none"> • “R”, which indicates the destination is a router. “N”, which indicates the destination is a network. • An SPF node router ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format <i><router-id>:<interface-id></i>.
Bits	A bit that indicates the capability of the Brocade device. The bit can be set to one of the following: <ul style="list-style-type: none"> • B – The device is an area border router. • E – The device is an AS boundary router. • V – The device is a virtual link endpoint. • W – The device is a wildcard multicast receiver.
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: <ul style="list-style-type: none"> V6 – The router should be included in IPv6 routing calculations. E – The router floods AS-external-LSAs as described in RFC 2740. MC – The router forwards multicast packets as described in RFC 1586. N – The router handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC – The router handles demand circuits.
Cost	The cost of traversing the SPF node to reach the destination.
Next hop	The IPv6 address of the next hop-router.
Interface	The router interface through which to access the next-hop router.

For example, to display the SPF tree for area 0, enter the **show ipv6 ospf spf tree area** command at any level of the CLI.

```
Brocade#show ipv6 ospf spf tree area 0
  SPF tree for Area 0
+- 223.223.223.223 cost 0
   +- 223.223.223.223:88 cost 1
      +- 1.1.1.1:0 cost 1
```

Syntax: **show ipv6 ospf spf tree area** *<area-id>*

The **tree** keyword displays the SPF table.

The **area** *<area-id>* parameter specifies a particular area. You can specify the *<area-id>* in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

In this sample output, consider the SPF node with the router ID 223.223.223.223 to be the top (root) of the tree and the local router. Consider all other layers of the tree (223.223.223.223:88 and 1.1.1.1:0) to be destinations in the network. Therefore, traffic destined from router 223.223.223.223 to router 1.1.1.1:0 must first traverse router 223.223.223.223:88.

Displaying IPv6 OSPF virtual link information

To display OSPF V3 virtual link information for the Brocade device, enter the **show ipv6 ospf virtual-link** command at any level of the CLI.

```
Brocade#show ipv6 ospf virtual-link
Index Transit Area ID Router ID Interface Address State
1      1              1.1.1.1      3003::2      P2P
```

Syntax: **show ipv6 ospf virtual-link**

This display shows the following information.

TABLE 232 OSPF V3 virtual link information

Field	Description
Index	An index number associated with the virtual link.
Transit Area ID	The ID of the shared area of two ABRs that serves as a connection point between the two routers.
Router ID	IPv4 address of the router at the other end of the virtual link (virtual neighbor).
Interface Address	The local address used to communicate with the virtual neighbor.
State	The state of the virtual link. Possible states include the following: <ul style="list-style-type: none"> P2P – The link is functioning as a point-to-point interface. DOWN – The link is down.

Displaying OSPF V3 virtual neighbor information

To display OSPF V3 virtual neighbor information for the Brocade device, enter the **show ipv6 ospf virtual-neighbor** command at any level of the CLI.

```
Brocade#show ipv6 ospf virtual-neighbor
Index Router ID Address State Interface
1      1.1.1.1    3002::1 Full ethe 2/3
```

Syntax: **show ipv6 ospf virtual-neighbor**

This display shows the following information.

TABLE 233 OSPF V3 virtual neighbor information

Field	Description
Index	An index number associated with the virtual neighbor.
Router ID	IPv4 address of the virtual neighbor.
Address	The IPv6 address to be used for communication with the virtual neighbor.

TABLE 233 OSPF V3 virtual neighbor information (Continued)

Field	Description
State	The state between the Brocade device and the virtual neighbor. The state can be one of the following: <ul style="list-style-type: none"> • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
Interface	The IPv6 address of the virtual neighbor.

IPsec examples

This section contains examples of IPsec configuration and the output from the IPsec-specific **show** commands. In addition, IPsec-related information appears in general **show** command output for interfaces and areas.

The **show** commands that are specific to IPsec are:

- **show ipsec sa**
- **show ipsec policy**
- **show ipsec statistics**

The other **show** commands with IPsec-related information are:

- **show ipv6 ospf area**
- **show ipv6 ospf interface**

Showing IPsec security association information

The **show ipsec sa** command displays the IPsec security association databases, as follows.

```
Brocade#show ipsec sa
IPSEC Security Association Database(Entries:8)
SPDID(if)   Dir  Encap  SPI      Destination      AuthAlg  EncryptAlg
ALL         in   ESP    512      35:1:1::1        sha1     Null
eth1/1/1    out  ESP    302      ::               sha1     Null
eth1/1/1    in   ESP    302      FE80::           sha1     Null
eth1/1/1    out  ESP    512      10:1:1::2        sha1     Null
ALL         in   ESP    512      35:1:1::1        sha1     Null
eth1/1/2    out  ESP    302      ::               sha1     Null
eth1/1/2    in   ESP    302      FE80::           sha1     Null
eth1/1/2    out  ESP    512      10:1:1::2        sha1     Null
```

Syntax: **show ipsec sa**

Showing IPsec policy

The **show ipsec policy** command displays the database for the IPsec security policies. The fields for this **show** command output appear in the screen output example that follows. However, you should understand the layout and column headings for the display before trying to interpret the information in the example screen.

Each policy entry consists of two categories of information:

- The policy information
- The SA used by the policy

The policy information line in the screen begins with the heading Ptype and also has the headings Dir, Proto, Source (Prefix:TCP/UDP Port), and Destination (Prefix:TCP/UDPPort). The SA line contains the SPDID, direction, encapsulation (always ESP in the current release), the user-specified SPI, For readability, the policy information is described in [Table 234](#), and SA-specific information is in [Table 235](#).

```
Brocade#show ipsec policy
          IPSEC Security Policy Database(Entries:8)
PType  Dir Proto Source(Prefix:TCP/UDP Port)  Destination(Prefix:TCP/UDPPort)
SA: SPDID(if) Dir Encap SPI      Destination
use   in  OSPF FE80::/10:any          ::/0:any
SA: eth1/1/2 in  ESP   302          FE80::
use   out OSPF FE80::/10:any          ::/0:any
SA: eth1/1/2 out ESP   302          ::
use   in  OSPF FE80::/10:any          ::/0:any
SA: eth1/1/1 in  ESP   302          FE80::
use   out OSPF FE80::/10:any          ::/0:any
SA: eth1/1/1 out ESP   302          ::
use   in  OSPF 35:1:1::1/128:any      10:1:1::2/128:any
SA: ethALL in  ESP   512          10:1:1::2
use   out OSPF 10:1:1::2/128:any      35:1:1::1/128:any
SA: eth1/1/1 out ESP   512          35:1:1::1
use   in  OSPF 35:1:1::1/128:any      10:1:1::2/128:any
SA: ethALL in  ESP   512          10:1:1::2
use   out OSPF 10:1:1::2/128:any      35:1:1::1/128:any
```

Syntax: **show ipsec policy**

This command takes no parameters.

TABLE 234 IPsec policy information

Field	Description
PType	This field contains the policy type. Of the existing policy types, only the “use” policy type is supported, so each entry can have only “use.”
Dir	The direction of traffic flow to which the IPsec policy is applied. Each direction has its own entry.
Proto	The only possible routing protocol for the security policy in the current release is OSPFv3.

TABLE 234 IPsec policy information (Continued)

Field	Description
Source	The source address consists of the IPv6 prefix and the TCP or UDP port identifier.
Destination	<p>The destination address consists of the IPv6 prefix. Certain logical elements have a bearing on the meaning of the destination address and its format, as follows:</p> <p>For IPsec on an interface or area, the destination address is shown as a prefix of 0xFE80 (link local). The solitary "::" (no prefix) indicates a "do not-care" situation because the connection is multicast. In this case, the security policy is enforced without regard for the destination address.</p> <p>For a virtual link (SPDID = 0), the address is required.</p>

TABLE 235 SA used by the policy

Field	Description
SA	This heading points at the SA-related headings for information used by the security policy. Thereafter, on each line of this part of the IPsec entry (which alternates with lines of policy information Table 234), "SA:" points at the fields under those SA-related headings. The remainder of this table describes each of the SA-related items.
SPDID	The Security policy database identifier (SPDID) consists of interface type and Interface ID.
Dir	The Dir field is either 'in' for inbound or "out" for outbound.
Encap	The type of encapsulation in the current release is ESP.
SPI	Security parameter index.
Destination	<p>The IPv6 address of the destination endpoint. From the standpoint of the near interface and the area, the destination is not relevant and therefore appears as ::/0:any.</p> <p>For a virtual link, both the inbound and outbound destination addresses are relevant.</p>

Showing IPsec statistics

The **show ipsec statistics** command displays the error and other counters for IPSec, as this example shows.

```

Brocade#show ipsec statistics
                               IPsecrity Statistics
secEspCurrentInboundSAs 1      ipsecEspTotalInboundSAs: 2
secEspCurrentOutboundSA 1      ipsecEspTotalOutboundSAs: 2
                               IPsecrity Packet Statistics
secEspTotalInPkts:          19      ipsecEspTotalInPktsDrop: 0
secEspTotalOutPkts:         83
                               IPsecrity Error Statistics
secAuthenticationErrors 0
secReplayErrors:          0          ipsecPolicyErrors:          13
secOtherReceiveErrors:    0          ipsecSendErrors:          0
secAuthenticationErrors 0
secReplayErrors:          0          ipsecPolicyErrors:          13
secOtherReceiveErrors:    0          ipsecSendErrors:          0
secUnknownSpiErrors:      0

```

Syntax: `show ipsec statistics`

This command takes no parameters.

Displaying IPsec configuration for an area

The `show ipv6 ospf area [<area-id>]` command includes information about IPsec for one area or all areas. In the example that follows, the IPsec information is in bold. IPsec is enabled in the first area (area 0) in this example but not in area 3. Note that in area 3, the IPsec key was specified as not encrypted.

```
Brocade(config-ospf6-router)#show ipv6 ospf area
Authentication: Configured
KeyRolloverTime(sec): Configured: 25 Current: 20
KeyRolloverState: Active,Phase1
Current: None
New: SPI:400, ESP, SHA1
Key:$Z|83OmYW{QZ|83OmYW{QZ|83OmYW{QZ|83OmYW{Q
Interface attached to this area: eth 1/1/1
Number of Area scoped LSAs is 6
Sum of Area LSAs Checksum is 0004f7de
Statistics of Area 0:
    SPF algorithm executed 6 times
    SPF last updated: 482 sec ago
    Current SPF node count: 1
        Router: 1 Network: 0
        Maximum of Hop count to nodes: 0
Area 3:
Authentication: Not Configured
Interface attached to this area:
Number of Area scoped LSAs is 3
```

Syntax: `show ipv6 ospf area [<area-id>]`

The `<area-id>` parameter restricts the display to the specified OSPF area. You can specify the `<area-id>` parameter in the following formats:

- An IPv4 address, for example, 192.168.1.1
- A numerical value in the range 0 – 2,147,483,647

TABLE 236 Area configuration of IPsec

Field	Description
Authentication	This field shows whether or not authentication is configured. If this field says "Not Configured," the IPsec-related fields (bold in example screen output) are not displayed at all.
KeyRolloverTime	The number of seconds between each initiation of a key rollover. This field shows the configured and current times.
KeyRolloverState	Can be: Not active: key rollover is not active> Active phase 1: rollover is in its first interval. Active phase 2: rollover is in its second interval.
Current	Shows current SPI, authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the current key.

TABLE 236 Area configuration of IPsec (Continued)

Field	Description
New	Shows new SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the new key.
Old	Shows old SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the old key.

Displaying IPsec for an interface

To see IPsec configuration for a particular interface or all interfaces, use the **show ipv6 ospf interface** command as in the following example (IPsec information in bold).

```

Brocade#show ipv6 ospf interface
eth 1/1/3 is down, type BROADCAST
  Interface is disabled

eth 1/1/8 is up, type BROADCAST
  IPv6 Address:
    2100:18:18:18:18::1/64
    2100:18:18:18:18::/64
  Instance ID 255, Router ID 1.1.1.1
  Area ID 1, Cost 1
  State BDR, Transmit Delay 1 sec, Priority 1
  Timer intervals :
    Hello 10, Hello Jitter 10  Dead 40, Retransmit 5
Authentication: Enabled
KeyRolloverTime(sec): Configured: 30 Current: 0
KeyRolloverState: NotActive
Outbound: SPI:121212, ESP, SHA1
Key:1234567890123456789012345678901234567890
Inbound: SPI:121212, ESP, SHA1
Key:1234567890123456789012345678901234567890
DR:2.2.2.2 BDR:1.1.1.1 Number of I/F scoped LSAs is 2
DRElection: 1 times, DelayedLSAck: 83 times
Neighbor Count = 1, Adjacent Neighbor Count= 1
  Neighbor:
    2.2.2.2 (DR)
  Statistics of interface eth 1/1/8:
    Type      tx      rx      tx-byte  rx-byte
    Unknown   0        0        0        0
    Hello    1415      1408     56592    56320
    DbDesc     3         3        804      804
    LSReq      1         1        28       28
    LSUpdate  193       121     15616    9720
    LSAck     85       109     4840     4924
OSPF messages dropped,no authentication: 0

Syntax: show ipv6 ospf interface [ethernet <slot/port> | pos <slot/port> | loopback <number> |
      tunnel <number> | ve <number>]

```


TABLE 237 Area configuration of IPsec

Field	Description
Authentication	This field shows whether or not authentication is configured. If this field says "Not Configured," the IPsec-related fields (bold in example screen output) are not displayed at all.
KeyRolloverTime	The number of seconds between each initiation of a key rollover. This field shows the configured and current times.
KeyRolloverState	Can be: Not active: key rollover is not active> Active phase 1: rollover is in its first interval. Active phase 2: rollover is in its second interval.
Current	Shows current SPI, authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the current key.
New (Inbound or Outbound)	Shows new SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the new key.
Old (Inbound or Outbound)	Shows old SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the old key.
OSPF messages dropped	Shows the number of packets dropped because the packets failed authentication (for any reason).

Displaying IPsec for a virtual link

To display IPsec for a virtual link, run the **show ipv6 ospf virtual-link brief** or **show ipv6 ospf virtual-link** command, as the following examples illustrate.

```

Brocade#show ipv6 ospf virtual-link brief
Index Transit Area ID Router ID Interface Address State
1 1 14.14.14.14 3000:1:1:1::1 P2P

Brocade#show ipv6 ospf virtual-link
Transit Area ID Router ID Interface Address State
1 14.14.14.14 3000:1:1:1::1 P2P
Timer intervals(sec) :
Hello 10, Hello Jitter 10, Dead 40, Retransmit 5, TransmitDelay 1
DelayedLSAck: 5 times
Authentication: Configured
KeyRolloverTime(sec): Configured: 10 Current: 0
KeyRolloverState: NotActive
Outbound: SPI:100004, ESP, SHA1
Key:1234567890123456789012345678901234567890
Inbound: SPI:100004, ESP, SHA1
Key:1234567890123456789012345678901234567890
Statistics:
Type tx rx tx-byte rx-byte
Unknown 0 0 0 0
Hello 65 65 2600 2596
DbDesc 4 4 2752 2992
LSReq 1 1 232 64
LSUpdate 11 5 1040 1112
LSAck 5 8 560 448
OSPF messages dropped,no authentication: 0
Neighbor: State: Full Address: 2004:44:44:44::4 Interface: eth 2/2

```

Syntax: `show ipv6 ospf virtual-link [brief]`

The optional [brief] keyword limits the display to the Transit, Area ID, Router ID, Interface Address, and State fields for each link.

Changing a key

In this example, the key is changed as illustrated in the two command lines that follow. Note that the SPI value is changed from 300 to 310 to comply with the requirement that you change the SPI when you change the key.

Initial configuration command.

```
Brocade(config-if-e10000-1/1/3)#ipv6 ospf auth ipsec spi 300 esp sha1  
no-encrypt 12345678900987655431234567890aabbccdddef
```

Command line for changing the key.

```
Brocade(config-if-e10000-1/1/3)#ipv6 ospf auth ipsec spi 310 esp sha1  
no-encrypt 989898989009876554321234567890aabbccdddef
```

BGP (IPv4)

[Table 238](#) lists individual Brocade switches and the Border Gateway Protocol (BGP4) features they support. BGP4 features are supported on FastIron X Series and Brocade FCX Series-ADV devices running the full Layer 3 software image.

NOTE

If the Brocade FCX Series device does not have a BGP license, you cannot configure BGP with the "router bgp" command at all. For details, see the chapter [“Software-based Licensing”](#) on page 199.

TABLE 238 Supported BGP4 features

Feature	FESX FSX 800 FSX 1600	FWS	FCX (-ADV models only)	ICX 6610	ICX 6450
BGP4	Yes	No	Yes	Yes	No
BGP Software License	No	No	Yes	Yes	No
BGP4 graceful restart	Yes (FSX 800 and FSX 1600 only)	No	Yes (FCX stack only)	Yes (ICX 6610 stack only)	No
BGP4 peer group	Yes	No	Yes	Yes	No
Route redistribution	Yes	No	Yes	Yes	No
Route aggregation	Yes	No	Yes	Yes	No
BGP null0 routing	Yes	No	Yes	Yes	No
Route reflection	Yes	No	Yes	Yes	No
BGP filters	Yes	No	Yes	Yes	No
Cooperative BGP4 route filtering	Yes	No	Yes	Yes	No
Route flap dampening	Yes	No	Yes	Yes	No
Multipath load sharing	Yes	No	Yes	Yes	No
Traps for BGP4	Yes	No	Yes	Yes	No

This chapter provides details on how to configure Border Gateway Protocol version 4 (BGP4) on Brocade products using the CLI.

BGP4 is described in RFC 1771. The Brocade implementation fully complies with RFC 1771. The Brocade BGP4 implementation also supports the following RFCs:

- RFC 1745 (OSPF Interactions)
- RFC 1997 (BGP Communities Attributes)
- RFC 2385 (TCP MD5 Signature Option)
- RFC 2439 (Route Flap Dampening)
- RFC 2796 (Route Reflection)

- RFC 2842 (Capability Advertisement)
- RFC 3065 (BGP4 Confederations)

To display BGP4 configuration information and statistics, refer to [“Displaying BGP4 information”](#) on page 1421.

NOTE

Your Layer 3 switch management module must have 32 MB or higher to run BGP4.

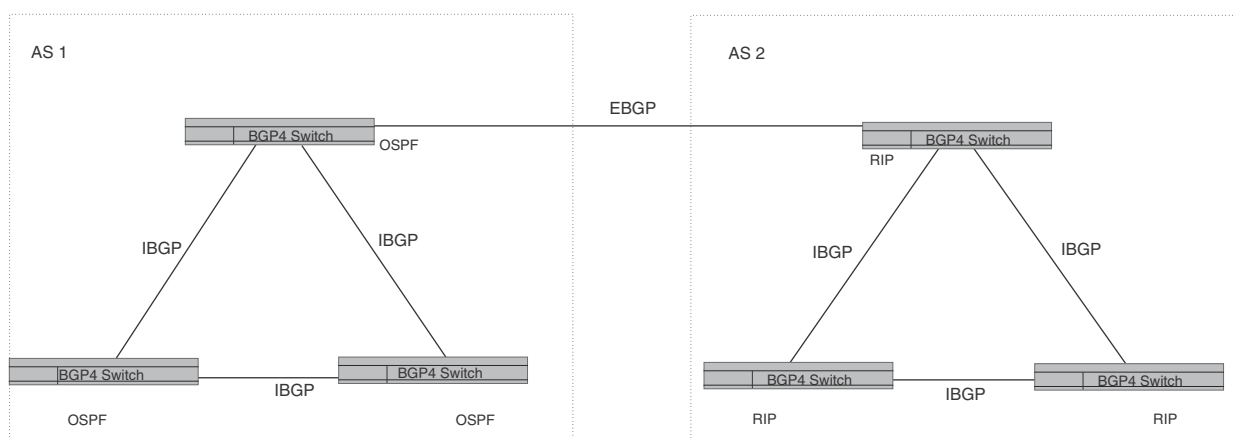
BGP4 overview

Border Gateway Protocol 4 (BGP4) is the standard Exterior Gateway Protocol (EGP) used on the Internet to route traffic between Autonomous Systems (AS) and to maintain loop-free routing. An autonomous system is a collection of networks that share the same routing and administration characteristics. For example, a corporate intranet consisting of several networks under common administrative control might be considered an AS. The networks in an AS can but do not need to run the same routing protocol to be in the same AS, nor do they need to be geographically close.

Routers within an AS can use different Interior Gateway Protocols (IGPs) such as RIP and OSPF to communicate with one another. However, for routers in different autonomous systems to communicate, they need to use an EGP. BGP4 is the standard EGP used by Internet routers and therefore is the EGP implemented on Brocade Layer 3 switches.

[Figure 166](#) on page 1342 shows a simple example of two BGP4 autonomous systems. Each AS contains three BGP4 switches. All of the BGP4 switches within an AS communicate using IBGP. BGP4 switches communicate with other autonomous systems using EBGP. Notice that each of the switches also is running an Interior Gateway Protocol (IGP). The switches in AS1 are running OSPF and the switches in AS2 are running RIP. Brocade Layer 3 switches can be configured to redistribute routes among BGP4, RIP, and OSPF. They also can redistribute static routes.

FIGURE 166 Example BGP4 autonomous systems



Relationship between the BGP4 route table and the IP route table

The Brocade Layer 3 switch BGP4 route table can have multiple routes to the same destination, which are learned from different BGP4 neighbors. A BGP4 neighbor is another switch that also is running BGP4. BGP4 neighbors communicate using Transmission Control Protocol (TCP) port 179 for BGP communication. When you configure the Brocade Layer 3 switch for BGP4, one of the configuration tasks you perform is to identify the Layer 3 switch BGP4 neighbors.

Although a Layer 3 Switch BGP4 route table can have multiple routes to the same destination, the BGP4 protocol evaluates the routes and chooses only one of the routes to send to the IP route table. The route that BGP4 chooses and sends to the IP route table is the preferred route and will be used by the Brocade Layer 3 switch. If the preferred route goes down, BGP4 updates the route information in the IP route table with a new BGP4 preferred route.

NOTE

If IP load sharing is enabled and you enable multiple equal-cost paths for BGP4, BGP4 can select more than one equal-cost path to a destination.

A BGP4 route consists of the following information:

- Network number (prefix) – A value comprised of the network mask bits and an IP address (*<IP address>/ <mask bits>*); for example, 192.215.129.0/18 indicates a network mask of 18 bits applied to the IP address 192.215.129.0. When a BGP4 Layer 3 switch advertises a route to one of its neighbors, the route is expressed in this format.
 - AS-path – A list of the other autonomous systems through which a route passes. BGP4 routers can use the AS-path to detect and eliminate routing loops. For example, if a route received by a BGP4 router contains the AS that the router is in, the router does not add the route to its own BGP4 table. (The BGP4 RFCs refer to the AS-path as “AS_PATH”.)
 - Additional path attributes – A list of additional parameters that describe the route. The route origin and next hop are examples of these additional path attributes.
-

NOTE

The Layer 3 switch re-advertises a learned best BGP4 route to the Layer 3 switch neighbors even when the software does not select that route for installation in the IP route table. The best BGP4 route is the route that the software selects based on comparison of the BGP4 route path attributes.

After a Brocade Layer 3 switch successfully negotiates a BGP4 session with a neighbor (a BGP4 peer), the Brocade Layer 3 switch exchanges complete BGP4 route tables with the neighbor. After this initial exchange, the Brocade Layer 3 switch and all other RFC 1771-compliant BGP4 routers send UPDATE messages to inform neighbors of new, changed, or no longer feasible routes. BGP4 routers do not send regular updates. However, if configured to do so, a BGP4 router does regularly send KEEPALIVE messages to its peers to maintain BGP4 sessions with them if the router does not have any route information to send in an UPDATE message. Refer to [“BGP4 message types”](#) on page 1345 for information about BGP4 messages.

How BGP4 selects a path for a route

When multiple paths for the same route are known to a BGP4 router, the router uses the following algorithm to weigh the paths and determine the optimal path for the route. The optimal path depends on various parameters, which can be modified. (Refer to [“Optional BGP4 configuration tasks”](#) on page 1365.)

1. Is the next hop accessible through an Interior Gateway Protocol (IGP) route? If not, ignore the route.

NOTE

The device does not use the default route to resolve BGP4 next hop. Also refer to [“Enabling next-hop recursion”](#) on page 1371.

2. Use the path with the largest weight.
3. If the weights are the same, prefer the route with the largest local preference.
4. If the routes have the same local preference, prefer the route that was originated locally (by this BGP4 Layer 3 switch).
5. If the local preferences are the same, prefer the route with the shortest AS-path. An AS-SET counts as 1. A confederation path length, if present, is not counted as part of the path length.
6. If the AS-path lengths are the same, prefer the route with the lowest origin type. From low to high, route origin types are valued as follows:
 - IGP is lowest
 - EGP is higher than IGP but lower than INCOMPLETE
 - INCOMPLETE is highest
7. If the routes have the same origin type, prefer the route with the lowest MED. For a definition of MED, refer to [“Configuring the Layer 3 switch to always compare Multi-Exit Discriminators \(MEDs\)”](#) on page 1377.

BGP4 compares the MEDs of two otherwise equivalent paths if and only if the routes were learned from the same neighboring AS. This behavior is called deterministic MED.

Deterministic MED is always enabled and cannot be disabled. In addition, you can enable the Layer 3 switch to always compare the MEDs, regardless of the AS information in the paths. To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.

NOTE

By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the Layer 3 switch favoring the route paths that are missing their MEDs. You can use the **med-missing-as-worst** command to make the Layer 3 switch regard a BGP route with a missing MED attribute as the least favorable route, when comparing the MEDs of the routes.

NOTE

MED comparison is not performed for internal routes originated within the local AS or confederation.

8. Prefer routes in the following order:
 - Routes received through EBGp from a BGP4 neighbor outside of the confederation
 - Routes received through EBGp from a BGP4 router within the confederation
 - Routes received through IBGP
9. If all the comparisons above are equal, prefer the route with the lowest IGP metric to the BGP4 next hop. This is the closest internal path inside the AS to reach the destination.
10. If the internal paths also are the same and BGP4 load sharing is enabled, load share among the paths. Otherwise, prefer the route that comes from the BGP4 router with the lowest router ID.

NOTE

Brocade Layer 3 switches support BGP4 load sharing among multiple equal-cost paths. BGP4 load sharing enables the Layer 3 switch to balance the traffic across the multiple paths instead of choosing just one path based on router ID. For EBGp routes, load sharing applies only when the paths are from neighbors within the same remote AS. EBGp paths from neighbors in different autonomous systems are not compared.

BGP4 message types

BGP4 routers communicate with their neighbors (other BGP4 routers) using the following types of messages:

- OPEN
- UPDATE
- KEEPALIVE
- NOTIFICATION

OPEN messages exchanged with BGP4 routers

After a BGP4 router establishes a TCP connection with a neighboring BGP4 router, the routers exchange OPEN messages. An OPEN message indicates the following:

- **BGP version** – Indicates the version of the protocol that is in use on the router. BGP version 4 supports Classless Interdomain Routing (CIDR) and is the version most widely used in the Internet. Version 4 also is the only version supported on Brocade Layer 3 switches.
- **AS number** – A two-byte number that identifies the AS to which the BGP4 router belongs.
- **Hold Time** – The number of seconds a BGP4 router will wait for an UPDATE or KEEPALIVE message (described below) from a BGP4 neighbor before assuming that the neighbor is dead. BGP4 routers exchange UPDATE and KEEPALIVE messages to update route information and maintain communication. If BGP4 neighbors are using different Hold Times, the lowest Hold Time is used by the neighbors. If the Hold Time expires, the BGP4 router closes its TCP connection to the neighbor and clears any information it has learned from the neighbor and cached.

You can configure the Hold Time to be 0, in which case a BGP4 router will consider its

neighbors to always be up. For directly-attached neighbors, you can configure the Brocade Layer 3 switch to immediately close the TCP connection to the neighbor and clear entries learned from an EBGp neighbor if the interface to that neighbor goes down. This capability is provided by the fast external fallover feature, which is disabled by default.

- **BGP Identifier** – The router ID. The BGP Identifier (router ID) identifies the BGP4 router to other BGP4 routers. Brocade Layer 3 switches use the same router ID for OSPF and BGP4. If you do not set a router ID, the software uses the IP address on the lowest numbered loopback interface configured on the router. If the Layer 3 switch does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, refer to [“Changing the router ID”](#) on page 976.
- **Parameter list** – An optional list of additional parameters used in peer negotiation with BGP4 neighbors.

UPDATE messages from BGP4 routers

After BGP4 neighbors establish a BGP4 connection over TCP and exchange their BGP4 routing tables, they do not send periodic routing updates. Instead, a BGP4 neighbor sends an update to its neighbor when it has a new route to advertise or routes have changed or become unfeasible. An UPDATE message can contain the following information:

- **Network Layer Reachability Information (NLRI)** – The mechanism by which BGP4 supports Classless Interdomain Routing (CIDR). An NLRI entry consists of an IP prefix that indicates a network being advertised by the UPDATE message. The prefix consists of an IP network number and the length of the network portion of the number. For example, an UPDATE message with the NLRI entry 192.215.129.0/18 indicates a route to IP network 192.215.129.0 with network mask 255.255.192.0. The binary equivalent of this mask is 18 consecutive one bits, thus “18” in the NLRI entry.
- **Path attributes** – Parameters that indicate route-specific information such as path information, route preference, next hop values, and aggregation information. BGP4 uses the path attributes to make filtering and routing decisions.
- **Unreachable routes** – A list of routes that have been in the sending router BGP4 table but are no longer feasible. The UPDATE message lists unreachable routes in the same format as new routes.

KEEPALIVE messages from BGP4 routers

BGP4 routers do not regularly exchange UPDATE messages to maintain the BGP4 sessions. For example, if a Layer 3 switch configured to perform BGP4 routing has already sent the latest route information to its peers in UPDATE messages, the router does not send more UPDATE messages. Instead, BGP4 routers send KEEPALIVE messages to maintain the BGP4 sessions. KEEPALIVE messages are 19 bytes long and consist only of a message header; they contain no routing data.

BGP4 routers send KEEPALIVE messages at a regular interval, the Keep Alive Time. The default Keep Alive Time on Brocade Layer 3 switches is 60 seconds.

A parameter related to the Keep Alive Time is the Hold Time. A BGP4 router Hold Time determines how many seconds the router will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead. The Hold Time is negotiated when BGP4 routers exchange OPEN messages; the lower Hold Time is then used by both neighbors. For example, if

BGP4 Router A sends a Hold Time of 5 seconds and BGP4 Router B sends a Hold Time of 4 seconds, both routers use 4 seconds as the Hold Time for their BGP4 session. The default Hold Time is 180 seconds. Generally, the Hold Time is configured to three times the value of the Keep Alive Time.

If the Hold Time is 0, a BGP4 router assumes that its neighbor is alive regardless of how many seconds pass between receipt of UPDATE or KEEPALIVE messages.

NOTIFICATION messages from BGP4 routers

When you close the router BGP4 session with a neighbor, or the router detects an error in a message received from the neighbor, or an error occurs on the router, the router sends a NOTIFICATION message to the neighbor. No further communication takes place between the BGP4 router that sent the NOTIFICATION and the neighbors that received the NOTIFICATION.

BGP4 graceful restart

BGP4 graceful restart is a high-availability routing feature that minimizes disruption in traffic forwarding, diminishes route flapping, and provides continuous service during a system restart, switchover, failover, or hitless OS upgrade. During such events, routes remain available between devices. BGP4 graceful restart operates between a device and its peers, and must be configured on each participating device.

Under normal operation, when a BGP4 device is restarted, the network is automatically reconfigured. Routes available through the restarting device are deleted when the device goes down, and are then rediscovered and added back to the routing tables when the device is back up and running. In a network with devices that are regularly restarted, performance can degrade significantly and the availability of network resources can be limited.

BGP4 graceful restart is enabled globally by default. A BGP4 graceful restart-enabled device advertises this capability to establish peering relationships with other devices. When a restart begins, neighbor devices mark all of the routes from the restarting device as stale, but continue to use the routes for the length of time specified by the restart timer. After the device is restarted, it begins to receive routing updates from the peers. When it receives the end-of-RIB marker that indicates it has received all of the BGP4 route updates, it recomputes the new routes and replaces the stale routes in the route map with the newly computed routes. If the device does not come back up within the time configured for the purge timer, the stale routes are removed.

NOTE

BGP4 graceful restart is supported in FSX 800 and FSX 1600 Layer 3 switches with dual management modules as well as in FCX switches in a stack. If the switch will function as a restart helper device only, a secondary management module is not required.

This implementation of BGP4 graceful restart supports the Internet Draft-ietf-idr-restart-10.txt: restart mechanism for BGP4

For details concerning configuration of BGP4 graceful restart, refer to [“Configuring BGP4 graceful restart”](#) on page 1385.

Basic configuration and activation for BGP4

BGP4 is disabled by default. Follow the steps below to enable BGP4 and place your Brocade Layer 3 switch into service as a BGP4 router.

1. Enable the BGP4 protocol.
2. Set the local AS number.

NOTE

You must specify the local AS number for BGP4 to become functional.

3. Add each BGP4 neighbor (peer BGP4 router) and identify the AS the neighbor is in.
4. Save the BGP4 configuration information to the system configuration file.

NOTE

By default, the Brocade router ID is the IP address configured on the lowest numbered loopback interface. If the Layer 3 switch does not have a loopback interface, the default router ID is the lowest numbered IP interface address configured on the device. For more information or to change the router ID, refer to [“Changing the router ID”](#) on page 976. If you change the router ID, all current BGP4 sessions are cleared.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
Brocade(config-bgp-router)#local-as 10
Brocade(config-bgp-router)#neighbor 209.157.23.99 remote-as 100
Brocade(config-bgp-router)#write memory
```

NOTE

When BGP4 is enabled on a Brocade Layer 3 switch, you do not need to reset the system. The protocol is activated as soon as you enable it. Moreover, the router begins a BGP4 session with a BGP4 neighbor as soon as you add the neighbor.

Note regarding disabling BGP4

If you disable BGP4, the Layer 3 switch removes all the running configuration information for the disabled protocol from the running-config. To restore the BGP4 configuration, you must reload the software to load the configuration from the startup-config. Moreover, when you save the configuration to the startup-config file after disabling the protocol, all the configuration information for the disabled protocol is removed from the startup-config file.

The CLI displays a warning message such as the following.

```
Brocade(config-bgp-router)#no router bgp
router bgp mode now disabled. All bgp config data will be lost when writing to
flash!
```

If you are testing a BGP4 configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup-config file containing the protocol configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

NOTE

To disable BGP4 without losing the BGP4 configuration information, remove the local AS (for example, by entering the **no local-as <num>** command). In this case, BGP4 retains the other configuration information but is not operational until you set the local AS again.

BGP4 parameters

You can modify or set the following BGP4 parameters:

- Optional – Define the router ID. (The same router ID also is used by OSPF.)
- Required – Specify the local AS number.
- Optional – Add a loopback interface for use with neighbors.
- Required – Identify BGP4 neighbors.
- Optional – Change the Keep Alive Time and Hold Time.
- Optional – Change the update timer for route changes.
- Optional – Enable fast external fallover.
- Optional – Specify a list of individual networks in the local AS to be advertised to remote autonomous systems using BGP4.
- Optional – Change the default local preference for routes.
- Optional – Enable the default route (default-information-originate).
- Optional – Enable use of a default route to resolve a BGP4 next-hop route.
- Optional – Change the default MED (metric).
- Optional – Enable next-hop recursion.
- Optional – Change the default administrative distances for EBGp, IBGP, and locally originated routes.
- Optional – Require the first AS in an Update from an EBGp neighbor to be the neighbor AS.
- Optional – Change MED comparison parameters.
- Optional – Disable comparison of the AS-Path length.
- Optional – Enable comparison of the router ID.
- Optional – Enable auto summary to summarize routes at an IP class boundary (A, B, or C).
- Optional – Aggregate routes in the BGP4 route table into CIDR blocks.
- Optional – Configure the router as a BGP4 router reflector.
- Optional – Configure the Layer 3 switch as a member of a BGP4 confederation.
- Optional – Change the default metric for routes that BGP4 redistributes into RIP or OSPF.
- Optional – Change the parameters for RIP, OSPF, or static routes redistributed into BGP4.
- Optional – Change the number of paths for BGP4 load sharing.
- Optional – Change other load-sharing parameters
- Optional – Define BGP4 address filters.
- Optional – Define BGP4 AS-path filters.
- Optional – Define BGP4 community filters.
- Optional – Define IP prefix lists.

- Optional – Define neighbor distribute lists.
- Optional – Define BGP4 route maps for filtering routes redistributed into RIP and OSPF.
- Optional – Define route flap dampening parameters.

NOTE

When using the CLI, you set global level parameters at the BGP CONFIG level of the CLI. You can reach the BGP CONFIG level by entering **router bgp...** at the global CONFIG level.

BGP4 parameter changes

Some parameter changes take effect immediately while others do not take full effect until the router sessions with its neighbors are reset. Some parameters do not take effect until the router is rebooted.

Parameter changes that take effect immediately

- Enable or disable BGP.
- Set or change the local AS.
- Add neighbors.
- Change the update timer for route changes.
- Disable or enable fast external fallover.
- Specify individual networks that can be advertised.
- Change the default local preference, default information originate setting, or administrative distance.
- Enable or disable use of a default route to resolve a BGP4 next-hop route.
- Enable or disable MED (metric) comparison.
- Require the first AS in an Update from an EBGp neighbor to be the neighbor AS.
- Change MED comparison parameters.
- Disable comparison of the AS-Path length.
- Enable comparison of the router ID.
- Enable next-hop recursion.
- Enable or disable auto summary.
- Change the default metric.
- Disable or re-enable route reflection.
- Configure confederation parameters.
- Disable or re-enable load sharing.
- Change the maximum number of load-sharing paths.
- Change other load-sharing parameters.
- Define route flap dampening parameters.
- Add, change, or negate redistribution parameters (except changing the default MED; see below).
- Add, change, or negate route maps (when used by the **network** command or a redistribution command).

BGP4 parameter changes after resetting neighbor sessions

The following parameter changes take effect only after the router BGP4 sessions are cleared, or reset using the “soft” clear option. (Refer to [“Closing or resetting a neighbor session”](#) on page 1457.)

The parameter are as follows:

- Change the Hold Time or Keep Alive Time.
- Aggregate routes.
- Add, change, or negate filter tables.

BGP4 parameter changes after disabling and re-enabling redistribution

The following parameter change takes effect only after you disable and then re-enable redistribution:

- Change the default MED (metric).

BGP4 memory considerations

BGP4 handles a very large number of routes and therefore requires a lot of memory. For example, in a typical configuration with just a single BGP4 neighbor, a BGP4 router may need to be able to hold up to 80,000 routes. Many configurations, especially those involving more than one neighbor, can require the router to hold even more routes. Brocade Layer 3 switches provide dynamic memory allocation for BGP4 data. These devices automatically allocate memory when needed to support BGP4 neighbors, routes, and route attribute entries. Dynamic memory allocation is performed automatically by the software and does not require a reload.

[Table 239](#) lists the maximum total amount of system memory (DRAM) BGP4 can use. The maximum depends on the total amount of system memory on the device.

TABLE 239 Maximum memory usage

Platform	Maximum memory BGP4 can use
FESX with 128 MB	30 MB
FSX with Management 1 module with 256 MB	130 MB
FSX with Management 2 module with 512 MB	400 MB

The memory amounts listed in the table are for all BGP4 data, including routes received from neighbors, BGP route advertisements (routes sent to neighbors), and BGP route attribute entries. The routes sent to and received from neighbors use the most BGP4 memory. Generally, the actual limit to the number of neighbors, routes, or route attribute entries the device can accommodate depends on how many routes the Layer 3 switch sends to and receives from the neighbors.

In some cases, where most of the neighbors do not send or receive a full BGP route table (about 80,000 routes), the memory can support a larger number of BGP4 neighbors. However, if most of the BGP4 neighbors send or receive full BGP route tables, the number of BGP neighbors the memory can support is less than in configurations where the neighbors send smaller route tables.

As a guideline, Layer 3 switches with a 512 MB Management 4 module can accommodate 150 through 200 neighbors, with the assumption that the Layer 3 switch receives about one million routes total from all neighbors and sends about eight million routes total to neighbors. For each additional one million incoming routes, the capacity for outgoing routes decreases by around two million.

Memory configuration options obsoleted by dynamic memory

Devices that support dynamic BGP4 memory allocation do not require or even support static configuration of memory for BGP4 neighbors, routes, or route attributes. Consequently, the following CLI commands and equivalent Web management options are not supported on these devices:

- **max-neighbors** *<num>*
- **max-routes** *<num>*
- **max-attribute-entries** *<num>*

If you boot a device that has a startup-config file that contains these commands, the software ignores the commands and uses dynamic memory allocation for BGP4. The first time you save the device running configuration (running-config) to the startup-config file, the commands are removed from the file.

Basic configuration tasks required for BGP4

The following sections describe how to perform the configuration tasks that are required to use BGP4 on the Brocade Layer 3 switch. You can modify many parameters in addition to the ones described in this section. Refer to [“Optional BGP4 configuration tasks”](#) on page 1365.

Enabling BGP4 on the router

When you enable BGP4 on the router, BGP4 is automatically activated. To enable BGP4 on the router, enter the following commands.

```
Brocade> enable
Brocade#configure terminal
Brocade(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
Brocade(config-bgp-router)#local-as 10
Brocade(config-bgp-router)#neighbor 209.157.23.99 remote-as 100
Brocade(config-bgp-router)#write memory
```

Changing the router ID

The OSPF and BGP4 protocols use router IDs to identify the routers that are running the protocols. A router ID is a valid, unique IP address and sometimes is an IP address configured on the router. The router ID cannot be an IP address in use by another device.

By default, the router ID on a Brocade Layer 3 switch is one of the following:

- If the router has loopback interfaces, the default router ID is the IP address configured on the lowest numbered loopback interface configured on the Layer 3 switch. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default router ID is 9.9.9.9/24:
 - Loopback interface 1, 9.9.9.9/24
 - Loopback interface 2, 4.4.4.4/24
 - Loopback interface 3, 1.1.1.1/24
- If the device does not have any loopback interfaces, the default router ID is the lowest numbered IP interface address configured on the device.

NOTE

Brocade Layer 3 switches use the same router ID for both OSPF and BGP4. If the router is already configured for OSPF, you may want to use the router ID that is already in use on the router rather than set a new one. To display the router ID, enter the **show ip** CLI command at any CLI level.

To change the router ID, enter a command such as the following.

```
Brocade(config)#ip router-id 209.157.22.26
```

Syntax: **ip router-id** <ip-addr>

The <ip-addr> can be any valid, unique IP address.

NOTE

You can specify an IP address used for an interface on the Brocade Layer 3 switch, but do not specify an IP address in use by another device.

Setting the local AS number

The local AS number identifies the AS the Brocade BGP4 router is in. The AS number can be from 1 through 65535. There is no default. AS numbers 64512 through 65535 are the well-known private BGP4 AS numbers and are not advertised to the Internet community.

To set the local AS number, enter commands such as the following.

```
Brocade(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
Brocade(config-bgp-router)#local-as 10
Brocade(config-bgp-router)#write memory
```

Syntax: **[no] local-as** <num>

The <num> parameter specifies the local AS number.

Adding a loopback interface

You can configure the router to use a loopback interface instead of a specific port or virtual routing interface to communicate with a BGP4 neighbor. A loopback interface adds stability to the network by working around route flap problems that can occur due to unstable links between the router and its neighbors.

Loopback interfaces are always up, regardless of the states of physical interfaces. Loopback interfaces are especially useful for IBGP neighbors (neighbors in the same AS) that are multiple hops away from the router. When you configure a BGP4 neighbor on the router, you can specify whether the router uses the loopback interface to communicate with the neighbor. As long as a path exists between the router and its neighbor, BGP4 information can be exchanged. The BGP4 session is not associated with a specific link but instead is associated with the virtual interfaces.

You can add up to 24 IP addresses to each loopback interface.

NOTE

If you configure the Brocade Layer 3 switch to use a loopback interface to communicate with a BGP4 neighbor, the peer IP address on the remote router pointing to your loopback address must be configured.

To add a loopback interface, enter commands such as those shown in the following example.

```
Brocade(config-bgp-router)#exit
Brocade(config)#int loopback 1
Brocade(config-lbif-1)#ip address 10.0.0.1/24
```

Syntax: `interface loopback <num>`

The <num> value can be from 1 through 8 on Chassis Layer 3 switches. The value can be from 1 through 4 on the Compact Layer 3 switch.

Adding BGP4 neighbors

The BGP4 protocol does not contain a peer discovery process. Therefore, for each of the router BGP4 neighbors (peers), you must indicate the neighbor IP address and the AS each neighbor is in. Neighbors that are in different autonomous systems communicate using EBGP. Neighbors within the same AS communicate using IBGP.

NOTE

If the Layer 3 switch has multiple neighbors with similar attributes, you can simplify configuration by configuring a peer group, then adding individual neighbors to it. The configuration steps are similar, except you specify a peer group name instead of a neighbor IP address when configuring the neighbor parameters, then add individual neighbors to the peer group. Refer to [“Adding a BGP4 peer group”](#) on page 1361.

NOTE

The Layer 3 switch attempts to establish a BGP4 session with a neighbor as soon as you enter a command specifying the neighbor IP address. If you want to completely configure the neighbor parameters before the Layer 3 switch establishes a session with the neighbor, you can administratively shut down the neighbor. Refer to [“Administratively shutting down a session with a BGP4 neighbor”](#) on page 1364.

To add a BGP4 neighbor with IP address 209.157.22.26, enter the following command.

```
Brocade(config-bgp-router)#neighbor 209.157.22.26
```

The neighbor <ip-addr> must be a valid IP address.

The **neighbor** command has some additional parameters, as shown in the following syntax:

Syntax: `[no] neighbor <ip-addr> | <peer-group-name>
[advertisement-interval <num>]`


```

[capability orf prefixlist [send | receive]]
[default-originate [route-map <map-name>]]
[description <string>]
[distribute-list in | out <num,num,...> | <ACL-num> in | out]
[ebgp-multihop [<num>]]
[filter-list in | out <num,num,...> | <ACL-num> in | out | weight]
[maximum-prefix <num> [<threshold>] [teardown]]
[next-hop-self]
[nlri multicast | unicast | multicast unicast]
[password [0 | 1] <string>]
[prefix-list <string> in | out]
[remote-as <as-number>]
[remove-private-as]
[route-map in | out <map-name>]
[route-reflector-client]
[send-community]
[soft-reconfiguration inbound]
[shutdown]
[timers keep-alive <num> hold-time <num>]
[unsuppress-map <map-name>]
[update-source <ip-addr> | ethernet <port> | loopback <num> | ve <num>]
[weight <num>]

```

The `<ip-addr> | <peer-group-name>` parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group. Refer to [“Adding a BGP4 peer group”](#) on page 1361.

advertisement-interval `<num>` specifies the minimum delay (in seconds) between messages to the specified neighbor. The default is 30 for EBGp neighbors (neighbors in other autonomous systems). The default is 5 for IBGP neighbors (neighbors in the same AS). The range is 0 through 600.

NOTE

The Layer 3 switch applies the advertisement interval only under certain conditions. The Layer 3 switch does not apply the advertisement interval when sending initial updates to a BGP4 neighbor. As a result, the Layer 3 switch sends the updates one immediately after another, without waiting for the advertisement interval.

capability orf prefixlist `[send | receive]` configures cooperative router filtering. The **send | receive** parameter specifies the support you are enabling:

- **send** – The Layer 3 switch sends the IP prefix lists as Outbound Route Filters (ORFs) to the neighbor.
- **receive** – The Layer 3 switch accepts filters as Outbound Route Filters (ORFs) from the neighbor.

If you do not specify the capability, both capabilities are enabled. The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

For more information, refer to [“Configuring cooperative BGP4 route filtering”](#) on page 1411.

NOTE

The current release supports cooperative filtering only for filters configured using IP prefix lists.

default-originate [**route-map** <map-name>] configures the Layer 3 switch to send the default route 0.0.0.0 to the neighbor. If you use the route-map <map-name> parameter, the route map injects the default route conditionally, based on the match conditions in the route map.

description <string> specifies a name for the neighbor. You can enter an alphanumeric text string up to 80 characters long.

distribute-list in | out <num,num,...> specifies a distribute list to be applied to updates to or from the specified neighbor. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor. The <num,num,...> parameter specifies the list of address-list filters. The router applies the filters in the order in which you list them and stops applying the filters in the distribute list when a match is found.

Alternatively, you can specify **distribute-list** <ACL-num> **in | out** to use an IP ACL instead of a distribute list. In this case, <ACL-num> is an IP ACL.

NOTE

By default, if a route does not match any of the filters, the Layer 3 switch denies the route. To change the default behavior, configure the last filter as “permit any any”.

NOTE

The address filter must already be configured. Refer to [“Specific IP address filtering”](#) on page 1394.

ebgp-multihop [<num>] specifies that the neighbor is more than one hop away and that the session type with the neighbor is thus EBGp-multihop. This option is disabled by default. The <num> parameter specifies the TTL you are adding for the neighbor. You can specify a number from 0 through 255. The default is 0. If you leave the EBGp TTL value set to 0, the software uses the IP TTL value.

filter-list in | out <num,num,...> specifies an AS-path filter list or a list of AS-path ACLs. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor. If you specify in or out, The <num,num,...> parameter specifies the list of AS-path filters. The router applies the filters in the order in which you list them and stops applying the filters in the AS-path filter list when a match is found. The **weight** <num> parameter specifies a weight that the Layer 3 switch applies to routes received from the neighbor that match the AS-path filter or ACL. You can specify a number from 0 through 65535.

Alternatively, you can specify filter-list <ACL-num> **in | out | weight** to use an AS-path ACL instead of an AS-path filter list. In this case, <ACL-num> is an AS-path ACL.

NOTE

By default, if an AS-path does not match any of the filters or ACLs, the Layer 3 switch denies the route. To change the default behavior, configure the last filter or ACL as “permit any any”.

NOTE

The AS-path filter or ACL must already be configured. Refer to [“AS-path filtering”](#) on page 1395.

maximum-prefix <num> specifies the maximum number of IP network prefixes (routes) that can be learned from the specified neighbor or peer group. You can specify a value from 0 through 4294967295. The default is 0 (unlimited):

- The <num> parameter specifies the maximum number. You can specify a value from 0 through 4294967295. The default is 0 (unlimited).

- The **<threshold>** parameter specifies the percentage of the value you specified for the **maximum-prefix <num>**, at which you want the software to generate a Syslog message. You can specify a value from 1 (one percent) to 100 (100 percent). The default is 100.
- The **teardown** parameter tears down the neighbor session if the maximum-prefix limit is exceeded. The session remains shutdown until you clear the prefixes using the **clear ip bgp neighbor all** or **clear ip bgp neighbor <ip-addr>** command, or change the neighbor maximum-prefix configuration. The software also generates a Syslog message.

next-hop-self specifies that the router should list itself as the next hop in updates sent to the specified neighbor. This option is disabled by default.

The **nlri multicast | unicast | multicast** unicast parameter specifies whether the neighbor is a multicast neighbor or a unicast neighbor. Optionally, you also can specify unicast if you want the Layer 3 switch to exchange unicast (BGP4) routes as well as multicast routes with the neighbor. The default is unicast only.

password [0 | 1] <string> specifies an MD5 password for securing sessions between the Layer 3 switch and the neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters, but the first character cannot be a number. If the password contains a number, do not enter a space following the number.

The **0 | 1** parameter is the encryption option, which you can omit (the default) or which can be one of the following:

- **0** – Disables encryption for the authentication string you specify with the command. The password or string is shown as clear text in the output of commands that display neighbor or peer group configuration information.
- **1** – Assumes that the authentication string you enter is the encrypted form, and decrypts the value before using it.

For more information, refer to [“Encryption of BGP4 MD5 authentication keys”](#) on page 1359.

NOTE

If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

prefix-list <string> in | out specifies an IP prefix list. You can use IP prefix lists to control routes to and from the neighbor. IP prefix lists are an alternative method to AS-path filters. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor. You can configure up to 1000 prefix list filters. The filters can use the same prefix list or different prefix lists. To configure an IP prefix list, refer to [“Defining IP prefix lists”](#) on page 1401.

remote-as <as-number> specifies the AS the remote neighbor is in. The **<as-number>** can be a number from 1 through 65535. There is no default.

remove-private-as configures the router to remove private AS numbers from UPDATE messages the router sends to this neighbor. The router will remove AS numbers 64512 through 65535 (the well-known BGP4 private AS numbers) from the AS-path attribute in UPDATE messages the Layer 3 switch sends to the neighbor. This option is disabled by default.

route-map in | out <map-name> specifies a route map the Layer 3 switch will apply to updates sent to or received from the specified neighbor. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor.

NOTE

The route map must already be configured. Refer to [“Defining route maps”](#) on page 1403.

route-reflector-client specifies that this neighbor is a route-reflector client of the router. Use the parameter only if this router is going to be a route reflector. For information, refer to [“Route reflection parameter configuration”](#) on page 1378. This option is disabled by default.

send-community enables sending the community attribute in updates to the specified neighbor. By default, the router does not send the community attribute.

shutdown administratively shuts down the session with this neighbor. Shutting down the session allows you to completely configure the neighbor and save the configuration without actually establishing a session with the neighbor. This option is disabled by default.

soft-reconfiguration inbound enables the soft reconfiguration feature, which stores all the route updates received from the neighbor. If you request a soft reset of inbound routes, the software performs the reset by comparing the policies against the stored route updates, instead of requesting the neighbor BGP4 route table or resetting the session with the neighbor. Refer to [“Using soft reconfiguration”](#) on page 1452.

timers keep-alive <num> **hold-time** <num> overrides the global settings for the Keep Alive Time and Hold Time. For the Keep Alive Time, you can specify from 0 through 65535 seconds. For the Hold Time, you can specify 0 or 3 through 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the router waits indefinitely for messages from a neighbor without concluding that the neighbor is dead. The defaults for these parameters are the currently configured global Keep Alive Time and Hold Time. For more information about these parameters, refer to [“Changing the Keep Alive Time and Hold Time”](#) on page 1365.

unsuppress-map <map-name> removes route dampening from a neighbor routes when those routes have been dampened due to aggregation. Refer to [“Removing route dampening from a neighbor routes suppressed due to aggregation”](#) on page 1418.

update-source <ip-addr> | **ethernet** <port> | **loopback** <num> | **ve** <num> configures the router to communicate with the neighbor through the specified interface. There is no default.

weight <num> specifies a weight the Layer 3 switch will add to routes received from the specified neighbor. BGP4 prefers larger weights over smaller weights. The default weight is 0.

Encryption of BGP4 MD5 authentication keys

When you configure a BGP4 neighbor or neighbor peer group, you can specify an MD5 authentication string for authenticating packets exchanged with the neighbor or peer group of neighbors.

For added security, the software encrypts display of the authentication string by default. The software also provides an optional parameter to disable encryption of the authentication string, on an individual neighbor or peer group basis. By default, the MD5 authentication strings are displayed in encrypted format in the output of the following commands:

- **show running-config** (or **write terminal**)
- **show configuration**
- **show ip bgp config**

When encryption of the authentication string is enabled, the string is encrypted in the CLI regardless of the access level you are using.

If you display the running-config after reloading, the BGP4 commands that specify an authentication string show the string in encrypted form.

In addition, when you save the configuration to the startup-config file, the file contains the new BGP4 command syntax and encrypted passwords or strings.

NOTE

Brocade recommends that you save a copy of the startup-config file for each switch you plan to upgrade.

Encryption example

The following commands configure a BGP4 neighbor and a peer group, and specify MD5 authentication strings (passwords) for authenticating packets exchanged with the neighbor or peer group.

```
Brocade(config-bgp-router)#local-as 2
Brocade(config-bgp-router)#neighbor xyz peer-group
Brocade(config-bgp-router)#neighbor xyz password abc
Brocade(config-bgp-router)#neighbor 10.10.200.102 peer-group xyz
Brocade(config-bgp-router)#neighbor 10.10.200.102 password test
```

Here is how the commands appear when you display the BGP4 configuration commands.

```
Brocade#show ip bgp config
Current BGP configuration:
router bgp
  local-as 2
  neighbor xyz peer-group
  neighbor xyz password 1 $!2d
  neighbor 10.10.200.102 peer-group xyz
  neighbor 10.10.200.102 remote-as 1
  neighbor 10.10.200.102 password 1 $on-o
```

Notice that the software has converted the commands that specify an authentication string into the new syntax (described below), and has encrypted display of the authentication strings.

Command syntax

Since the default behavior does not affect the BGP4 configuration itself but does encrypt display of the authentication string, the CLI does not list the encryption options.

Syntax: `[no] neighbor <ip-addr> | <peer-group-name> password [0 | 1] <string>`

The `<ip-addr> | <peer-group-name>` parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group.

The **password** `<string>` parameter specifies an MD5 authentication string for securing sessions between the Layer 3 switch and the neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters, but the first character cannot be a number. If the password contains a number, do not enter a space following the number.

The **0 | 1** parameter is the encryption option, which you can omit (the default) or which can be one of the following:

- **0** – Disables encryption for the authentication string you specify with the command. The password or string is shown as clear text in the output of commands that display neighbor or peer group configuration information.
- **1** – Assumes that the authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE

If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

Displaying the Authentication String

If you want to display the authentication string, enter the following commands.

```
Brocade(config)#enable password-display
Brocade#show ip bgp neighbors
```

The **enable password-display** command enables display of the authentication string, but only in the output of the **show ip bgp neighbors** command. Display of the string is still encrypted in the startup-config file and running-config. Enter the command at the global CONFIG level of the CLI.

NOTE

The command also displays SNMP community strings in clear text, in the output of the **show snmp server** command.

Adding a BGP4 peer group

A **peer group** is a set of BGP4 neighbors that share common parameters. Peer groups provide the following benefits:

- **Simplified neighbor configuration** – You can configure a set of neighbor parameters and then apply them to multiple neighbors. You do not need to individually configure the common parameters individually on each neighbor.
- **Flash memory conservation** – Using peer groups instead of individually configuring all the parameters for each neighbor requires fewer configuration commands in the startup-config file.

You can perform the following tasks on a peer-group basis:

- Reset neighbor sessions
- Perform soft-outbound resets (the Layer 3 switch updates outgoing route information to neighbors but does not entirely reset the sessions with those neighbors)
- Clear BGP message statistics
- Clear error buffers

Peer group parameters

You can set all neighbor parameters in a peer group. When you add a neighbor to the peer group, the neighbor receives all the parameter settings you set in the group, except parameter values you have explicitly configured for the neighbor. If you do not set a neighbor parameter in the peer group and the parameter also is not set for the individual neighbor, the neighbor uses the default value.

Peer group configuration rules

The following rules apply to peer group configuration:

- You must configure a peer group before you can add neighbors to the peer group.
- If you remove a parameter from a peer group, the value for that parameter is reset to the default for all the neighbors within the peer group, unless you have explicitly set that parameter on individual neighbors. In this case, the value you set on the individual neighbors applies to those neighbors, while the default value applies to neighbors for which you have not explicitly set the value.

NOTE

If you enter a command to remove the remote AS parameter from a peer group, the software checks to ensure that the peer group does not contain any neighbors. If the peer group does contain neighbors, the software does not allow you to remove the remote AS. The software prevents removing the remote AS in this case so that the neighbors in the peer group that are using the remote AS do not lose connectivity to the Layer 3 switch.

- Once you add a neighbor to a peer group, you cannot configure the following outbound parameters (the parameters governing outbound traffic) for the neighbor:
 - Default-information-originate
 - Next-hop-self
 - Outbound route map
 - Outbound filter list

- Outbound distribute list
- Outbound prefix list
- Remote AS, if configured for the peer group
- Remove private AS
- Route reflector client
- Send community
- Timers
- Update source

If you want to change an outbound parameter for an individual neighbor, you must first remove the neighbor from the peer group. In this case, you cannot re-add the neighbor to the same peer group, but you can add the neighbor to a different peer group. All the neighbors within a peer group must have the same values for the outbound parameters. To change an outbound parameter to the same value for all neighbors within a peer group, you can change the parameter on a peer-group basis. In this case, you do not need to remove the neighbors and change the parameter individually for each neighbor.

- If you add an outbound parameter to a peer group, that parameter is automatically applied to all neighbors within the peer group.
- When you add a neighbor to a peer group, the software removes any outbound parameters for that neighbor from the running configuration (running-config). As a result, when you save the configuration to the startup-config file, the file does not contain any outbound parameters for the individual neighbors you have placed in a peer group. The only outbound parameters the startup-config file contains for neighbors within a peer group are the parameters associated with the peer group itself. However, the running-config and the startup-config file can contain individual parameters listed in the previous section as well as the settings for those parameters within a peer group.

You can override neighbor parameters that do not affect outbound policy on an individual neighbor basis.

- If you do not specify a parameter for an individual neighbor, the neighbor uses the value in the peer group.
- If you set the parameter for the individual neighbor, that value overrides the value you set in the peer group.
- If you add a parameter to a peer group that already contains neighbors, the parameter value is applied to neighbors that do not already have the parameter explicitly set. If a neighbor has the parameter explicitly set, the explicitly set value overrides the value you set for the peer group.
- If you remove the setting for a parameter from a peer group, the value for that parameter changes to the default value for all the neighbors in the peer group that do not have that parameter individually set.

Configuring a peer group

To configure a BGP4 peer group, enter commands such as the following at the BGP configuration level.

```
Brocade(config-bgp-router)#neighbor PeerGroup1 peer-group
Brocade(config-bgp-router)#neighbor PeerGroup1 description "EastCoast Neighbors"
Brocade(config-bgp-router)#neighbor PeerGroup1 remote-as 100
Brocade(config-bgp-router)#neighbor PeerGroup1 distribute-list out 1
```


The commands in this example configure a peer group called “PeerGroup1” and set the following parameters for the peer group:

- A description, “EastCoast Neighbors”
- A remote AS number, 100
- A distribute list for outbound traffic

The software applies these parameters to each neighbor you add to the peer group. You can override the description parameter for individual neighbors. If you set the description parameter for an individual neighbor, the description overrides the description configured for the peer group. However, you cannot override the remote AS and distribute list parameters for individual neighbors. Since these parameters control outbound traffic, the parameters must have the same values for all neighbors within the peer group.

Syntax: `neighbor <peer-group-name> peer-group`

The `<peer-group-name>` parameter specifies the name of the group and can be up to 80 characters long. The name can contain special characters and internal blanks. If you use internal blanks, you must use quotation marks around the name. For example, the command **neighbor “My Three Peers” peer-group** is valid, but the command **neighbor My Three Peers peer-group** is not valid.

Syntax: `[no] neighbor <ip-addr> | <peer-group-name>`
`[advertisement-interval <num>]`
`[default-originate [route-map <map-name>]]`
`[description <string>]`
`[distribute-list in | out <num,num,...> | <ACL-num> in | out]`
`[ebgp-multihop [<num>]]`
`[filter-list in | out <num,num,...> | <ACL-num> in | out | weight]`
`[maximum-prefix <num> [<threshold>] [teardown]]`
`[next-hop-self]`
`[password [0 | 1] <string>]`
`[prefix-list <string> in | out]`
`[remote-as <as-number>]`
`[remove-private-as]`
`[route-map in | out <map-name>]`
`[route-reflector-client]`
`[send-community]`
`[soft-reconfiguration inbound]`
`[shutdown]`
`[timers keep-alive <num> hold-time <num>]`
`[update-source loopback <num>]`
`[weight <num>]`

The `<ip-addr> | <peer-group-name>` parameter indicates whether you are configuring a peer group or an individual neighbor. You can specify a peer group name or IP address with the **neighbor** command. If you specify a peer group name, you are configuring a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. Use the `<ip-addr>` parameter if you are configuring an individual neighbor instead of a peer group. Refer to [“Adding BGP4 neighbors”](#) on page 1354.

The remaining parameters are the same ones supported for individual neighbors. Refer to [“Adding BGP4 neighbors”](#) on page 1354.

Applying a peer group to a neighbor

After you configure a peer group, you can add neighbors to the group. When you add a neighbor to a peer group, you are applying all the neighbor attributes specified in the peer group to the neighbor.

To add neighbors to a peer group, enter commands such as the following.

```
Brocade(config-bgp-router)#neighbor 192.168.1.12 peer-group PeerGroup1
Brocade(config-bgp-router)#neighbor 192.168.2.45 peer-group PeerGroup1
Brocade(config-bgp-router)#neighbor 192.168.3.69 peer-group PeerGroup1
```

The commands in this example add three neighbors to the peer group “PeerGroup1”. As members of the peer group, the neighbors automatically receive the neighbor parameter values configured for the peer group. You also can override the parameters (except parameters that govern outbound traffic) on an individual neighbor basis. For neighbor parameters not specified for the peer group, the neighbors use the default values.

Syntax: `neighbor <ip-addr> peer-group <peer-group-name>`

The `<ip-addr>` parameter specifies the IP address of the neighbor.

The `<peer-group-name>` parameter specifies the peer group name.

NOTE

You must add the peer group before you can add neighbors to it.

Administratively shutting down a session with a BGP4 neighbor

You can prevent the Layer 3 switch from starting a BGP4 session with a neighbor by administratively shutting down the neighbor. This option is very useful for situations in which you want to configure parameters for a neighbor but are not ready to use the neighbor. You can shut the neighbor down as soon as you have added it the Layer 3 switch, configure the neighbor parameters, then allow the Layer 3 switch to re-establish a session with the neighbor by removing the shutdown option from the neighbor.

When you apply the new option to shut down a neighbor, the option takes place immediately and remains in effect until you remove the option. If you save the configuration to the startup-config file, the shutdown option remains in effect even after a software reload.

NOTE

The software also contains an option to end the session with a BGP4 neighbor and thus clear the routes learned from the neighbor. Unlike this clear option, the option for shutting down the neighbor can be saved in the startup-config file and thus can prevent the Layer 3 switch from establishing a BGP4 session with the neighbor even after reloading the software.

NOTE

If you notice that a particular BGP4 neighbor never establishes a session with the Brocade Layer 3 switch, check the Layer 3 switch running-config and startup-config files to see whether the configuration contains a command that is shutting down the neighbor. The neighbor may have been shut down previously by an administrator.

To shut down a BGP4 neighbor, enter commands such as the following.

```
Brocade(config)#router bgp
Brocade(config-bgp-router)#neighbor 209.157.22.26 shutdown
Brocade(config-bgp-router)#write memory
```

Syntax: [no] neighbor <ip-addr> shutdown

The <ip-addr> parameter specifies the IP address of the neighbor.

Optional BGP4 configuration tasks

The following sections describe how to perform optional BGP4 configuration tasks.

Changing the Keep Alive Time and Hold Time

The Keep Alive Time specifies how frequently the router will send KEEPALIVE messages to its BGP4 neighbors. The Hold Time specifies how long the router will wait for a KEEPALIVE or UPDATE message from a neighbor before concluding that the neighbor is dead. When the router concludes that a BGP4 neighbor is dead, the router ends the BGP4 session and closes the TCP connection to the neighbor.

The default Keep Alive time is 60 seconds. The default Hold Time is 180 seconds. To change the timers, use either of the following methods.

NOTE

Generally, you should set the Hold Time to three times the value of the Keep Alive Time.

NOTE

You can override the global Keep Alive Time and Hold Time on individual neighbors. Refer to [“Adding BGP4 neighbors”](#) on page 1354.

To change the Keep Alive Time to 30 and Hold Time to 90, enter the following command.

```
Brocade(config-bgp-router)#timers keep-alive 30 hold-time 90
```

Syntax: timers keep-alive <num> hold-time <num>

For each keyword, <num> indicates the number of seconds. The Keep Alive Time can be 0 through 65535. The Hold Time can be 0 or 3 through 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the router waits indefinitely for messages from a neighbor without concluding that the neighbor is dead.

Changing the BGP4 next-hop update timer

By default, the Layer 3 switch updates its BGP4 next-hop tables and affected BGP4 routes five seconds after IGP route changes. You can change the update timer to a value from 1 through 30 seconds.

To change the BGP4 update timer value, enter the **update-time** command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#update-time 15
```

This command changes the update timer to 15 seconds.

Syntax: `[no] update-time <secs>`

The `<secs>` parameter specifies the number of seconds and can be from 1 through 30. The default is 5.

Enabling fast external fallover

BGP4 routers rely on KEEPALIVE and UPDATE messages from neighbors to signify that the neighbors are alive. For BGP4 neighbors that are two or more hops away, such messages are the only indication that the BGP4 protocol has concerning the alive state of the neighbors. As a result, if a neighbor dies, the router will wait until the Hold Time expires before concluding that the neighbor is dead and closing its BGP4 session and TCP connection with the neighbor.

The router waits for the Hold Time to expire before ending the connection to a directly-attached BGP4 neighbor that dies.

For directly attached neighbors, the router immediately senses loss of a connection to the neighbor from a change of state of the port or interface that connects the router to its neighbor. For directly attached EBGP neighbors, the router can use this information to immediately close the BGP4 session and TCP connection to locally attached neighbors that die.

NOTE

The fast external fallover feature applies only to directly attached EBGP neighbors. The feature does not apply to IBGP neighbors.

If you want to enable the router to immediately close the BGP4 session and TCP connection to locally attached neighbors that die, use either of the following methods.

To enable fast external fallover, enter the following command.

```
Brocade(config-bgp-router)#fast-external-fallover
```

To disable fast external fallover again, enter the following command.

```
Brocade(config-bgp-router)#no fast-external-fallover
```

Syntax: `[no] fast-external-fallover`

Changing the maximum number of paths for BGP4 load sharing

Load sharing enables the Layer 3 switch to balance traffic to a route across multiple equal-cost paths of the same type (EBGP or IBGP) for the route.

To configure the Layer 3 switch to perform BGP4 load sharing:

- Enable IP load sharing if it is disabled.
- Set the maximum number of paths. The default maximum number of BGP4 load sharing paths is 1, which means no BGP4 load sharing takes place by default.

NOTE

The maximum number of BGP4 load sharing paths cannot be greater than the maximum number of IP load sharing paths.

How load sharing affects route selection

During evaluation of multiple paths to select the best path to a given destination for installment in the IP route table, the last comparison the Layer 3 switch performs is a comparison of the internal paths:

- When IP load sharing is disabled, the Layer 3 switch prefers the path to the router with the lower router ID.
- When IP load sharing and BGP4 load sharing are enabled, the Layer 3 switch balances the traffic across the multiple paths instead of choosing just one path based on router ID.

Refer to [“How BGP4 selects a path for a route”](#) on page 1344 for a description of the BGP4 algorithm.

When you enable IP load sharing, the Layer 3 switch can load balance BGP4 or OSPF routes across up to four equal paths by default. You can change the number of IP load sharing paths to a value from 2 through 6.

How load sharing works

Load sharing is performed in round-robin fashion and is based on the destination IP address only. The first time the router receives a packet destined for a specific IP address, the router uses a round-robin algorithm to select the path that was not used for the last newly learned destination IP address. Once the router associates a path with a particular destination IP address, the router will always use that path as long as the router contains the destination IP address in its cache.

NOTE

The Layer 3 switch does not perform source routing. The router is concerned only with the paths to the next-hop routers, not the entire paths to the destination hosts.

A BGP4 destination can be learned from multiple BGP4 neighbors, leading to multiple BGP4 paths to reach the same destination. Each of the paths may be reachable through multiple IGP paths (multiple OSPF or RIP paths). In this case, the software installs all the multiple equal-cost paths in the BGP4 route table, up to the maximum number of BGP4 equal-cost paths allowed. The IP load sharing feature then distributes traffic across the equal-cost paths to the destination.

If an IGP path used by a BGP4 next-hop route path installed in the IP route table changes, then the BGP4 paths and IP paths are adjusted accordingly. For example, if one of the OSPF paths to reach the BGP4 next hop goes down, the software removes this path from the BGP4 route table and the IP route table. Similarly, if an additional OSPF path becomes available to reach the BGP4 next-hop router for a particular destination, the software adds the additional path to the BGP4 route table and the IP route table.

Changing the maximum number of shared BGP4 paths

When IP load sharing is enabled, BGP4 can balance traffic to a specific destination across up to four equal paths. You can set the maximum number of paths to a value from 1 through 4. The default is 1.

NOTE

The maximum number of BGP4 load sharing paths cannot be greater than the maximum number of IP load sharing paths. To increase the maximum number of IP load sharing paths, use the **ip load sharing <num>** command at the global CONFIG level of the CLI.

To change the maximum number of shared paths, enter commands such as the following.

```
Brocade(config)#router bgp
Brocade(config-bgp-router)#maximum-paths 4
Brocade(config-bgp-router)#write memory
```

Syntax: [no] maximum-paths <num>

The <num> parameter specifies the maximum number of paths across which the Layer 3 switch can balance traffic to a given BGP4 destination. You can change the maximum number of paths to a value from 2 through 4. The default is 1.

Customizing BGP4 load sharing

By default, when BGP4 load sharing is enabled, both IBGP and EBGP paths are eligible for load sharing, while paths from different neighboring autonomous systems are not eligible. You can change load sharing to apply only to IBGP or EBGP paths, or to support load sharing among paths from different neighboring autonomous systems.

To enable load sharing of IBGP paths only, enter the following command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#multipath ibgp
```

To enable load sharing of EBGP paths only, enter the following command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#multipath ebgp
```

To enable load sharing of paths from different neighboring autonomous systems, enter the following command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#multipath multi-as
```

Syntax: [no] multipath ebgp | ibgp | multi-as

The **ebgp** | **ibgp** | **multi-as** parameter specifies the change you are making to load sharing:

- **ebgp** – Load sharing applies only to EBGP paths. Load sharing is disabled for IBGP paths.
- **ibgp** – Load sharing applies only to IBGP paths. Load sharing is disabled for EBGP paths.
- **multi-as** – Load sharing is enabled for paths from different autonomous systems.

By default, load sharing applies to EBGP and IBGP paths, and does not apply to paths from different neighboring autonomous systems.

Specifying a list of networks to advertise

By default, the router sends BGP4 routes only for the networks you identify using the **network** command or that are redistributed into BGP4 from RIP or OSPF. You can specify up to 600 networks.

To specify a network to be advertised, use either of the following methods.

NOTE

The exact route must exist in the IP route table before the Layer 3 switch can create a local BGP route.

To configure the Layer 3 switch to advertise network 209.157.22.0/24, enter the following command.

```
Brocade(config-bgp-router)#network 209.157.22.0 255.255.255.0
```

Syntax: `network <ip-addr> <ip-mask> [nlri multicast | unicast | multicast unicast] [route-map <map-name>] | [weight <num>] | [backdoor]`

The `<ip-addr>` is the network number and the `<ip-mask>` specifies the network mask.

The `nlri multicast | unicast | multicast unicast` parameter specifies whether the neighbor is a multicast neighbor or a unicast neighbor. Optionally, you also can specify **unicast** if you want the Layer 3 switch to exchange unicast (BGP4) routes as well as multicast routes with the neighbor. The default is **unicast** only.

The `route-map <map-name>` parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured.

The `weight <num>` parameter specifies a weight to be added to routes to this network.

The `backdoor` parameter changes the administrative distance of the route to this network from the EBGp administrative distance (20 by default) to the Local BGP weight (200 by default), thus tagging the route as a backdoor route. Use this parameter when you want the router to prefer IGP routes such as RIP or OSPF routes over the EBGp route for the network.

Specifying a route map name when configuring BGP4 network information

You can specify a route map as one of the parameters when you configure a BGP4 network to be advertised. The Layer 3 switch can use the route map to set or change BGP4 attributes when creating a local BGP4 route.

To configure network information and use a route map to set or change BGP4 attributes, use the following CLI method.

NOTE

You must configure the route map before you can specify the route map name in a BGP4 network configuration.

To configure a route map, and use it to set or change route attributes for a network you define for BGP4 to advertise, enter commands such as the following.

```
Brocade(config)#route-map set_net permit 1
Brocade(config-routemap set_net)#set community no-export
Brocade(config-routemap set_net)#exit
Brocade(config)#router bgp
Brocade(config-bgp-router)#network 100.100.1.0/24 route-map set_net
```

The first two commands in this example create a route map named “set_net” that sets the community attribute for routes that use the route map to “NO_EXPORT”. The next two commands change the CLI to the BGP4 configuration level. The last command configures a network for advertising from BGP4, and associates the “set_net” route map with the network. When BGP4 originates the 100.100.1.0/24 network, BGP4 also sets the community attribute for the network to “NO_EXPORT”.

Syntax: `network <ip-addr> <ip-mask> [route-map <map-name>] | [weight <num>] | [backdoor]`

The **route-map** *<map-name>* parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured.

For information about the other parameters, refer to [“Defining route maps”](#) on page 1403.

Changing the default local preference

When the router uses the BGP4 algorithm to select a route to send to the IP route table, one of the parameters the algorithm uses is the local preference. Local preference is an attribute that indicates a degree of preference for a route relative to other routes. BGP4 neighbors can send the local preference value as an attribute of a route in an UPDATE message.

Local preference applies only to routes within the local AS. BGP4 routers can exchange local preference information with neighbors who also are in the local AS, but BGP4 routers do not exchange local preference information with neighbors in remote autonomous systems.

The default local preference is 100. For routes learned from EBGp neighbors, the default local preference is assigned to learned routes. For routes learned from IBGP neighbors, the local preference value is not changed for the route.

When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen.

NOTE

To set the local preference for individual routes, use route maps. Refer to [“Defining route maps”](#) on page 1403. Refer to [“How BGP4 selects a path for a route”](#) on page 1344 for information about the BGP4 algorithm.

To change the default local preference to 200, enter the following command.

```
Brocade(config-bgp-router)#default-local-preference 200
```

Syntax: **default-local-preference** *<num>*

The *<num>* parameter indicates the preference and can be a value from 0 through 4294967295.

Using the IP default route as a valid next hop for a BGP4 route

By default, the Layer 3 switch does not use a default route to resolve a BGP4 next-hop route. If the IP route lookup for the BGP4 next hop does not result in a valid IGP route (including static or direct routes), the BGP4 next hop is considered to be unreachable and the BGP4 route is not used.

In some cases, such as when the Layer 3 switch is acting as an edge router, you might want to allow the device to use the default route as a valid next hop. To do so, enter the following command at the BGP4 configuration level of the CLI.

```
Brocade(config-bgp-router)#next-hop-enable-default
```

Syntax: **[no] next-hop-enable-default**

Advertising the default route

By default, the Layer 3 switch does not originate and advertise a default route using BGP4. A BGP4 default route is the IP address 0.0.0.0 and the route prefix 0 or network mask 0.0.0.0. For example, 0.0.0.0/0 is a default route. You can enable the router to advertise a default BGP4 route using either of the following methods.

NOTE

The Brocade Layer 3 switch checks for the existence of an IGP route for 0.0.0.0/0 in the IP route table before creating a local BGP route for 0.0.0.0/0.

To enable the router to originate and advertise a default BGP4 route, enter the following command.

```
Brocade(config-bgp-router)#default-information-originate
```

Syntax: [no] default-information-originate

Changing the default MED (Metric) used for route redistribution

The Brocade Layer 3 switch can redistribute directly connected routes, static IP routes, RIP routes, and OSPF routes into BGP4. The MED (metric) is a global parameter that specifies the cost that will be applied to all routes by default when they are redistributed into BGP4. When routes are selected, lower metric values are preferred over higher metric values. The default BGP4 MED value is 0 and can be assigned a value from 0 through 4294967295.

NOTE

RIP and OSPF also have default metric parameters. The parameters are set independently for each protocol and have different ranges.

To change the default metric to 40, enter the following command.

```
Brocade(config-bgp-router)#default-metric 40
```

Syntax: default-metric <num>

The <num> indicates the metric and can be a value from 0 through 4294967295.

Enabling next-hop recursion

For each BGP4 route a Layer 3 switch learns, the Layer 3 switch performs a route lookup to obtain the IP address of the route next hop. A BGP4 route becomes eligible for installation into the IP route table only if the following conditions are true:

- The lookup succeeds in obtaining a valid next-hop IP address for the route.
- The path to the next-hop IP address is an Interior Gateway Protocol (IGP) path or a static route path.

By default, the software performs only one lookup for a BGP route next-hop IP address. If the next-hop lookup does not result in a valid next-hop IP address or the path to the next-hop IP address is a BGP path, the software considers the BGP route destination to be unreachable. The route is not eligible to be installed in the IP route table.

It is possible for the BGP route table to contain a route whose next-hop IP address is not reachable through an IGP route, even though a hop farther away can be reached by the Layer 3 switch through an IGP route. This can occur when the IGP does not learn a complete set of IGP routes, resulting in the Layer 3 switch learning about an internal route through IBGP instead of through an IGP. In this case, the IP route table does not contain a route that can be used to reach the BGP route destination.

To enable the Layer 3 switch to find the IGP route to a BGP route next-hop gateway, enable recursive next-hop lookups. When you enable recursive next-hop lookup, if the first lookup for a BGP route results in an IBGP path originated within the same Autonomous System (AS), rather than an IGP path or static route path, the Layer 3 switch performs a lookup on the next-hop gateway next-hop IP address. If this second lookup results in an IGP path, the software considers the BGP route to be valid and thus eligible for installation in the IP route table. Otherwise, the Layer 3 switch performs a lookup on the next-hop IP address of the next-hop gateway next hop, and so on, until one of the lookups results in an IGP route.

NOTE

The software does not support using the default route to resolve a BGP4 route's next hop. Instead, you must configure a static route or use an IGP to learn the route to the EBGP multihop peer.

Previous software releases support use of the default route to resolve routes learned from EBGP multihop neighbors. However, even in this case Brocade recommends that you use a static route for the EBGP multihop neighbor instead. In general, we recommend that you do not use the default route as the next hop for BGP4 routes, especially when there are two or more BGP4 neighbors. Using the default route can cause loops.

Example when recursive route lookups are disabled

Here is an example of the results of an unsuccessful next-hop lookup for a BGP route. In this case, next-hop recursive lookups are disabled. The example is for the BGP route to network 240.0.0.0/24.

```
Brocade#show ip bgp route
Total number of BGP Routes: 5
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix      Next Hop      Metric      LocPrf      Weight      Status
1       0.0.0.0/0    10.1.0.2      0           100         0          BI
       AS_PATH: 65001 4355 701 80
2       102.0.0.0/24 10.0.0.1      1           100         0          BI
       AS_PATH: 65001 4355 1
3       104.0.0.0/24 10.1.0.2      0           100         0          BI
       AS_PATH: 65001 4355 701 1 189
4       240.0.0.0/24 102.0.0.1    1          100        0          I
       AS_PATH: 65001 4355 3356 7170 1455
5       250.0.0.0/24 209.157.24.1  1           100         0          I
       AS_PATH: 65001 4355 701
```

In this example, the Layer 3 switch cannot reach 240.0.0.0/24, because the next-hop IP address for the route is an IBGP route instead of an IGP route, and thus is considered unreachable by the Layer 3 switch. Here is the IP route table entry for the BGP route next-hop gateway (102.0.0.1/24).

```

Brocade#show ip route 102.0.0.1
Total number of IP routes: 37

```

Network Address	NetMask	Gateway	Port	Cost	Type
102.0.0.0	255.255.255.0	10.0.0.1	1/1	1	B

The route to the next-hop gateway is a BGP route, not an IGP route, and thus cannot be used to reach 240.0.0.0/24. In this case, the Layer 3 switch tries to use the default route, if present, to reach the subnet that contains the BGP route next-hop gateway.

```

Brocade#show ip route 240.0.0.0/24
Total number of IP routes: 37

```

Network Address	NetMask	Gateway	Port	Cost	Type
0.0.0.0	0.0.0.0	10.0.0.202	1/1	1	S

Example when recursive route lookups are enabled

When recursive next-hop lookups are enabled, the Layer 3 switch recursively looks up the next-hop gateways along the route until the Layer 3 switch finds an IGP route to the BGP route destination. Here is an example.

```

Brocade#show ip bgp route
Total number of BGP Routes: 5
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED

```

	Prefix	Next Hop	Metric	LocPrf	Weight	Status
1	0.0.0.0/0	10.1.0.2	0	100	0	BI
	AS_PATH: 65001 4355 701 80					
2	102.0.0.0/24	10.0.0.1	1	100	0	BI
	AS_PATH: 65001 4355 1					
3	104.0.0.0/24	10.1.0.2	0	100	0	BI
	AS_PATH: 65001 4355 701 1 189					
4	240.0.0.0/24	102.0.0.1	1	100	0	BI
	AS_PATH: 65001 4355 3356 7170 1455					
5	250.0.0.0/24	209.157.24.1	1	100	0	I
	AS_PATH: 65001 4355 701					

The first lookup results in an IBGP route, to network 102.0.0.0/24.

```

Brocade#show ip route 102.0.0.1
Total number of IP routes: 38

```

Network Address	NetMask	Gateway	Port	Cost	Type
102.0.0.0	255.255.255.0	10.0.0.1	1/1	1	B
AS_PATH: 65001 4355 1					

Since the route to 102.0.0.1/24 is not an IGP route, the Layer 3 switch cannot reach the next hop through IP, and thus cannot use the BGP route. In this case, since recursive next-hop lookups are enabled, the Layer 3 switch next performs a lookup for 102.0.0.1 next-hop gateway, 10.0.0.1.

Optional BGP4 configuration tasks

```
Brocade#show ip bgp route 102.0.0.0
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
Prefix          Next Hop          Metric      LocPrf      Weight Status
1      102.0.0.0/24    10.0.0.1      1           100         0      BI
      AS_PATH: 65001 4355 1
```

The next-hop IP address for 102.0.0.1 is not an IGP route, which means the BGP route destination still cannot be reached through IP. The recursive next-hop lookup feature performs a lookup on 10.0.0.1 next-hop gateway.

```
Brocade#show ip route 10.0.0.1
Total number of IP routes: 38
Network Address    NetMask          Gateway          Port      Cost      Type
10.0.0.0           255.255.255.0    0.0.0.0          1/1       1         D
      AS_PATH: 65001 4355 1
```

This lookup results in an IGP route. In fact, this route is a directly-connected route. As a result, the BGP route destination is now reachable through IGP, which means the BGP route is eligible for installation in the IP route table. Here is the BGP route in the IP route table.

```
Brocade#show ip route 240.0.0.0/24
Total number of IP routes: 38
Network Address    NetMask          Gateway          Port      Cost      Type
240.0.0.0          255.255.255.0    10.0.0.1         1/1       1         B
      AS_PATH: 65001 4355 1
```

This Layer 3 switch can use this route because the Layer 3 switch has an IP route to the next-hop gateway. Without recursive next-hop lookups, this route would not be in the IP route table.

Enabling recursive next-hop lookups

The recursive next-hop lookups feature is disabled by default. To enable recursive next-hop lookups, enter the **next-hop-recursion** command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#next-hop-recursion
```

Syntax: [no] next-hop-recursion

Changing administrative distances

BGP4 routers can learn about networks from various protocols, including the EBGP portion of BGP4 and IGP's such as OSPF and RIP. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned.

To select one route over another based on the source of the route information, the Layer 3 switch can use the administrative distances assigned to the sources. The administrative distance is a protocol-independent metric that IP routers use to compare routes from different sources.

The Layer 3 switch re-advertises a learned best BGP4 route to the Layer 3 switch neighbors even when the software does not also select that route for installation in the IP route table. The best BGP4 routes is the BGP4 path that the software selects based on comparison of the paths' BGP4 route parameters. Refer to [“How BGP4 selects a path for a route”](#) on page 1344.

When selecting a route from among different sources (BGP4, OSPF, RIP, static routes, and so on), the software compares the routes on the basis of each route administrative distance. If the administrative distance of the paths is lower than the administrative distance of paths from other sources (such as static IP routes, RIP, or OSPF), the BGP4 paths are installed in the IP route table.

NOTE

The software will replace a statically configured default route with a learned default route if the learned route administrative distance is lower than the statically configured default route distance. However, the default administrative distance for static routes is changed to 1, so only directly-connected routes are preferred over static routes when the default administrative distances for the routes are used.

The following default administrative distances are found on the Brocade Layer 3 switch:

- Directly connected – 0 (this value is not configurable)
- Static – 1 (applies to all static routes, including default routes)
- EBGp – 20
- OSPF – 110
- RIP – 120
- IBGP – 200
- Local BGP – 200
- Unknown – 255 (the router will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the router receives routes for the same network from OSPF and from RIP, the router will prefer the OSPF route by default. The administrative distances are configured in different places in the software. The Layer 3 switch re-advertises a learned best BGP4 route to neighbors by default, regardless of whether the route administrative distance is lower than other routes from different route sources to the same destination.

- To change the EBGp, IBGP, and Local BGP default administrative distances, see the instructions in this section.
- To change the default administrative distance for OSPF, refer to [“Administrative distance”](#) on page 1265.
- To change the default administrative distance for RIP, refer to [“Changing the administrative distance”](#) on page 1204.
- To change the default administrative distance for static routes, refer to [“Static routes configuration”](#) on page 991.

You can change the default EBGp, IBGP, and Local BGP administrative distances using either of the following methods.

To change the default administrative distances for EBGp, IBGP, and Local BGP, enter a command such as the following.

```
Brocade(config-bgp-router)#distance 180 160 40
```

Syntax: **distance** <external-distance> <internal-distance> <local-distance>

The <external-distance> sets the EBGp distance and can be a value from 1 through 255.

The <internal-distance> sets the IBGP distance and can be a value from 1 through 255.

The <local-distance> sets the Local BGP distance and can be a value from 1 through 255.

Requiring the first AS to be the neighbor AS

By default, the Brocade device does not require the first AS listed in the AS_SEQUENCE field of an AS path Update from an EBGP neighbor to be the AS that the neighbor who sent the Update is in. You can enable the Brocade device for this requirement.

When you enable the Brocade device to require the AS that an EBGP neighbor is in to be the same as the first AS in the AS_SEQUENCE field of an Update from the neighbor, the Brocade device accepts the Update only if the autonomous systems match. If the autonomous systems do not match, the Brocade device sends a Notification message to the neighbor and closes the session. The requirement applies to all Updates received from EBGP neighbors.

To enable this feature, enter the following command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#enforce-first-as
```

Syntax: [no] enforce-first-as

Disabling or re-enabling comparison of the AS-Path length

AS-Path comparison is Step 5 in the algorithm BGP4 uses to select the next path for a route. Comparison of the AS-Path length is enabled by default. To disable it, enter the following command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#as-path-ignore
```

This command disables comparison of the AS-Path lengths of otherwise equal paths. When you disable AS-Path length comparison, the BGP4 algorithm shown in [“How BGP4 selects a path for a route”](#) on page 1344 skips from Step 4 to Step 6.

Syntax: [no] as-path-ignore

Enabling or disabling comparison of the router IDs

Router ID comparison is Step 10 in the algorithm BGP4 uses to select the next path for a route.

NOTE

Comparison of router IDs is applicable only when BGP4 load sharing is disabled.

When router ID comparison is enabled, the path comparison algorithm compares the router IDs of the neighbors that sent the otherwise equal paths:

- If BGP4 load sharing is disabled (maximum-paths 1), the Layer 3 switch selects the path that came from the neighbor with the lower router ID.
- If BGP4 load sharing is enabled, the Layer 3 switch load shares among the remaining paths. In this case, the router ID is not used to select a path.

NOTE

Router ID comparison is disabled by default. In previous releases, router ID comparison is enabled by default and cannot be disabled.

To enable router ID comparison, enter the **compare-routerid** command at the BGP configuration level of the CLI.

```
Brocade(config-bgp-router)#compare-routerid
```

Syntax: [no] compare-routerid

For more information, refer to [“How BGP4 selects a path for a route”](#) on page 1344.

Configuring the Layer 3 switch to always compare Multi-Exit Discriminators (MEDs)

A Multi-Exit Discriminator (MED) is a value that the BGP4 algorithm uses when comparing multiple paths received from different BGP4 neighbors in the same AS for the same route. In BGP4, a route MED is equivalent to its “metric”:

- BGP4 compares the MEDs of two otherwise equivalent paths **if and only if** the routes were learned from the same neighboring AS. This behavior is called **deterministic MED**. Deterministic MED is always enabled and cannot be disabled.

In addition, you can enable the Layer 3 switch to always compare the MEDs, regardless of the AS information in the paths. To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.

- The Layer 3 switch compares the MEDs based on one or more of the following conditions. By default, the Layer 3 switch compares the MEDs of paths **only if** the first AS in the paths is the same. (The Layer 3 switch skips over the AS-CONFED-SEQUENCE if present.)

You can enable the Layer 3 switch to always compare the MEDs, regardless of the AS information in the paths. For example, if the router receives UPDATES for the same route from neighbors in three autonomous systems, the router would compare the MEDs of all the paths together, rather than comparing the MEDs for the paths in each AS individually.

NOTE

By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the Layer 3 switch favoring the route paths that are missing their MEDs. You can use the **med-missing-as-worst** command to make the Layer 3 switch regard a BGP route with a missing MED attribute as the least favorable route, when comparing the MEDs of the routes.

NOTE

MED comparison is not performed for internal routes originated within the local AS or confederation.

To configure the router to always compare MEDs, enter the following command.

```
Brocade(config-bgp-router)#always-compare-med
```

Syntax: [no] always-compare-med

Treating missing MEDs as the worst MEDs

By default, the Layer 3 switch favors a lower MED over a higher MED during MED comparison. Since the Layer 3 switch assigns the value 0 to a route path MED if the MED value is missing, the default MED comparison results in the Layer 3 switch favoring the route paths that are missing their MEDs.

To change this behavior so that the Layer 3 switch favors a route that has a MED over a route that is missing its MED, enter the following command at the BGP4 configuration level of the CLI.

```
Brocade(config-bgp-router)#med-missing-as-worst
```

Syntax: [no] med-missing-as-worst

NOTE

This command affects route selection only when route paths are selected based on MED comparison. It is still possible for a route path that is missing its MED to be selected based on other criteria. For example, a route path with no MED can be selected if its weight is larger than the weights of the other route paths.

Route reflection parameter configuration

Normally, all the BGP routers within an AS are fully meshed. Each of the routers has an IBGP session with each of the other BGP routers in the AS. Each IBGP router thus has a route for each of its IBGP neighbors. For large autonomous systems containing many IBGP routers, the IBGP route information in each of the fully-meshed IBGP routers can introduce too much administrative overhead.

To avoid this problem, you can hierarchically organize your IGP routers into clusters:

- A **cluster** is a group of IGP routers organized into route reflectors and route reflector clients. You configure the cluster by assigning a cluster ID on the route reflector and identifying the IGP neighbors that are members of that cluster. All the configuration for route reflection takes place on the route reflectors. The clients are unaware that they are members of a route reflection cluster. All members of the cluster must be in the same AS. The cluster ID can be any number from 0 through 4294967295. The default is the router ID, expressed as a 32-bit number.

NOTE

If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

- A **route reflector** is an IGP router configured to send BGP route information to all the clients (other BGP4 routers) within the cluster. Route reflection is enabled on all Brocade BGP4 routers by default but does not take effect unless you add route reflector clients to the router.
- A **route reflector client** is an IGP router identified as a member of a cluster. You identify a router as a route reflector client on the router that is the route reflector, not on the client. The client itself requires no additional configuration. In fact, the client does not know that it is a route reflector client. The client just knows that it receives updates from its neighbors and does not know whether one or more of those neighbors are route reflectors.

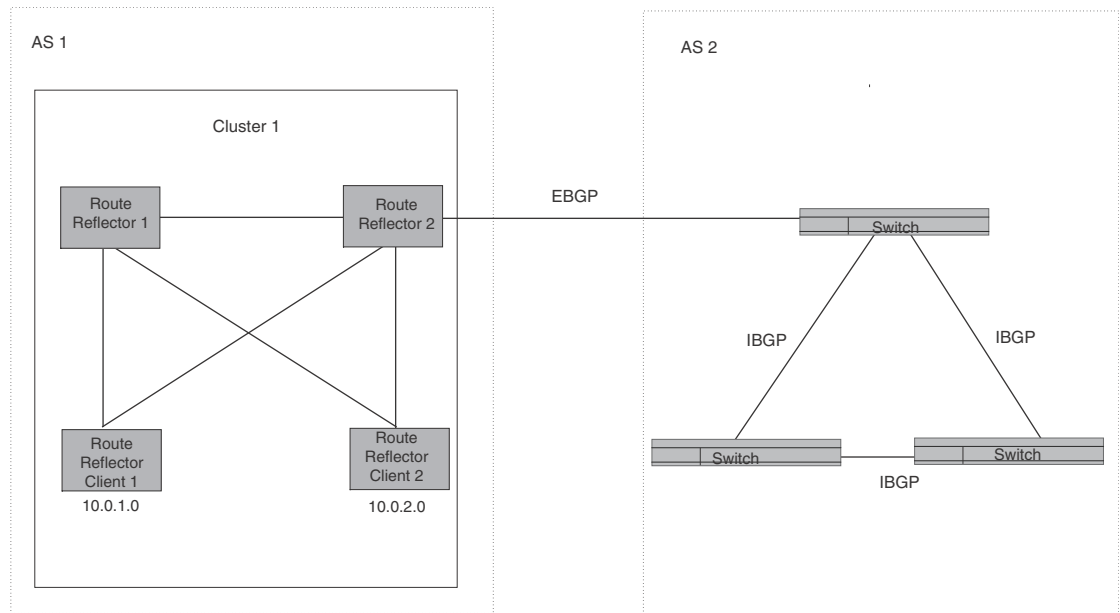
NOTE

Route reflection applies only among IBGP routers within the same AS. You cannot configure a cluster that spans multiple autonomous systems.

Figure 167 shows an example of a route reflector configuration. In this example, two Layer 3 switches are configured as route reflectors for the same cluster. The route reflectors provide redundancy in case one of the reflectors becomes unavailable. Without redundancy, if a route reflector becomes unavailable, its clients are cut off from BGP4 updates.

AS1 contains a cluster with two route reflectors and two clients. The route reflectors are fully meshed with other BGP4 routers, but the clients are not fully meshed. They rely on the route reflectors to propagate BGP4 route updates.

FIGURE 167 Example of a route reflector configuration



Route reflection based on RFC 2796

Route reflection on Brocade devices is based on RFC 2796. This updated RFC helps eliminate routing loops that are possible in some implementations of the older specification, RFC 1966.

NOTE

The configuration procedure for route reflection is the same regardless of whether your software release is using RFC 1966 or RFC 2796. However, the operation of the feature is different as explained below.

RFC 2796 provides more details than RFC 1966 regarding the use of the route reflection attributes, `ORIGINATOR_ID` and `CLUSTER_LIST`, to help prevent loops:

- **ORIGINATOR_ID** – Specifies the router ID of the BGP4 switch that originated the route. The route reflector inserts this attribute when reflecting a route to an IBGP neighbor. If a BGP4 switch receives an advertisement that contains its own router ID as the `ORIGINATOR_ID`, the switch discards the advertisement and does not forward it.
- **CLUSTER_LIST** – A list of the route reflection clusters through which the advertisement has passed. A cluster contains a route reflector and its clients. When a route reflector reflects a route, the route reflector adds its cluster ID to the front of the `CLUSTER_LIST`. If a route reflector receives a route that has its own cluster ID, the switch discards the advertisement and does not forward it.

The Brocade device handles the attributes as follows:

- The Layer 3 switch adds the attributes only if it is a route reflector, and only when advertising IBGP route information to other IBGP neighbors. The attributes are not used when communicating with EBGP neighbors.
- A Layer 3 switch configured as a route reflector sets the ORIGINATOR_ID attribute to the router ID of the router that originated the route. Moreover, the route reflector sets the attribute only if this is the first time the route is being reflected (sent by a route reflector). In previous software releases, the route reflector set the attribute to the router ID of the route reflector itself. When a Layer 3 switch receives a route that already has the ORIGINATOR_ID attribute set, the Layer 3 switch does not change the value of the attribute.
- If a Layer 3 switch receives a route whose ORIGINATOR_ID attribute has the value of the Layer 3 switch own router ID, the Layer 3 switch discards the route and does not advertise it. By discarding the route, the Layer 3 switch prevents a routing loop. The Layer 3 switch did not discard the route in previous software releases.
- The first time a route is reflected by a Layer 3 switch configured as a route reflector, the route reflector adds the CLUSTER_LIST attribute to the route. Other route reflectors who receive the route from an IBGP neighbor add their cluster IDs to the front of the route CLUSTER_LIST. If the route reflector does not have a cluster ID configured, the Layer 3 switch adds its router ID to the front of the CLUSTER_LIST.
- If a Layer 3 switch configured as a route reflector receives a route whose CLUSTER_LIST contains the route reflector own cluster ID, the route reflector discards the route and does not forward it.

Configuration procedures for BGP4 route reflector

To configure a Brocade Layer 3 switch to be a BGP4 route reflector, use either of the following methods.

NOTE

All configuration for route reflection takes place on the route reflectors, not on the clients.

Enter the following commands to configure a Brocade Layer 3 switch as route reflector 1 in [Figure 167](#) on page 1379. To configure route reflector 2, enter the same commands on the Layer 3 switch that will be route reflector 2. The clients require no configuration for route reflection.

```
Brocade(config-bgp-router)#cluster-id 1
Brocade(config-bgp-router)#neighbor 10.0.1.0 route-reflector-client
Brocade(config-bgp-router)#neighbor 10.0.2.0 route-reflector-client
```

Syntax: [no] cluster-id <num> | <ip-addr>

The <num> | <ip-addr> parameter specifies the cluster ID and can be a number from 0 through 4294967295 or an IP address. The default is the router ID. You can configure one cluster ID on the router. All route-reflector clients for the router are members of the cluster.

NOTE

If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

To add an IBGP neighbor to the cluster, enter the following command.

Syntax: neighbor <ip-addr> route-reflector-client

For more information about the **neighbor** command, refer to [“Adding BGP4 neighbors”](#) on page 1354.

By default, the clients of a route reflector are not required to be fully meshed; the routes from a client are reflected to other clients. However, if the clients are fully meshed, route reflection is not required between clients.

If you need to disable route reflection between clients, enter the following command. When the feature is disabled, route reflection does not occur between clients but reflection does still occur between clients and non-clients.

```
Brocade(config-bgp-router)#no client-to-client-reflection
```

Enter the following command to re-enable the feature.

```
Brocade(config-bgp-router)#client-to-client-reflection
```

Syntax: [no] client-to-client-reflection

Configuration notes for BGP4 autonomous systems

A **confederation** is a BGP4 Autonomous System (AS) that has been subdivided into multiple, smaller autonomous systems. Subdividing an AS into smaller autonomous systems simplifies administration and reduces BGP-related traffic, thus reducing the complexity of the Interior Border Gateway Protocol (IBGP) mesh among the BGP routers in the AS.

The Brocade implementation of this feature is based on RFC 3065.

Normally, all BGP routers within an AS must be fully meshed, so that each BGP router has interfaces to all the other BGP routers within the AS. This is feasible in smaller autonomous systems but becomes unmanageable in autonomous systems containing many BGP routers.

When you configure BGP routers into a confederation, all the routers within a sub-AS (a subdivision of the AS) use IBGP and must be fully meshed. However, routers use EBGP to communicate between different sub-autonomous systems.

NOTE

Another method for reducing the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, configure a confederation. You can run a separate IGP within each sub-AS.

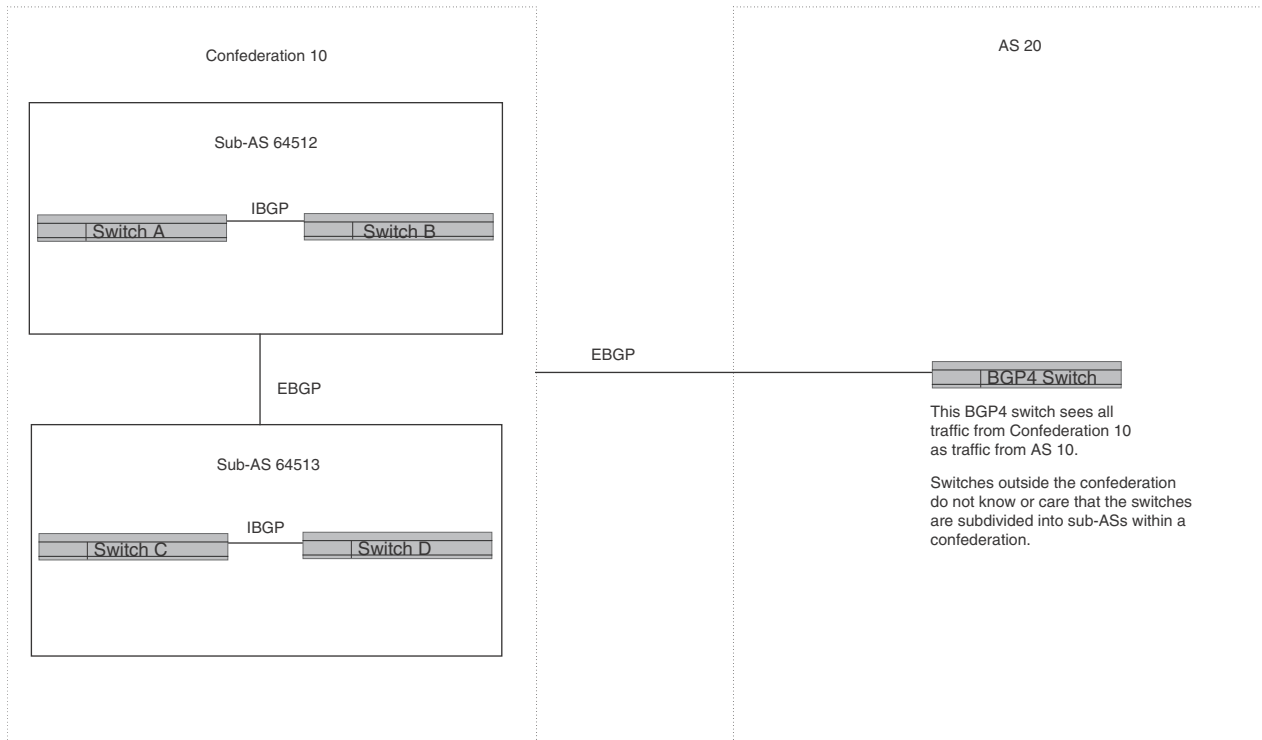
To configure a confederation, configure groups of BGP routers into sub-autonomous systems. A sub-AS is simply an AS. The term “sub-AS” distinguishes autonomous systems within a confederation from autonomous systems that are not in a confederation. For the viewpoint of remote autonomous systems, the confederation ID is the AS ID. Remote autonomous systems do not know that the AS represents multiple sub-autonomous systems with unique AS IDs.

NOTE

You can use any valid AS numbers for the sub-autonomous systems. If your AS is connected to the Internet, Brocade recommends that you use numbers from within the private AS range (64512 through 65535). These are private autonomous systems numbers and BGP4 routers do not propagate these AS numbers to the Internet.

Figure 168 shows an example of a BGP4 confederation.

FIGURE 168 Example of a BGP4 confederation



In this example, four switches are configured into two sub-autonomous systems, each containing two of the switches. The sub-autonomous systems are members of confederation 10. Switches within a sub-AS must be fully meshed and communicate using IBGP. In this example, Switches A and B use IBGP to communicate. Switches C and D also use IBGP. However, the sub-autonomous systems communicate with one another using EBGP. For example, Switch A communicates with Switch C using EBGP. The switches in the confederation communicate with other autonomous systems using EBGP.

Switches in other autonomous systems are unaware that Switches A through D are configured in a confederation. In fact, when switches in confederation 10 send traffic to switches in other autonomous systems, the confederation ID is the same as the AS number for the switches in the confederation. Thus, switches in other autonomous systems see traffic from AS 10 and are unaware that the switches in AS 10 are subdivided into sub-autonomous systems within a confederation.

Configuring a BGP confederation

Perform the following configuration tasks on each BGP router within the confederation:

- Configure the local AS number. The local AS number indicates membership in a sub-AS. All BGP switches with the same local AS number are members of the same sub-AS. BGP switches use the local AS number when communicating with other BGP switches within the confederation.

- Configure the confederation ID. The confederation ID is the AS number by which BGP switches outside the confederation know the confederation. Thus, a BGP switch outside the confederation is not aware and does not care that your BGP switches are in multiple sub-autonomous systems. BGP switches use the confederation ID when communicating with switches outside the confederation. The confederation ID must be different from the sub-AS numbers.
- Configure the list of the sub-AS numbers that are members of the confederation. All the switches within the same sub-AS use IBGP to exchange switch information. Switches in different sub-autonomous systems within the confederation use EBGP to exchange switch information.

To configure four Layer 3 switches to be a member of confederation 10 (as shown in [Figure 168](#)), consisting of two sub-autonomous systems (64512 and 64513), enter commands such as the following.

Commands for router A

```
BrocadeA(config)#router bgp
BrocadeA(config-bgp-router)#local-as 64512
BrocadeA(config-bgp-router)#confederation identifier 10
BrocadeA(config-bgp-router)#confederation peers 64512 64513
BrocadeA(config-bgp-router)#write memory
```

Syntax: local-as <num>

The <num> parameter with the **local-as** command indicates the AS number for the BGP switches within the sub-AS. You can specify a number from 1 through 65535. Brocade recommends that you use a number within the range of well-known private autonomous systems, 64512 through 65535.

Syntax: confederation identifier <num>

The <num> parameter with the **confederation identifier** command indicates the confederation number. The confederation ID is the AS number by which BGP switches outside the confederation know the confederation. Thus, a BGP switch outside the confederation is not aware and does not care that your BGP switches are in multiple sub-autonomous systems. BGP switches use the confederation ID when communicating with switches outside the confederation. The confederation ID must be different from the sub-AS numbers. You can specify a number from 1 through 65535.

Syntax: confederation peers <num> [<num> ...]

The <num> parameter with the **confederation peers** command indicates the sub-AS numbers for the sub-autonomous systems in the confederation. You must specify all the sub-autonomous systems contained in the confederation. All the switches within the same sub-AS use IBGP to exchange switch information. Switches in different sub-autonomous systems within the confederation use EBGP to exchange switch information. You can specify a number from 1 through 65535.

Commands for router B

```
BrocadeB(config)#router bgp
BrocadeB(config-bgp-router)#local-as 64512
BrocadeB(config-bgp-router)#confederation identifier 10
BrocadeB(config-bgp-router)#confederation peers 64512 64513
BrocadeB(config-bgp-router)#write memory
```

Commands for router C

```
BrocadeC(config)#router bgp
BrocadeC(config-bgp-router)#local-as 64513
BrocadeC(config-bgp-router)#confederation identifier 10
BrocadeC(config-bgp-router)#confederation peers 64512 64513
BrocadeC(config-bgp-router)#write memory
```

Commands for router D

```
BrocadeD(config)#router bgp
BrocadeD(config-bgp-router)#local-as 64513
BrocadeD(config-bgp-router)#confederation identifier 10
BrocadeD(config-bgp-router)#confederation peers 64512 64513
BrocadeD(config-bgp-router)#write memory
```

Aggregating routes advertised to BGP4 neighbors

By default, the Layer 3 switch advertises individual routes for all the networks. The aggregation feature allows you to configure the Layer 3 switch to aggregate routes in a range of networks into a single CIDR number. For example, without aggregation, the Layer 3 switch will individually advertise routes for networks 207.95.1.0, 207.95.2.0, and 207.95.3.0. You can configure the Layer 3 switch to instead send a single, aggregate route for the networks. The aggregate route would be advertised as 207.95.0.0.

NOTE

To summarize CIDR networks, you must use the aggregation feature. The auto summary feature does not summarize networks that use CIDR numbers instead of class A, B, or C numbers.

To aggregate routes for 209.157.22.0, 209.157.23.0, and 209.157.24.0, enter the following command.

```
Brocade(config-bgp-router)#aggregate-address 209.157.0.0 255.255.0.0
```

Syntax: `aggregate-address <ip-addr> <ip-mask> [as-set] [nlri multicast | unicast | multicast unicast] [summary-only] [suppress-map <map-name>] [advertise-map <map-name>] [attribute-map <map-name>]`

The `<ip-addr>` and `<ip-mask>` parameters specify the aggregate value for the networks. Specify 0 for the host portion and for the network portion that differs among the networks in the aggregate. For example, to aggregate 10.0.1.0, 10.0.2.0, and 10.0.3.0, enter the IP address 10.0.0.0 and the network mask 255.255.0.0.

The `as-set` parameter causes the router to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The `nlri multicast | unicast | multicast unicast` parameter specifies whether the neighbor is a multicast neighbor or a unicast neighbor. Optionally, you also can specify `unicast` if you want the Layer 3 switch to exchange unicast (BGP4) routes as well as multicast routes with the neighbor. The default is `unicast` only.

The `summary-only` parameter prevents the router from advertising more specific routes contained within the aggregate route.

The `suppress-map <map-name>` parameter prevents the more specific routes contained in the specified route map from being advertised.

The **advertise-map** <map-name> parameter configures the router to advertise the more specific routes in the specified route map.

The **attribute-map** <map-name> parameter configures the router to set attributes for the aggregate routes based on the specified route map.

NOTE

For the **suppress-map**, **advertise-map**, and **attribute-map** parameters, the route map must already be defined. Refer to [“Defining route maps”](#) on page 1403 for information on defining a route map.

Configuring BGP4 graceful restart

By default, BGP4 graceful restart is enabled for the global routing instance. This section describes how to disable and re-enable the BGP4 restart feature and change the default values for associated timers.

For information about displaying BGP4 graceful restart neighbor information, refer to [“Displaying BGP4 graceful restart neighbor information”](#) on page 1451.

Configuring BGP4 graceful restart

BGP4 graceful restart is enabled by default on a FastIron Layer 3 switch. To disable it, use the following commands:

```
Brocade(config)# router bgp
Brocade(config-bgp)# no graceful-restart
```

To re-enable BGP4 graceful restart after it has been disabled, enter the following commands.

```
Brocade(config)# router bgp
Brocade(config-bgp)# graceful-restart
```

Syntax: [no] graceful-restart

Configuring timers for BGP4 graceful restart (optional)

You can change the default values for the following timers:

- Restart timer
- Stale routes timer
- Purge timer

Configuring the restart timer for BGP4 graceful restart

Use the following command to specify the maximum amount of time a device will maintain routes from and forward traffic to a restarting device.

```
Brocade(config-bgp)# graceful-restart restart-timer 150
```

Syntax: [no] graceful-restart restart-timer <seconds>

The `<seconds>` variable is the maximum restart wait time advertised to neighbors. Possible values are from 1 through 3600 seconds. The default value is 120 seconds.

Configuring the BGP4 graceful restart stale routes timer

Use the following command to specify the maximum amount of time a helper device will wait for an end-of-RIB message from a peer before deleting routes from that peer.

```
Brocade(config-bgp)# graceful-restart stale-routes-time 120
```

Syntax: `[no] graceful-restart stale-routes-time <seconds>`

The `<seconds>` variable is the maximum time before a helper device cleans up stale routes. Possible values are from 1 through 3600 seconds. The default value is 360 seconds.

Configuring the BGP4 graceful restart purge timer

Use the following command to specify the maximum amount of time a device will maintain stale routes in its routing table before purging them.

```
Brocade(config-bgp)# graceful-restart purge-time 900
```

Syntax: `[no] graceful-restart purge-time <seconds>`

The `<seconds>` variable sets the maximum time before a restarting device cleans up stale routes. Possible values are from 1 through 3600 seconds. The default value is 600 seconds.

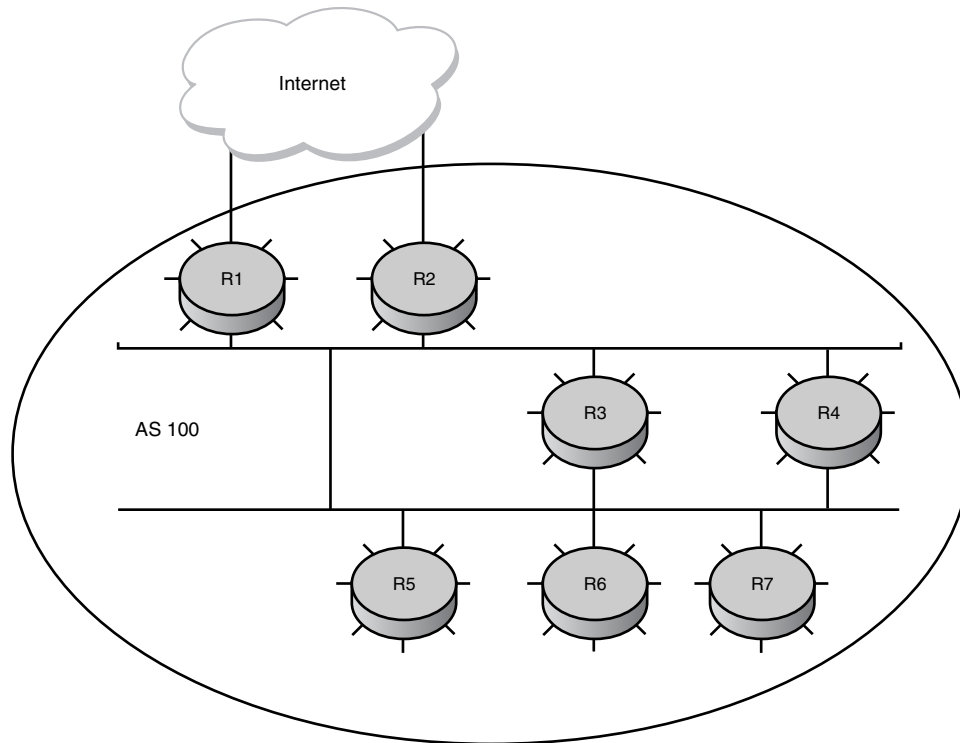
BGP null0 routing

The null0 routes were previously treated as invalid routes for BGP next hop resolution. BGP now uses the null0 route to resolve its next hop. Thus, null0 route in the routing table (for example, static route) is considered as a valid route by BGP. If the next hop for BGP resolves into a null0 route, the BGP route is also installed as a null0 route in the routing table.

The null0 routing feature allows network administrators to block certain network prefixes, by using null0 routes and route-maps. The combined use of null0 routes and route maps blocks traffic from a particular network prefix, telling a remote router to drop all traffic for this network prefix by redistributing a null0 route into BGP.

Figure 169 shows a topology for a null0 routing application example.

FIGURE 169 Example of a null0 routing application



The following steps configure a null0 routing application for stopping denial of service attacks from remote hosts on the internet.

Configuration steps for BGP null 0 routing

1. Select one switch, S6, to distribute null0 routes throughout the BGP network.
2. Configure a route-map to match a particular tag (50) and set the next-hop address to an unused network address (199.199.1.1).
3. Set the local-preference to a value higher than any possible internal or external local-preference (50).
4. Complete the route map by setting origin to IGP.
5. On S6, redistribute the static routes into BGP, using route-map <route-map-name> (redistribute static route-map block user).
6. On S1, the router facing the internet, configure a null0 route matching the next-hop address in the route-map (ip route 199.199.1.1/32 null0).
7. Repeat step 3 for all switches interfacing with the internet (edge corporate routers). In this case, S2 has the same null0 route as S1.
8. On S6, configure the network prefixes associated with the traffic you want to drop. The static route IP address references a destination address. You are required to point the static route to the egress port, for example, Ethernet 3/7, and specify the tag 50, matching the route-map configuration.

Configuration examples for BGP null 0 routing

S6

The following configuration defines specific prefixes to filter.

```
Brocade(config)#ip route 110.0.0.40/29 ethernet 3/7 tag 50
Brocade(config)#ip route 115.0.0.192/27 ethernet 3/7 tag 50
Brocade(config)#ip route 120.014.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP.

```
Brocade(config)#router bgp
Brocade(config-bgp-router)#local-as 100
Brocade(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
Brocade(config-bgp-router)#redistribute static route-map blockuser
Brocade(config-bgp-router)#exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred.

```
Brocade(config)#route-map blockuser permit 10
Brocade(config-routemap blockuser)#match tag 50
Brocade(config-routemap blockuser)#set ip next-hop 199.199.1.1
Brocade(config-routemap blockuser)#set local-preference 1000000
Brocade(config-routemap blockuser)#set origin igp
Brocade(config-routemap blockuser)#exit
```

S1

The following configuration defines the null0 route to the specific next hop address. The next hop address 199.199.1.1 points to the null0 route.

```
Brocade(config)#ip route 199.199.1.1/32 null0
Brocade(config)#router bgp
Brocade(config-bgp-router)#local-as 100
Brocade(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

S2

The following configuration defines a null0 route to the specific next hop address. The next hop address 199.199.1.1 points to the null0 route, which gets blocked.

```
Brocade(config)#ip route 199.199.1.1/32 null0
Brocade(config)#router bgp
Brocade(config-bgp-router)#local-as 100
Brocade(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
```

```

Brocade(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
Brocade (config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
Brocade(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100

```

Show commands for BGP null 0 routing

After configuring the null0 application, you can display the output.

S6

The following is the **show ip route static** output for S6.

```

Brocade#show ip route static
Type Codes - B:BGP D:Connected S:Static R:RIP O:OSPF; Cost - Dist/Metric
      Destination      Gateway      Port      Cost      Type
1      110.0.0.40/29      DIRECT      eth 3/7      1/1      S
2      115.0.0.192/27      DIRECT      eth 3/7      1/1      S
3      120.0.14.0/23      DIRECT      eth 3/7      1/1      S

```

S1 and S2

The following is the **show ip route static** output for S1 and S2.

```

Brocade#show ip route static
Type Codes - B:BGP D:Connected S:Static R:RIP O:OSPF; Cost - Dist/Metric
      Destination      Gateway      Port      Cost      Type
1      199.199.1.1/32      DIRECT      drop      1/1      S

```

S6

The following is the **show ip bgp route** output for S6

BGP null0 routing

```

Brocade#show ip bgp route
Total number of BGP Routes: 126
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED E:EBGP
H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED s:STALE

```

	Prefix	Next Hop	Metric	LocPrf	Weight	Status
1	30.0.1.0/24	40.0.1.3	0	100	0	BI
	AS_PATH:					
.
9	110.0.0.16/30	90.0.1.3		100	0	I
	AS_PATH: 85					
10	110.0.0.40/29	199.199.1.1/32	1	1000000	32768	BL
	AS_PATH:					
11	110.0.0.80/28	90.0.1.3		100	0	I
.
.
36	115.0.0.96/28	30.0.1.3		100	0	I
	AS_PATH: 50					
37	115.0.0.192/27	199.199.1.1/32	1	10000000	32768	BL
	AS_PATH:					
.
64	120.0.7.0/24	70.0.1.3		100	0	I
	AS_PATH: 10					
65	120.0.14.0/23	199.199.1.1/32	1	1000000	32768	BL
	AS_PATH: ..					

S1 and S2

The **show ip route** output for S1 and S2 shows "drop" under the Port column for the network prefixes you configured with null0 routing.

```

Brocade#show ip route
Total number of IP routes: 133

```

Type	Codes	B:BGP	D:Connected	S:Static	R:RIP	O:OSPF	Cost	Dist/Metric
	Destination	Gateway	Port	Cost	Type			
1	9.0.1.24/32	DIRECT	loopback 1	0/0	D			
2	30.0.1.0/24	DIRECT	eth 2/7	0/0	D			
3	40.0.1.0/24	DIRECT	eth 2/1	0/0	D			
.								
13	110.0.0.6/31	90.0.1.3	eth 2/2	20/1	B			
14	110.0.0.16/30	90.0.1.3	eth 2/2	20/1	B			
15	110.0.0.40/29	DIRECT	drop	200/0	B			
.			
42	115.0.0.192/27	DIRECT	drop	200/0	B			
43	115.0.1.128/26	30.0.1.3	eth 2/7	20/1	B			
.			
69	120.0.7.0/24	70.0.1.3	eth 2/10	20/1	B			
70	120.0.14.0/23	DIRECT	drop	200/0	B			
.			
.			
131	130.144.0.0/12	80.0.1.3	eth 3/4	20/1	B			
132	199.199.1.1/32	DIRECT	drop	1/1	S			

Modifying redistribution parameters

By default, the Layer 3 Switch does not redistribute route information between BGP4 and the IP IGPs (RIP and OSPF). You can configure the switch to redistribute OSPF routes, RIP routes, directly connected routes, or static routes into BGP4 by using the following methods.

To enable redistribution of all OSPF routes and directly attached routes into BGP4, enter the following commands.

```
Brocade(config)#router bgp
Brocade(config-bgp-router)#redistribute ospf
Brocade(config-bgp-router)#redistribute connected
Brocade(config-bgp-router)#write memory
```

Syntax: [no] **redistribute connected** | **ospf** | **rip** | **static**

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP.

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

NOTE

Entering **redistribute ospf** simply redistributes internal OSPF routes. If you want to redistribute external OSPF routes also, you must use the **redistribute ospf match external...** command. Refer to [“Redistributing OSPF external routes”](#) on page 1392.

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **static** parameter indicates that you are redistributing static routes into BGP.

Refer to the following sections for details on redistributing specific routes using the CLI:

- [“Redistributing connected routes”](#) on page 1391
- [“Redistributing RIP routes”](#) on page 1392
- [“Redistributing OSPF external routes”](#) on page 1392
- [“Redistributing static routes”](#) on page 1393

Redistributing connected routes

To configure BGP4 to redistribute directly connected routes, enter the following command.

```
Brocade(config-bgp-router)#redistribute connected
```

Syntax: **redistribute connected** [**metric** <num>] [**route-map** <map-name>]

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP4.

The **metric** <num> parameter changes the metric. You can specify a value from 0 through 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE

The route map you specify must already be configured on the switch. Refer to [“Defining route maps”](#) on page 1403 for information about defining route maps.

Redistributing RIP routes

To configure BGP4 to redistribute RIP routes and add a metric of 10 to the redistributed routes, enter the following command.

```
Brocade(config-bgp-router)#redistribute rip metric 10
```

Syntax: **redistribute rip** [**metric** <num>] [**route-map** <map-name>]

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **metric** <num> parameter changes the metric. Specify a value from 0 through 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE

The route map you specify must already be configured on the switch. Refer to [“Defining route maps”](#) on page 1403 for information about defining route maps.

Redistributing OSPF external routes

To configure the Layer 3 switch to redistribute OSPF external type 1 routes, enter the following command.

```
Brocade(config-bgp-router)#redistribute ospf match external1
```

Syntax: **redistribute ospf** [**match internal** | **external1** | **external2**] [**metric** <num>] [**route-map** <map-name>]

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

The **match internal** | **external1** | **external2** parameter applies only to OSPF. This parameter specifies the types of OSPF routes to be redistributed into BGP4. The default is internal.

NOTE

If you do not enter a value for the **match** parameter, (for example, you enter **redistribute ospf** only) then only internal OSPF routes will be redistributed.

The **metric** <num> parameter changes the metric. Specify a value from 0 through 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the OSPF route to the BGP4 route table.

NOTE

The route map you specify must already be configured on the switch. Refer to [“Defining route maps”](#) on page 1403 for information about defining route maps.

NOTE

If you use both the **redistribute ospf route-map** <map-name> command and the **redistribute ospf match internal** | **external1** | **external2** command, the software uses only the route map for filtering.

Redistributing static routes

To configure the Layer 3 switch to redistribute static routes, enter the following command.

```
Brocade(config-bgp-router)#redistribute static
```

Syntax: **redistribute static** [**metric** <num>] [**route-map** <map-name>]

The **static** parameter indicates that you are redistributing static routes into BGP4.

The **metric** <num> parameter changes the metric. Specify a value from 0 through 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the static route to the BGP4 route table.

NOTE

The route map you specify must already be configured on the switch. Refer to [“Defining route maps”](#) on page 1403 for information about defining route maps.

Disabling or re-enabling re-advertisement of all learned BGP4 routes to all BGP4 neighbors

By default, the Layer 3 switch re-advertises all learned best BGP4 routes to BGP4 neighbors, unless the routes are discarded or blocked by route maps or other filters.

If you want to prevent the Layer 3 switch from re-advertising a learned best BGP4 route unless that route also is installed in the IP route table, use the following CLI method.

To disable re-advertisement of BGP4 routes to BGP4 neighbors except for routes that the software also installs in the route table, enter the following command.

```
Brocade(config-bgp-router)#no readvertise
```

Syntax: [**no**] **readvertise**

To re-enable re-advertisement, enter the following command.

```
Brocade(config-bgp-router)#readvertise
```

Redistributing IBGP routes into RIP and OSPF

By default, the Layer 3 switch does not redistribute IBGP routes from BGP4 into RIP or OSPF. This behavior helps eliminate routing loops. However, if your network can benefit from redistributing the IBGP routes from BGP4 into OSPF or RIP, you can enable the Layer 3 switch to redistribute the routes. To do so, use the following CLI method.

To enable the Layer 3 switch to redistribute BGP4 routes into OSPF and RIP, enter the following command.

```
Brocade(config-bgp-router)#bgp-redistribute-internal
```

Syntax: [**no**] **bgp-redistribute-internal**

To disable redistribution of IBGP routes into RIP and OSPF, enter the following command.

```
Brocade(config-bgp-router)#no bgp-redistribute-internal
```

Filtering

This section describes the following:

- [“Specific IP address filtering”](#) on page 1394
- [“AS-path filtering”](#) on page 1395
- [“BGP4 filtering communities”](#) on page 1399
- [“Defining IP prefix lists”](#) on page 1401
- [“Defining neighbor distribute lists”](#) on page 1402
- [“Defining route maps”](#) on page 1403
- [“Using a table map to set the rag value”](#) on page 1411
- [“Configuring cooperative BGP4 route filtering”](#) on page 1411

Specific IP address filtering

You can configure the router to explicitly permit or deny specific IP addresses received in updates from BGP4 neighbors by defining IP address filters. The router permits all IP addresses by default. You can define up to 100 IP address filters for BGP4.

- If you want **permit** to remain the default behavior, define individual filters to deny specific IP addresses.
- If you want to change the default behavior to **deny**, define individual filters to permit specific IP addresses.

NOTE

Once you define a filter, the default action for addresses that do not match a filter is “deny”. To change the default action to “permit”, configure the last filter as “permit any any”.

Address filters can be referred to by a BGP neighbor's distribute list number as well as by match statements in a route map.

NOTE

If the filter is referred to by a route map match statement, the filter is applied in the order in which the filter is listed in the match statement.

NOTE

You also can filter on IP addresses by using IP ACLs.

To define an IP address filter to deny routes to 209.157.0.0, enter the following command.

```
Brocade(config-bgp-router)#address-filter 1 deny 209.157.0.0 255.255.0.0
```

Syntax: `address-filter <num> permit | deny <ip-addr> <wildcard> <mask> <wildcard>`

The `<num>` parameter is the filter number.

The **permit** | **deny** parameter indicates the action the Layer 3 switch takes if the filter match is true.

- If you specify **permit**, the Layer 3 switch permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the Layer 3 switch denies the route from entering the BGP4 table if the filter match is true.

NOTE

Once you define a filter, the default action for addresses that do not match a filter is “deny”. To change the default action to “permit”, configure the last filter as “permit any any”.

The `<ip-addr>` parameter specifies the IP address. If you want the filter to match on all addresses, enter **any**.

The `<wildcard>` parameter specifies the portion of the IP address to match against. The `<wildcard>` is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the `<source-ip>`. Ones mean any value matches. For example, the `<ip-addr>` and `<wildcard>` values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of “209.157.22.26 0.0.0.255” as “209.157.22.26/24”. The CLI automatically converts the CIDR number into the appropriate mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in “/`<mask-bits>`” format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the filter regardless of whether the software is configured to display the masks in CIDR format.

The `<mask>` parameter specifies the network mask. If you want the filter to match on all destination addresses, enter **any**. The wildcard works the same as described above.

AS-path filtering

You can filter updates received from BGP4 neighbors based on the contents of the AS-path list accompanying the updates. For example, if you want to deny routes that have the AS 4.3.2.1 in the AS-path from entering the BGP4 route table, you can define a filter to deny such routes.

The Layer 3 switch provides the following methods for filtering on AS-path information:

- AS-path filters
- AS-path ACLs

NOTE

The Layer 3 switch cannot actively support AS-path filters and AS-path ACLs at the same time. Use one method or the other but do not mix methods.

NOTE

Once you define a filter or ACL, the default action for updates that do not match a filter is “deny”. To change the default action to “permit”, configure the last filter or ACL as “permit any any”.

AS-path filters or AS-path ACLs can be referred to by a BGP neighbor's filter list number as well as by match statements in a route map.

Defining an AS-path filter

To define AS-path filter 4 to permit AS 2500, enter the following command.

```
Brocade(config-bgp-router)#as-path-filter 4 permit 2500
```

Syntax: `as-path-filter <num> permit | deny <as-path>`

The `<num>` parameter identifies the filter position in the AS-path filter list and can be from 1 through 100. Thus, the AS-path filter list can contain up to 100 filters. The Brocade Layer 3 switch applies the filters in numerical order, beginning with the lowest-numbered filter. When a filter match is true, the Layer 3 switch stops and does not continue applying filters from the list.

NOTE

If the filter is referred to by a route map match statement, the filter is applied in the order in which the filter is listed in the match statement.

The `permit | deny` parameter indicates the action the router takes if the filter match is true.

- If you specify **permit**, the router permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the router denies the route from entering the BGP4 table if the filter match is true.

The `<as-path>` parameter indicates the AS-path information. You can enter an exact AS-path string if you want to filter for a specific value. You also can use regular expressions in the filter string.

Defining an AS-path ACL

To configure an AS-path list that uses ACL 1, enter a command such as the following.

```
Brocade(config)#ip as-path access-list 1 permit 100
Brocade(config)#router bgp
Brocade(config-bgp-router)#neighbor 10.10.10.1 filter-list 1 in
```

The `ip as-path` command configures an AS-path ACL that permits routes containing AS number 100 in their AS paths. The `neighbor` command then applies the AS-path ACL to advertisements and updates received from neighbor 10.10.10.1. In this example, the only routes the Layer 3 switch permits from neighbor 10.10.10.1 are those whose AS-paths contain AS-path number 100.

Syntax: `ip as-path access-list <string> [seq <seq-value>] deny | permit <regular-expression>`

The `<string>` parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The `seq <seq-value>` parameter is optional and specifies the AS-path list sequence number. You can configure up to 199 entries in an AS-path list. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with number 5. The software interprets the entries in an AS-path list in numerical order, beginning with the lowest sequence number.

The `deny | permit` parameter specifies the action the software takes if a route AS-path list matches a match statement in this ACL. To configure the AS-path match statements in a route map, use the `match as-path` command. Refer to [“Matching based on AS-path ACL”](#) on page 1406.

The `<regular-expression>` parameter specifies the AS path information you want to permit or deny to routes that match any of the match statements within the ACL. You can enter a specific AS number or use a regular expression. For the regular expression syntax, refer to [“Using regular expressions to filter”](#) on page 1397.

The **neighbor** command uses the **filter-list** parameter to apply the AS-path ACL to the neighbor. Refer to [“Adding BGP4 neighbors”](#) on page 1354.

Using regular expressions to filter

You use a regular expression for the `<as-path>` parameter to specify a single character or multiple characters as a filter pattern. If the AS-path matches the pattern specified in the regular expression, the filter evaluation is true; otherwise, the evaluation is false.

In addition, you can include special characters that influence the way the software matches the AS-path against the filter value.

To filter on a specific single-character value, enter the character for the `<as-path>` parameter. For example, to filter on AS-paths that contain the letter “z”, enter the following command.

```
Brocade(config-bgp-router)#as-path-filter 1 permit z
```

To filter on a string of multiple characters, enter the characters in brackets. For example, to filter on AS-paths that contain “x”, “y”, or “z”, enter the following command.

```
Brocade(config-bgp-router)#as-path-filter 1 permit [xyz]
```

BGP4 special characters

When you enter as single-character expression or a list of characters, you also can use the following special characters. [Table 240](#) on page 1397 lists the special characters. The description for each special character includes an example. Notice that you place some special characters in front of the characters they control but you place other special characters after the characters they control. In each case, the examples show where to place the special character.

TABLE 240 BGP4 special characters for regular expressions

Character	Operation
.	The period matches on any single character, including a blank space. For example, the following regular expression matches for “aa”, “ab”, “ac”, and so on, but not just “a”. a.
*	The asterisk matches on zero or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains the string “1111” followed by any value: 1111*
+	The plus sign matches on one or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains a sequence of “g”s, such as “deg”, “degg”, “deggg”, and so on: deg+
?	The question mark matches on zero occurrences or one occurrence of a pattern. For example, the following regular expression matches on an AS-path that contains “dg” or “deg”: de?g
^	A caret (when not used within brackets) matches on the beginning of an input string. For example, the following regular expression matches on an AS-path that begins with “3”: ^3
\$	A dollar sign matches on the end of an input string. For example, the following regular expression matches on an AS-path that ends with “deg”: deg\$

TABLE 240 BGP4 special characters for regular expressions (Continued)

Character	Operation
<code>_</code>	<p>An underscore matches on one or more of the following:</p> <ul style="list-style-type: none"> • <code>,</code> (comma) • <code>{</code> (left curly brace) • <code>}</code> (right curly brace) • <code>(</code> (left parenthesis) • <code>)</code> (right parenthesis) • The beginning of the input string • The end of the input string • A blank space <p>For example, the following regular expression matches on <code>"100"</code> but not on <code>"1002"</code>, <code>"2100"</code>, and so on.</p> <pre>_100_</pre>
<code>[]</code>	<p>Square brackets enclose a range of single-character patterns. For example, the following regular expression matches on an AS-path that contains <code>"1"</code>, <code>"2"</code>, <code>"3"</code>, <code>"4"</code>, or <code>"5"</code>:</p> <pre>[1-5]</pre> <p>You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets:</p> <ul style="list-style-type: none"> • <code>^</code> – The caret matches on any characters except the ones in the brackets. For example, the following regular expression matches on an AS-path that does not contain <code>"1"</code>, <code>"2"</code>, <code>"3"</code>, <code>"4"</code>, or <code>"5"</code>: <pre>[^1-5]</pre> <ul style="list-style-type: none"> • <code>-</code> – The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. See the example above.
<code> </code>	<p>A vertical bar (sometimes called a pipe or a "logical or") separates two alternative values or sets of values. The AS-path can match one or the other value. For example, the following regular expression matches on an AS-path that contains either <code>"abc"</code> or <code>"defg"</code>:</p> <pre>(abc) (defg)</pre> <p>NOTE: The parentheses group multiple characters to be treated as one value. See the following row for more information about parentheses.</p>
<code>()</code>	<p>Parentheses allow you to create complex expressions. For example, the following complex expression matches on <code>"abc"</code>, <code>"abcabc"</code>, or <code>"abcabcabcdefg"</code>, but not on <code>"abcdefgdefg"</code>:</p> <pre>((abc)+) ((defg)?)</pre>

If you want to filter for a special character instead of using the special character as described in [Table 239](#) on page 1351, enter `"\"` (backslash) in front of the character. For example, to filter on AS-path strings containing an asterisk, enter the asterisk portion of the regular expression as `"\"`.

```
Brocade(config-bgp-router)#as-path-filter 2 deny \*
```

To use the backslash as a string character, enter two slashes. For example, to filter on AS-path strings containing a backslash, enter the backslash portion of the regular expression as `"\"`.

```
Brocade(config-bgp-router)#as-path-filter 2 deny \\
```

BGP4 filtering communities

You can filter routes received from BGP4 neighbors based on community names. Use either of the following methods to do so.

A community is an optional attribute that identifies the route as a member of a user-defined class of routes. Community names are arbitrary values made of two five-digit integers joined by a colon. You determine what the name means when you create the community name as one of a route attributes. Each string in the community name can be a number from 0 through 65535.

This format allows you to easily classify community names. For example, a common convention used in community naming is to configure the first string as the local AS and the second string as the unique community within that AS. Using this convention, communities 1:10, 1:20, and 1:30 can be easily identified as member communities of AS 1.

The Layer 3 switch provides the following methods for filtering on community information:

- Community filters
- Community list ACLs

NOTE

The Layer 3 switch cannot actively support community filters and community list ACLs at the same time. Use one method or the other but do not mix methods.

NOTE

Once you define a filter or ACL, the default action for communities that do not match a filter or ACL is “deny”. To change the default action to “permit”, configure the last filter or ACL entry as “permit any any”.

Community filters or ACLs can be referred to by match statements in a route map.

Defining a community filter

To define filter 3 to permit routes that have the NO_ADVERTISE community, enter the following command.

```
Brocade(config-bgp-router)#community-filter 3 permit no-advertise
```

Syntax: **community-filter** <num> **permit** | **deny** <num>:<num> | **internet** | **local-as** | **no-advertise** | **no-export**

The <num> parameter identifies the filter position in the community filter list and can be from 1 through 100. Thus, the community filter list can contain up to 100 filters. The router applies the filters in numerical order, beginning with the lowest-numbered filter. When a filter match is true, the router stops and does not continue applying filters from the list.

NOTE

If the filter is referred to by a route map match statement, the filter is applied in the order in which the filter is listed in the match statement.

The **permit** | **deny** parameter indicates the action the router takes if the filter match is true.

- If you specify **permit**, the router permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the router denies the route from entering the BGP4 table if the filter match is true.

The `<num>:<num>` parameter indicates a specific community number to filter. Use this parameter to filter for a private (administrator-defined) community. You can enter up to 20 community numbers with the same command.

If you want to filter for the well-known communities “LOCAL_AS”, “NO_EXPORT” or “NO_ADVERTISE”, use the corresponding keyword (described below).

The **internet** keyword checks for routes that do not have the community attribute. Routes without a specific community are considered by default to be members of the largest community, the Internet.

The **local-as** keyword checks for routes with the well-known community “LOCAL_AS”. This community applies only to confederations. The Layer 3 switch advertises the route only within the sub-AS. For information about confederations, refer to [“Configuration notes for BGP4 autonomous systems”](#) on page 1381.

The **no-advertise** keyword filters for routes with the well-known community “NO_ADVERTISE”. A route in this community should not be advertised to any BGP4 neighbors.

The **no-export** keyword filters for routes with the well-known community “NO_EXPORT”. A route in this community should not be advertised to any BGP4 neighbors outside the local AS. If the router is a member of a confederation, the Layer 3 switch advertises the route only within the confederation. For information about confederations, refer to [“Configuration notes for BGP4 autonomous systems”](#) on page 1381.

Defining a community ACL

To configure community ACL 1, enter a command such as the following.

```
Brocade(config)#ip community-list 1 permit 123:2
```

This command configures a community ACL that permits routes that contain community 123:2.

NOTE

Refer to [“Matching based on community ACL”](#) on page 1406 for information about how to use a community list as a match condition in a route map.

Syntax: `ip community-list standard <string> [seq <seq-value>] deny | permit <community-num>`

Syntax: `ip community-list extended <string> [seq <seq-value>] deny | permit <community-num> | <regular-expression>`

The `<string>` parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The **standard** or **extended** parameter specifies whether you are configuring a standard community ACL or an extended one. A standard community ACL does not support regular expressions whereas an extended one does. This is the only difference between standard and extended IP community lists.

The **seq** `<seq-value>` parameter is optional and specifies the community list sequence number. You can configure up to 199 entries in a community list. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with number 5. The software interprets the entries in a community list in numerical order, beginning with the lowest sequence number.

The **deny** | **permit** parameter specifies the action the software takes if a route community list matches a match statement in this ACL. To configure the community-list match statements in a route map, use the **match community** command. Refer to [“Matching based on community ACL”](#) on page 1406.

The `<community-num>` parameter specifies the community type or community number. This parameter can have the following values:

- `<num>:<num>` – A specific community number
- **internet** – The Internet community
- **no-export** – The community of sub-autonomous systems within a confederation. Routes with this community can be exported to other sub-autonomous systems within the same confederation but cannot be exported outside the confederation to other autonomous systems or otherwise sent to EBGp neighbors.
- **local-as** – The local sub-AS within the confederation. Routes with this community can be advertised only within the local subAS.
- **no-advertise** – Routes with this community cannot be advertised to any other BGP4 routers at all.

The `<regular-expression>` parameter specifies a regular expression for matching on community names. For information about regular expression syntax, refer to [“Using regular expressions to filter”](#) on page 1397. You can specify a regular expression only in an extended community ACL.

Defining IP prefix lists

An IP prefix list specifies a list of networks. When you apply an IP prefix list to a neighbor, the Layer 3 switch sends or receives only a route whose destination is in the IP prefix list. You can configure up to 100 prefix lists. The software interprets the prefix lists in order, beginning with the lowest sequence number.

To configure an IP prefix list and apply it to a neighbor, enter commands such as the following.

```
Brocade(config)#ip prefix-list Routesfor20 permit 20.20.0.0/24
Brocade(config)#router bgp
Brocade(config-bgp-router)#neighbor 10.10.10.1 prefix-list Routesfor20 out
```

These commands configure an IP prefix list named Routesfor20, which permits routes to network 20.20.0.0/24. The **neighbor** command configures the Layer 3 switch to use IP prefix list Routesfor20 to determine which routes to send to neighbor 10.10.10.1. The Layer 3 switch sends routes that go to 20.20.x.x to neighbor 10.10.10.1 because the IP prefix list explicitly permits these routes to be sent to the neighbor.

Syntax: `ip prefix-list <name> [seq <seq-value>] [description <string>] deny | permit <network-addr>/<mask-bits> [ge <ge-value>] [le <le-value>]`

The `<name>` parameter specifies the prefix list name. You use this name when applying the prefix list to a neighbor.

The **description** `<string>` parameter is a text string describing the prefix list.

The **seq** `<seq-value>` parameter is optional and specifies the IP prefix list sequence number. You can configure up to 100 prefix list entries. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with prefix list entry 5. The software interprets the prefix list entries in numerical order, beginning with the lowest sequence number.

The **deny** | **permit** parameter specifies the action the software takes if a neighbor route is in this prefix list.

The prefix-list matches only on this network unless you use the **ge** <ge-value> or **le** <le-value> parameters. (See below.)

The <network-addr>/<mask-bits> parameter specifies the network number and the number of bits in the network mask.

You can specify a range of prefix length for prefixes that are more specific than <network-addr>/<mask-bits>.

- If you specify only **ge** <ge-value>, then the mask-length range is from <ge-value> to 32.
- If you specify only **le** <le-value>, then the mask-length range is from length to <le-value>.

The <ge-value> or <le-value> you specify must meet the following condition.

length < ge-value <= le-value <= 32

If you do not specify **ge** <ge-value> or **le** <le-value>, the prefix list matches only on the exact network prefix you specify with the <network-addr>/<mask-bits> parameter.

For the syntax of the **neighbor** command shown in the example above, refer to [“Adding BGP4 neighbors”](#) on page 1354.

Defining neighbor distribute lists

A neighbor distribute list is a list of BGP4 address filters or ACLs that filter the traffic to or from a neighbor. To configure a neighbor distribute list, use either of the following methods.

To configure a distribute list that uses ACL 1, enter a command such as the following.

```
Brocade(config-bgp-router)#neighbor 10.10.10.1 distribute-list 1 in
```

This command configures the Layer 3 switch to use ACL 1 to select the routes that the Layer 3 switch will accept from neighbor 10.10.10.1.

Syntax: **neighbor** <ip-addr> **distribute-list** <name-or-num> **in** | **out**

The <ip-addr> parameter specifies the neighbor.

The <name-or-num> parameter specifies the name or number of a standard, extended, or named ACL.

The **in** | **out** parameter specifies whether the distribute list applies to inbound or outbound routes:

- **in** – controls the routes the Layer 3 switch will accept from the neighbor.
- **out** – controls the routes sent to the neighbor.

NOTE

The command syntax shown above is new. However, the **neighbor** <ip-addr> **distribute-list in** | **out** <num> command (where the direction is specified before the filter number) is the same as in earlier software releases. Use the new syntax when you are using an IP ACL with the distribute list. Use the old syntax when you are using a BGP4 address filter with the distribute list.

Defining route maps

A **route map** is a named set of match conditions and parameter settings that the router can use to modify route attributes and to control redistribution of the routes into other protocols. A route map consists of a sequence of up to 50 **instances**. If you think of a route map as a table, an instance is a row in that table. The router evaluates a route according to a route map instances in ascending numerical order. The route is first compared against instance 1, then against instance 2, and so on. As soon as a match is found, the router stops evaluating the route against the route map instances.

Route maps can contain **match** statements and **set** statements. Each route map contains a “permit” or “deny” action for routes that match the match statements:

- If the route map contains a permit action, a route that matches a match statement is permitted; otherwise, the route is denied.
- If the route map contains a deny action, a route that matches a match statement is denied.
- If a route does not match any match statements in the route map, the route is denied. This is the default action. To change the default action, configure the last match statement in the last instance of the route map to “permit any any”.
- If there is no match statement, the software considers the route to be a match.
- For route maps that contain address filters, AS-path filters, or community filters, if the action specified by a filter conflicts with the action specified by the route map, the route map action takes precedence over the individual filter action.

If the route map contains set statements, routes that are permitted by the route map match statements are modified according to the set statements.

Match statements compare the route against one or more of the following:

- The route BGP4 MED (metric)
- A sequence of AS-path filters
- A sequence of community filters
- A sequence of address filters
- The IP address of the next hop router
- The route tag
- For OSPF routes only, the route type (internal, external type-1, or external type-2)
- An AS-path ACL
- A community ACL
- An IP prefix list
- An IP ACL

For routes that match all of the match statements, the route map set statements can perform one or more of the following modifications to the route attributes:

- Prepend AS numbers to the front of the route AS-path. By adding AS numbers to the AS-path, you can cause the route to be less preferred when compared to other routes on the basis of the length of the AS-path.
- Add a user-defined tag to the route or add an automatically calculated tag to the route.
- Set the community value.
- Set the local preference.

- Set the MED (metric).
- Set the IP address of the next hop router.
- Set the origin to IGP or INCOMPLETE.
- Set the weight.

For example, when you configure parameters for redistributing routes into RIP, one of the optional parameters is a route map. If you specify a route map as one of the redistribution parameters, the router will match the route against the match statements in the route map. If a match is found and if the route map contains set statements, the router will set attributes in the route according to the set statements.

To create a route map, you define instances of the map. Each instance is identified by a sequence number. A route map can contain up to 50 instances.

To define a route map, use the procedures in the following sections.

Entering the route map into the software

To add instance 1 of a route map named “GET_ONE” with a permit action, enter the following command.

```
Brocade(config)#route-map GET_ONE permit 1
Brocade(config-route-map GET_ONE)#
```

Syntax: [no] route-map <map-name> permit | deny <num>

As shown in this example, the command prompt changes to the Route Map level. You can enter the match and set statements at this level. Refer to [“Specifying the match conditions”](#) on page 1405 and [“Setting parameters in the routes”](#) on page 1408.

The <map-name> is a string of characters that names the map. Map names can be up to 32 characters in length.

The **permit** | **deny** parameter specifies the action the router will take if a route matches a match statement.

- If you specify **deny**, the Layer 3 switch does not advertise or learn the route.
- If you specify **permit**, the Layer 3 switch applies the match and set statements associated with this route map instance.

The <num> parameter specifies the instance of the route map you are defining. Each route map can have up to 50 instances.

To delete a route map, enter a command such as the following. When you delete a route map, all the permit and deny entries in the route map are deleted.

```
Brocade(config)#no route-map Map1
```

This command deletes a route map named “Map1”. All entries in the route map are deleted.

To delete a specific instance of a route map without deleting the rest of the route map, enter a command such as the following.

```
Brocade(config)#no route-map Map1 permit 10
```

This command deletes the specified instance from the route map but leaves the other instances of the route map intact.

Specifying the match conditions

Use the following command to define the match conditions for instance 1 of the route map GET_ONE. This instance compares the route updates against BGP4 address filter 11.

```
Brocade(config-routemap GET_ONE)#match address-filters 11
```

Syntax: **match** [**as-path** <num>] | [**address-filters** | **as-path-filters** | **community-filters** <num,num,...>] | [**community** <num>] | [**community** <ACL> **exact-match**] | [**ip address** <ACL> | **prefix-list** <string>] | [**ip route-source** <ACL> | **prefix** <name>] [**metric** <num>] | [**next-hop** <address-filter-list>] | [**nlri multicast** | **unicast** | **multicast unicast**] | [**route-type internal** | **external-type1** | **external-type2**] | [**tag** <tag-value>]

The **as-path** <num> parameter specifies an AS-path ACL. You can specify up to five AS-path ACLs. To configure an AS-path ACL, use the **ip as-path access-list** command. Refer to [“Defining an AS-path ACL”](#) on page 1396.

The **address-filters** | **as-path-filters** | **community-filters** <num,num,...> parameter specifies a filter or list of filters to be matched for each route. The router treats the first match as the best match. If a route does not match any filter in the list, then the router considers the match condition to have failed. To configure these types of filters, use commands at the BGP configuration level:

- To configure an address filter, refer to [“Specific IP address filtering”](#) on page 1394.
- To configure an AS-path filter or AS-path ACL, refer to [“AS-path filtering”](#) on page 1395.
- To configure a community filter or community ACL, refer to [“BGP4 filtering communities”](#) on page 1399.

You can enter up to six community names on the same command line.

NOTE

The filters must already be configured.

The **community** <num> parameter specifies a community ACL.

NOTE

The ACL must already be configured.

The **community** <ACL> **exact-match** parameter matches a route if (and only if) the route's community attributes field contains the same community numbers specified in the match statement.

The **ip address** | **next-hop** <ACL-num> | **prefix-list** <string> parameter specifies an ACL or IP prefix list. Use this parameter to match based on the destination network or next-hop gateway. To configure an IP ACL for use with this command, use the **ip access-list** command. Refer to [“ACL overview”](#) on page 1705. To configure an IP prefix list, use the **ip prefix-list** command. Refer to [“Defining IP prefix lists”](#) on page 1401.

The **ip route-source** <ACL> | **prefix** <name> parameter matches based on the source of a route (the IP address of the neighbor from which the Brocade device learned the route).

The **metric** <num> parameter compares the route MED (metric) to the specified value.

The **next-hop** <address-filter-list> parameter compares the IP address of the route next hop to the specified IP address filters. The filters must already be configured.

The **nlri multicast** | **unicast** | **multicast unicast** parameter specifies whether you want the route map to match on multicast routes, unicast routes, or both route types.

NOTE

By default, route maps apply to both unicast and multicast traffic.

The **route-type internal | external-type1 | external-type2** parameter applies only to OSPF routes. This parameter compares the route type to the specified value.

The **tag <tag-value>** parameter compares the route tag to the specified value.

Match examples using ACLs

The following sections show some detailed examples of how to configure route maps that include match statements that match on ACLs.

Matching based on AS-path ACL

To construct a route map that matches based on AS-path ACL 1, enter the following commands.

```
Brocade(config)#route-map PathMap permit 1
Brocade(config-routemap PathMap)#match as-path 1
```

Syntax: **match as-path <num>**

The **<num>** parameter specifies an AS-path ACL and can be a number from 1 through 199. You can specify up to five AS-path ACLs. To configure an AS-path ACL, use the **ip as-path access-list** command. Refer to [“Defining an AS-path ACL”](#) on page 1396.

Matching based on community ACL

To construct a route map that matches based on community ACL 1, enter the following commands.

```
Brocade(config)#ip community-list 1 permit 123:2
Brocade(config)#route-map CommMap permit 1
Brocade(config-routemap CommMap)#match community 1
```

Syntax: **match community <string>**

The **<string>** parameter specifies a community list ACL. To configure a community list ACL, use the **ip community-list** command. Refer to [“Defining a community ACL”](#) on page 1400.

Matching based on destination network

To construct match statements for a route map that match based on destination network, use the following method. You can use the results of an IP ACL or an IP prefix list as the match condition.

```
Brocade(config)#route-map NetMap permit 1
Brocade(config-routemap NetMap)#match ip address 1
```

Syntax: **match ip address <name-or-num>**

Syntax: **match ip address prefix-list <name>**

The **<name-or-num>** parameter with the first command specifies an IP ACL and can be a number from 1 through 199 or the ACL name if it is a named ACL. To configure an IP ACL, use the **ip access-list** or **access-list** command. Refer to [Chapter 40, “Rule-Based IP ACLs”](#).

The **<name>** parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, refer to [“Defining IP prefix lists”](#) on page 1401.

Matching based on next-hop router

To construct match statements for a route map that match based on the IP address of the next-hop router, use either of the following methods. You can use the results of an IP ACL or an IP prefix list as the match condition.

To construct a route map that matches based on the next-hop router, enter commands such as the following.

```
Brocade(config)#route-map HopMap permit 1
Brocade(config-route-map HopMap)#match ip next-hop 2
```

Syntax: `match ip next-hop <num>`

Syntax: `match ip next-hop prefix-list <name>`

The `<num>` parameter with the first command specifies an IP ACL and can be a number from 1 through 199 or the ACL name if it is a named ACL. To configure an IP ACL, use the `ip access-list` or `access-list` command. Refer to [Chapter 40, “Rule-Based IP ACLs”](#).

The `<name>` parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, refer to [“Defining IP prefix lists”](#) on page 1401.

Matching based on the route source

To match a BGP4 route based on its source, use the `match ip route-source` statement. Here is an example.

```
Brocade(config)#access-list 10 permit 192.168.6.0 0.0.0.255
Brocade(config)#route-map bgp1 permit 1
Brocade(config-route-map bgp1)#match ip route-source 10
```

The first command configures an IP ACL that matches on routes received from 192.168.6.0/24. The remaining commands configure a route map that matches on all BGP4 routes advertised by the BGP4 neighbors whose addresses match addresses in the IP prefix list. You can add a set statement to change a route attribute in the routes that match. You also can use the route map as input for other commands, such as the `neighbor` and `network` commands and some `show` commands.

Syntax: `match ip route-source <ACL> | prefix <name>`

The `<ACL> | prefix <name>` parameter specifies the name or ID of an IP ACL, or an IP prefix list.

Matching on routes containing a specific set of communities

Brocade software enables you to match routes based on the presence of a community name or number in a route, and to match when a route contains exactly the set of communities you specify. To match based on a set of communities, configure a community ACL that lists the communities, then compare routes against the ACL, as shown in the following example.

```
Brocade(config)#ip community-list standard std_1 permit 12:34 no-export
Brocade(config)#route-map bgp2 permit 1
Brocade(config-route-map bgp2)#match community std_1 exact-match
```

The first command configures a community ACL that contains community number 12:34 and community name no-export. The remaining commands configure a route map that matches the community attributes field in BGP4 routes against the set of communities in the ACL. A route matches the route map only if the route contains all the communities in the ACL and no other communities.

Syntax: `match community <ACL> exact-match`

The `<ACL>` parameter specifies the name of a community list ACL. You can specify up to five ACLs. Separate the ACL names or IDs with spaces.

Here is another example.

```
Brocade(config)#ip community-list standard std_2 permit 23:45 56:78
Brocade(config)#route-map bgp3 permit 1
Brocade(config-route-map bgp3)#match community std_1 std_2 exact-match
```

These commands configure an additional community ACL, `std_2`, that contains community numbers 23:45 and 57:68. Route map `bgp3` compares each BGP4 route against the sets of communities in ACLs `std_1` and `std_2`. A BGP4 route that contains **either but not both** sets of communities matches the route map. For example, a route containing communities 23:45 and 57:68 matches. However, a route containing communities 23:45, 57:68 and 12:34, or communities 23:45, 57:68, 12:34, and no-export does not match. To match, the route communities must be the same as those in exactly one of the community ACLs used by the match community statement.

Setting parameters in the routes

Use the following command to define a set statement that prepends an AS number to the AS path on each route that matches the corresponding match statement.

```
Brocade(config-route-map GET_ONE)#set as-path prepend 65535
```

Syntax: `set [as-path [prepend <as-num,as-num,...>]] | [automatic-tag] | [comm-list <ACL> delete] | [community <num>:<num> | <num> | internet | local-as | no-advertise | no-export] | [dampening [<half-life> <reuse> <suppress> <max-suppress-time>]] [[default] interface null0 | [ip [default] next hop <ip-addr>] [ip next-hop peer-address] | [local-preference <num>] | [metric [+ | -]<num> | none] | [metric-type type-1 | type-2] | [metric-type internal] | [next-hop <ip-addr>] | [nlri multicast | unicast | multicast unicast] | [origin igp | incomplete] | [tag <tag-value>] | [weight <num>]`

The **as-path prepend** `<num,num,...>` parameter adds the specified AS numbers to the front of the AS-path list for the route.

The **automatic-tag** parameter calculates and sets an automatic tag value for the route.

NOTE

This parameter applies only to routes redistributed into OSPF.

The **comm-list** parameter deletes a community from a BGP4 route community attributes field.

The **community** parameter sets the community attribute for the route to the number or well-known type you specify.

The **dampening** `[<half-life> <reuse> <suppress> <max-suppress-time>]` parameter sets route dampening parameters for the route. The `<half-life>` parameter specifies the number of minutes after which the route penalty becomes half its value. The `<reuse>` parameter specifies how low a route penalty must become before the route becomes eligible for use again after being suppressed. The `<suppress>` parameter specifies how high a route penalty can become before the Layer 3 switch suppresses the route. The `<max-suppress-time>` parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is. For information and examples, refer to [“Route flap dampening configuration”](#) on page 1414.

The **[default] interface null0** parameter redirects the traffic to the specified interface. You can send the traffic to the null0 interface, which is the same as dropping the traffic. You can specify more than one interface, in which case the Layer 3 switch uses the first available port. If the first port is unavailable, the Layer 3 switch sends the traffic to the next port in the list. If you specify **default**, the route map redirects the traffic to the specified interface only if the Layer 3 switch does not already have explicit routing information for the traffic. This option is used in Policy-Based Routing (PBR).

The **ip [default] next hop <ip-addr>** parameter sets the next-hop IP address for traffic that matches a match statement in the route map. If you specify **default**, the route map sets the next-hop gateway only if the Layer 3 switch does not already have explicit routing information for the traffic. This option is used in Policy-Based Routing (PBR).

The **ip next-hop peer-address** parameter sets the BGP4 next hop for a route to the specified neighbor address.

The **local-preference <num>** parameter sets the local preference for the route. You can set the preference to a value from 0 through 4294967295.

The **metric [+ | -]<num> | none** parameter sets the MED (metric) value for the route. The default MED value is 0. You can set the preference to a value from 0 through 4294967295.

- **set metric <num>** – Sets the route metric to the number you specify.
- **set metric +<num>** – Increases route metric by the number you specify.
- **set metric -<num>** – Decreases route metric by the number you specify.
- **set metric none** – Removes the metric from the route (removes the MED attribute from the BGP4 route).

The **metric-type type-1 | type-2** parameter changes the metric type of a route redistributed into OSPF.

The **metric-type internal** parameter sets the route's MED to the same value as the IGP metric of the BGP4 next-hop route. The parameter does this when advertising a BGP4 route to an EBGp neighbor.

The **next-hop <ip-addr>** parameter sets the IP address of the route next hop router.

The **nlri multicast | unicast | multicast unicast** parameter redistributes routes into the multicast Routing Information Base (RIB) instead of the unicast RIB.

NOTE

Setting the NLRI type to multicast applies only when you are using the route map to redistribute directly-connected routes. Otherwise, the set option is ignored.

The **origin igp | incomplete** parameter sets the route origin to IGP or INCOMPLETE.

The **tag <tag-value>** parameter sets the route tag. You can specify a tag value from 0 through 4294967295.

NOTE

This parameter applies only to routes redistributed into OSPF.

NOTE

You also can set the tag value using a table map. The table map changes the value only when the Layer 3 switch places the route in the IP route table instead of changing the value in the BGP route table. Refer to [“Using a table map to set the tag value”](#) on page 1411.

The **weight** <num> parameter sets the weight for the route. You can specify a weight value from 0 through 4294967295.

Setting a BGP4 route MED to the same value as the IGP metric of the next-hop route

To set a route's MED to the same value as the IGP metric of the BGP4 next-hop route, when advertising the route to a neighbor, enter commands such as the following.

```
Brocade(config)#access-list 1 permit 192.168.9.0 0.0.0.255
Brocade(config)#route-map bgp4 permit 1
Brocade(config-route-map bgp4)#match ip address 1
Brocade(config-route-map bgp4)#set metric-type internal
```

The first command configures an ACL that matches on routes with destination network 192.168.9.0. The remaining commands configure a route map that matches on the destination network in ACL 1, then sets the metric type for those routes to the same value as the IGP metric of the BGP4 next-hop route.

Syntax: **set metric-type internal**

Setting the next hop of a BGP4 route

To set the next hop address of a BGP4 route to a neighbor address, enter commands such as the following.

```
Brocade(config)#route-map bgp5 permit 1
Brocade(config-route-map bgp5)#match ip address 1
Brocade(config-route-map bgp5)#set ip next-hop peer-address
```

These commands configure a route map that matches on routes whose destination network is specified in ACL 1, and sets the next hop in the routes to the neighbor address (inbound filtering) or the local IP address of the BGP4 session (outbound filtering).

Syntax: **set ip next-hop peer-address**

The value that the software substitutes for **peer-address** depends on whether the route map is used for inbound filtering or outbound filtering:

- When you use the **set ip next-hop peer-address** command in an inbound route map filter, **peer-address** substitutes for the neighbor IP address.
- When you use the **set ip next-hop peer-address** command in an outbound route map filter, **peer-address** substitutes for the local IP address of the BGP4 session.

NOTE

You can use this command for a peer group configuration.

Deleting a community from a BGP4 route

To delete a community from a BGP4 route community attributes field, enter commands such as the following.

```
Brocade(config)#ip community-list standard std_3 permit 12:99 12:86
Brocade(config)#route-map bgp6 permit 1
Brocade(config-route-map bgp6)#match ip address 1
Brocade(config-route-map bgp6)#set comm-list std_3 delete
```


The first command configures a community ACL containing community numbers 12:99 and 12:86. The remaining commands configure a route map that matches on routes whose destination network is specified in ACL 1, and deletes communities 12:99 and 12:86 from those routes. The route does not need to contain all the specified communities in order for them to be deleted. For example, if a route contains communities 12:86, 33:44, and 66:77, community 12:86 is deleted.

Syntax: `set comm-list <ACL> delete`

The <ACL> parameter specifies the name of a community list ACL.

Using a table map to set the tag value

Route maps that contain set statements change values in routes when the routes are accepted by the route map. For inbound route maps (route maps that filter routes received from neighbors), this means that the routes are changed before they enter the BGP4 route table.

For tag values, if you do not want the value to change until a route enters the IP route table, you can use a table map to change the value. A table map is a route map that you have associated with the IP routing table. The Layer 3 switch applies the set statements for tag values in the table map to routes before adding them to the route table.

To configure a table map, you configure the route map, then identify it as a table map. The table map does not require separate configuration. You create it simply by calling an existing route map a table map. You can have one table map.

NOTE

Use table maps only for setting the tag value. Do not use table maps to set other attributes. To set other route attributes, use route maps or filters.

To create a route map and identify it as a table map, enter commands such as following. These commands create a route map that uses an address filter. For routes that match the address filter, the route map changes the tag value to 100. This route map is then identified as a table map. As a result, the route map is applied only to routes that the Layer 3 switch places in the IP route table. The route map is not applied to all routes. This example assumes that address filter 11 has already been configured.

```
Brocade(config)#route-map TAG_IP permit 1
Brocade(config-routemap TAG_IP)#match address-filters 11
Brocade(config-routemap TAG_IP)#set tag 100
Brocade(config-routemap TAG_IP)#router bgp
Brocade(config-bgp-router)#table-map TAG_IP
```

Configuring cooperative BGP4 route filtering

By default, the Layer 3 switch performs all filtering of incoming routes locally, on the Layer 3 switch itself. You can use cooperative BGP4 route filtering to cause the filtering to be performed by a neighbor before it sends the routes to the Layer 3 switch. Cooperative filtering conserves resources by eliminating unnecessary route updates and filter processing. For example, the Layer 3 switch can send a deny filter to its neighbor, which the neighbor uses to filter out updates before sending them to the Layer 3 switch. The neighbor saves the resources it would otherwise use to generate the route updates, and the Layer 3 switch saves the resources it would use to filter out the routes.

When you enable cooperative filtering, the Layer 3 switch advertises this capability in its Open message to the neighbor when initiating the neighbor session. The Open message also indicates whether the Layer 3 switch is configured to send filters, receive filters or both, and the types of filters it can send or receive. The Layer 3 switch sends the filters as Outbound Route Filters (ORFs) in Route Refresh messages.

To configure cooperative filtering, perform the following tasks on the Layer 3 switch and on its BGP4 neighbor:

- Configure the filter.

NOTE

The current release supports cooperative filtering only for filters configured using IP prefix lists.

- Apply the filter as in *inbound* filter to the neighbor.
- Enable the cooperative route filtering feature on the Layer 3 switch. You can enable the Layer 3 switch to send ORFs to the neighbor, to receive ORFs from the neighbor, or both. The neighbor uses the ORFs you send as outbound filters when it sends routes to the Layer 3 switch. Likewise, the Layer 3 switch uses the ORFs it receives from the neighbor as outbound filters when sending routes to the neighbor.
- Reset the BGP4 neighbor session to send and receive ORFs.
- Perform these steps on the other device.

NOTE

If the Layer 3 switch has inbound filters, the filters are still processed even if equivalent filters have been sent as ORFs to the neighbor.

Enabling cooperative filtering

To configure cooperative filtering, enter commands such as the following.

```
Brocade(config)#ip prefix-list Routesfrom1234 deny 20.20.0.0/24
Brocade(config)#ip prefix-list Routesfrom1234 permit 0.0.0.0/0 le 32
Brocade(config)#router bgp
Brocade(config-bgp-router)#neighbor 1.2.3.4 prefix-list Routesfrom1234 in
Brocade(config-bgp-router)#neighbor 1.2.3.4 capability orf prefixlist send
```

The first two commands configure statements for the IP prefix list Routesfrom1234. The first command configures a statement that denies routes to 20.20.20./24. The second command configures a statement that permits all other routes. (Once you configure an IP prefix list statement, all routes not explicitly permitted by statements in the prefix list are denied.)

The next two commands change the CLI to the BGP4 configuration level, then apply the IP prefix list to neighbor 1.2.3.4. The last command enables the Layer 3 switch to send the IP prefix list as an ORF to neighbor 1.2.3.4. When the Layer 3 switch sends the IP prefix list to the neighbor, the neighbor filters out the 20.20.0.x routes from its updates to the Layer 3 switch. (This assumes that the neighbor also is configured for cooperative filtering.)

The *<ip-addr> | <peer-group-name>* parameter specifies the IP address of a neighbor or the name of a peer group of neighbors.

The **send** | **receive** parameter specifies the support you are enabling:

- **send** – The Layer 3 switch sends the IP prefix lists to the neighbor.
- **receive** – The Layer 3 switch accepts filters from the neighbor.

If you do not specify the capability, both capabilities are enabled.

The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

NOTE

The current release supports cooperative filtering only for filters configured using IP prefix lists.

Sending and receiving ORFs

Cooperative filtering affects neighbor sessions that start after the filtering is enabled, but do not affect sessions that are already established.

To activate cooperative filtering, reset the session with the neighbor. This is required because the cooperative filtering information is exchanged in Open messages during the start of a session.

To place a prefix-list change into effect after activating cooperative filtering, perform a soft reset of the neighbor session. A soft reset does not end the current session, but sends the prefix list to the neighbor in the next route refresh message.

NOTE

Make sure cooperative filtering is enabled on the Layer 3 switch and on the neighbor before you send the filters.

To reset a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
Brocade#clear ip bgp neighbor 1.2.3.4
```

This command resets the BGP4 session with neighbor 1.2.3.4 and sends the ORFs to the neighbor. If the neighbor sends ORFs to the Layer 3 switch, the Layer 3 switch accepts them if the send capability is enabled.

To perform a soft reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
Brocade#clear ip bgp neighbor 1.2.3.4 soft in prefix-list
```

Syntax: **clear ip bgp neighbor** <ip-addr> [**soft in prefix-filter**]

If you use the **soft in prefix-filter** parameter, the Layer 3 switch sends the updated IP prefix list to the neighbor as part of its route refresh message to the neighbor.

NOTE

If the Layer 3 switch or the neighbor is not configured for cooperative filtering, the command sends a normal route refresh message.

Displaying cooperative filtering information

You can display the following cooperative filtering information:

- The cooperative filtering configuration on the Layer 3 switch.
- The ORFs received from neighbors.

To display the cooperative filtering configuration on the Layer 3 switch, enter a command such as the following. The line shown in bold type shows the cooperative filtering status.

Route flap dampening configuration

```
Brocade#show ip bgp neighbors 10.10.10.1
1  IP Address: 10.10.10.1, AS: 65200 (IBGP), RouterID: 10.10.10.1
   State: ESTABLISHED, Time: 0h0m7s, KeepAliveTime: 60, HoldTime: 180
   RefreshCapability: Received
   CooperativeFilteringCapability: Received
   Messages:      Open      Update  KeepAlive  Notification  Refresh-Req
     Sent       : 1         0        1           0             1
     Received: 1         0        1           0             1
   Last Update Time: NLRI      Withdraw      NLRI      Withdraw
                   Tx: ---      ---            Rx: ---      ---
   Last Connection Reset Reason: Unknown
   Notification Sent:      Unspecified
   Notification Received: Unspecified
   TCP Connection state: ESTABLISHED
   Byte Sent: 110, Received: 110
   Local host: 10.10.10.2, Local Port: 8138
   Remote host: 10.10.10.1, Remote Port: 179
   ISentSeq:      460  SendNext:      571  TotUnAck:      0
   TotSent:      111  ReTrans:      0    UnAckSeq:      571
   IRcvSeq:      7349 RcvNext:      7460  SendWnd:      16384
   TotalRcv:      111 DupliRcv:      0    RcvWnd:      16384
   SendQue:      0    RcvQue:      0    CngstWnd:      5325
```

Syntax: show ip bgp neighbors <ip-addr>

To display the ORFs received from a neighbor, enter a command such as the following.

```
Brocade#show ip bgp neighbors 10.10.10.1 received prefix-filter
ip prefix-list 10.10.10.1: 4 entries
  seq 5 permit 10.10.0.0/16 ge 18 le 28
  seq 10 permit 20.20.10.0/24
  seq 15 permit 30.0.0.0/8 le 32
  seq 20 permit 40.10.0.0/16 ge 18
```

Syntax: show ip bgp neighbors <ip-addr> received prefix-filter

Route flap dampening configuration

A “route flap” is the change in a route state, from up to down or down to up. When a route state changes, the state change causes changes in the route tables of the routers that support the route. Frequent changes in a route state can cause Internet instability and add processing overhead to the routers that support the route.

Route flap dampening is a mechanism that reduces the impact of route flap by changing a BGP4 router response to route state changes. When route flap dampening is configured, the Layer 3 switch suppresses unstable routes until the route state changes reduce enough to meet an acceptable degree of stability. The Brocade implementation of route flap dampening is based on RFC 2439.

Route flap dampening is disabled by default. You can enable the feature globally or on an individual route basis using route maps.

NOTE

The Layer 3 switch applies route flap dampening only to routes learned from EBGp neighbors.

The route flap dampening mechanism is based on penalties. When a route exceeds a configured penalty value, the Layer 3 switch stops using that route and also stops advertising it to other routers. The mechanism also allows a route penalties to reduce over time if the route stability improves. The route flap dampening mechanism uses the following parameters:

- **Suppression threshold** – Specifies the penalty value at which the Layer 3 switch stops using the route. Each time a route becomes unreachable or is withdrawn by a BGP4 UPDATE from a neighbor, the route receives a penalty of 1000. By default, when a route has a penalty value greater than 2000, the Layer 3 switch stops using the route. Thus, by default, if a route goes down more than twice, the Layer 3 switch stops using the route. You can set the suppression threshold to a value from 1 through 20000. The default is 2000.
- **Half-life** – Once a route has been assigned a penalty, the penalty decreases exponentially and decreases by half after the half-life period. The default half-life period is 15 minutes. The software reduces route penalties every five seconds. For example, if a route has a penalty of 2000 and does not receive any more penalties (it does not go down again) during the half-life, the penalty is reduced to 1000 after the half-life expires. You can configure the half-life to be from 1 through 45 minutes. The default is 15 minutes.
- **Reuse threshold** – Specifies the minimum penalty a route can have and still be suppressed by the Layer 3 switch. If the route's penalty falls below this value, the Layer 3 switch un-suppresses the route and can use it again. The software evaluates the dampened routes every ten seconds and un-suppresses the routes that have penalties below the reuse threshold. You can set the reuse threshold to a value from 1 through 20000. The default is 750.
- **Maximum suppression time** – Specifies the maximum number of minutes a route can be suppressed regardless of how unstable the route has been before this time. You can set the parameter to a value from 1 through 20000 minutes. The default is four times the half-life. When the half-life value is set to its default (15 minutes), the maximum suppression time defaults to 60 minutes.

You can configure route flap dampening globally or for individual routes using route maps. If you configure route flap dampening parameters globally and also use route maps, the settings in the route maps override the global values.

Globally configuring route flap dampening

To enable route flap dampening using the default values, enter the following command.

```
Brocade(config-bgp-router)#dampening
```

Syntax: `dampening [<half-life> <reuse> <suppress> <max-suppress-time>]`

The `<half-life>` parameter specifies the number of minutes after which the route penalty becomes half its value. The route penalty allows routes that have remained stable for a while despite earlier instability to eventually become eligible for use again. The decay rate of the penalty is proportional to the value of the penalty. After the half-life expires, the penalty decays to half its value. Thus, a dampened route that is no longer unstable can eventually become eligible for use again. You can configure the half-life to be from 1 - 45 minutes. The default is 15 minutes.

The `<reuse>` parameter specifies how low a route penalty must become before the route becomes eligible for use again after being suppressed. You can set the reuse threshold to a value from 1 through 20000. The default is 750 (0.75, or three-fourths, of the penalty assessed for a one "flap").

The `<suppress>` parameter specifies how high a route penalty can become before the Layer 3 switch suppresses the route. You can set the suppression threshold to a value from 1 through 20000. The default is 2000 (two “flaps”).

The `<max-suppress-time>` parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is. You can set the maximum suppression time to a value from 1 through 20000 minutes. The default is four times the half-life setting. Thus, if you use the default half-life of 15 minutes, the maximum suppression time is 60 minutes.

The following example shows how to change the dampening parameters.

```
Brocade(config-bgp-router)#dampening 20 200 2500 40
```

This command changes the half-life to 20 minutes, the reuse threshold to 200, the suppression threshold to 2500, and the maximum number of minutes a route can be dampened to 40.

NOTE

To change any of the parameters, you must specify all the parameters with the command. If you want to leave some parameters unchanged, enter their default values.

Using a route map to configure route flap dampening for specific routes

Route maps enable you to fine tune route flap dampening parameters for individual routes. To configure route flap dampening parameters using route maps, configure BGP4 address filters for each route you want to set the dampening parameters for, then configure route map entries that set the dampening parameters for those routes. The following sections show examples.

To configure address filters and a route map for dampening specific routes, enter commands such as the following.

```
Brocade(config)#router bgp
Brocade(config-bgp-router)#address-filter 9 permit 209.157.22.0 255.255.255.0
255.255.255.0 255.255.255.0
Brocade(config-bgp-router)#address-filter 10 permit 209.157.23.0 255.255.255.0
255.255.255.0 255.255.255.0
Brocade(config-bgp-router)#exit
Brocade(config)#route-map DAMPENING_MAP permit 9
Brocade(config-routemap DAMPENING_MAP)#match address-filters 9
Brocade(config-routemap DAMPENING_MAP)#set dampening 10 200 2500 40
Brocade(config-routemap DAMPENING_MAP)#exit
Brocade(config)#route-map DAMPENING_MAP permit 10
Brocade(config-routemap DAMPENING_MAP)#match address-filters 10
Brocade(config-routemap DAMPENING_MAP)#set dampening 20 200 2500 60
Brocade(config-routemap DAMPENING_MAP)#router bgp
Brocade(config-bgp-router)#dampening route-map DAMPENING_MAP
```

The **address-filter** commands in this example configure two BGP4 address filters, for networks 209.157.22.0 and 209.157.23.0. The first route-map command creates an entry in a route map called “DAMPENING_MAP”. Within this entry of the route map, the **match** command matches based on address filter 9, and the **set** command sets the dampening parameters for the route that matches. Thus, for BGP4 routes to 209.157.22.0, the Layer 3 switch uses the route map to set the dampening parameters. These parameters override the globally configured dampening parameters.

The commands for the second entry in the route map (instance 10 in this example) perform the same functions for route 209.157.23.0. Notice that the dampening parameters are different for each route.

Using a route map to configure route flap dampening for a specific neighbor

You can use a route map to configure route flap dampening for a specific neighbor by performing the following tasks:

- Configure an empty route map with no match or set statements. This route map does not specify particular routes for dampening but does allow you to enable dampening globally when you refer to this route map from within the BGP configuration level.
- Configure another route map that explicitly enables dampening. Use a set statement within the route map to enable dampening. When you associate this route map with a specific neighbor, the route map enables dampening for all routes associated with the neighbor. You also can use match statements within the route map to selectively perform dampening on some routes from the neighbor.

NOTE

You still need to configure the first route map to enable dampening globally. The second route map does not enable dampening by itself; it just applies dampening to a neighbor.

- Apply the route map to the neighbor.

To enable route flap dampening for a specific BGP4 neighbor, enter commands such as the following.

```
Brocade(config)#route-map DAMPENING_MAP_ENABLE permit 1
Brocade(config-route-map DAMPENING_MAP_ENABLE)#exit
Brocade(config)#route-map DAMPENING_MAP_NEIGHBOR_A permit 1
Brocade(config-route-map DAMPENING_MAP_NEIGHBOR_A)#set dampening
Brocade(config-route-map DAMPENING_MAP_NEIGHBOR_A)#exit
Brocade(config)#router bgp
Brocade(config-bgp-router)#dampening route-map DAMPENING_MAP_ENABLE
Brocade(config-bgp-router)#neighbor 10.10.10.1 route-map in
DAMPENING_MAP_NEIGHBOR_A
```

In this example, the first command globally enables route flap dampening. This route map does not contain any match or set statements. At the BGP configuration level, the **dampening route-map** command refers to the DAMPENING_MAP_ENABLE route map created by the first command, thus enabling dampening globally.

The third and fourth commands configure a second route map that explicitly enables dampening. Notice that the route map does not contain a match statement. The route map implicitly applies to all routes. Since the route map will be applied to a neighbor at the BGP configuration level, the route map will apply to all routes associated with the neighbor.

Although the second route map enables dampening, the first route map is still required. The second route map enables dampening for the neighbors to which the route map is applied. However, unless dampening is already enabled globally by the first route map, the second route map has no effect.

The last two commands apply the route maps. The **dampening route-map** command applies the first route map, which enables dampening globally. The **neighbor** command applies the second route map to neighbor 10.10.10.1. Since the second route map does not contain match statements for specific routes, the route map enables dampening for all routes received from the neighbor.

Removing route dampening from a route

You can un-suppress routes by removing route flap dampening from the routes. The Layer 3 switch allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI.

```
Brocade#clear ip bgp damping
```

Syntax: `clear ip bgp damping [<ip-addr> <ip-mask>]`

The `<ip-addr>` parameter specifies a particular network.

The `<ip-mask>` parameter specifies the network mask.

To un-suppress a specific route, enter a command such as the following.

```
Brocade#clear ip bgp damping 209.157.22.0 255.255.255.0
```

This command un-suppresses only the routes for network 209.157.22.0/24.

Removing route dampening from a neighbor routes suppressed due to aggregation

You can selectively unsuppress more-specific routes that have been suppressed due to aggregation, and allow the routes to be advertised to a specific neighbor or peer group.

Here is an example.

```
Brocade(config-bgp-router)#aggregate-address 209.1.0.0 255.255.0.0 summary-only
Brocade(config-bgp-router)#show ip bgp route 209.1.0.0/16 longer
Number of BGP Routes matching display condition : 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
      Prefix          Next Hop          Metric      LocPrf      Weight Status
1      209.1.0.0/16    0.0.0.0              101          32768    BAL
      AS_PATH:
2      209.1.44.0/24   10.2.0.1              1            101          32768    BLS
      AS_PATH:
```

The **aggregate-address** command configures an aggregate address. The **summary-only** parameter prevents the Layer 3 switch from advertising more specific routes contained within the aggregate route. The **show ip bgp route** command shows that the more specific routes aggregated into 209.1.0.0/16 have been suppressed. In this case, the route to 209.1.44.0/24 has been suppressed. The following command indicates that the route is not being advertised to the Layer 3 switch BGP4 neighbors.


```

Brocade#show ip bgp route 209.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
      Prefix          Next Hop      Metric      LocPrf      Weight Status
1      209.1.44.0/24    10.2.0.1      1           101         32768   BLS
      AS_PATH:
      Route is not advertised to any peers

```

If you want to override the **summary-only** parameter and allow a specific route to be advertised to a neighbor, enter commands such as the following.

```

Brocade(config)#ip prefix-list Unsuppress1 permit 209.1.44.0/24
Brocade(config)#route-map RouteMap1 permit 1
Brocade(config-routemap RouteMap1)#match ip prefix-list Unsuppress1
Brocade(config-routemap RouteMap1)#exit
Brocade(config)#router bgp
Brocade(config-bgp-router)#neighbor 10.1.0.2 unsuppress-map RouteMap1
Brocade(config-bgp-router)#clear ip bgp neighbor 10.1.0.2 soft-out

```

The **ip prefix-list** command configures an IP prefix list for network 209.1.44.0/24, which is the route you want to unsuppress. The next two commands configure a route map that uses the prefix list as input. The **neighbor** command enables the Layer 3 switch to advertise the routes specified in the route map to neighbor 10.1.0.2. The **clear** command performs a soft reset of the session with the neighbor so that the Layer 3 switch can advertise the unsuppressed route.

Syntax: [no] **neighbor** <ip-addr> | <peer-group-name> **unsuppress-map** <map-name>

The following command verifies that the route has been unsuppressed.

```

Brocade#show ip bgp route 209.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
      Prefix          Next Hop      Metric      LocPrf      Weight Status
1      209.1.44.0/24    10.2.0.1      1           101         32768   BLS
      AS_PATH:
      Route is advertised to 1 peers:
      10.1.0.2(4)

```

Displaying and clearing route flap dampening statistics

The software provides many options for displaying and clearing route flap statistics. To display the statistics, use either of the following methods.

Displaying route flap dampening statistics

To display route dampening statistics or all the dampened routes, enter the **show ip bgp flap-statistics** command at any level of the CLI.

Route flap dampening configuration

```
Brocade#show ip bgp flap-statistics
Total number of flapping routes: 414
      Status Code  >:best d:damped h:history *:valid
      Network      From      Flaps Since      Reuse      Path
h> 192.50.206.0/23 166.90.213.77 1      0 :0 :13 0 :0 :0 65001 4355 1 701
h> 203.255.192.0/20 166.90.213.77 1      0 :0 :13 0 :0 :0 65001 4355 1 7018
h> 203.252.165.0/24 166.90.213.77 1      0 :0 :13 0 :0 :0 65001 4355 1 7018
h> 192.50.208.0/23 166.90.213.77 1      0 :0 :13 0 :0 :0 65001 4355 1 701
h> 133.33.0.0/16 166.90.213.77 1      0 :0 :13 0 :0 :0 65001 4355 1 701
*> 204.17.220.0/24 166.90.213.77 1      0 :1 :4 0 :0 :0 65001 4355 701 62
```

Syntax: `show ip bgp flap-statistics [regular-expression <regular-expression> | <address> <mask> [longer-prefixes] | neighbor <ip-addr>]`

The **regular-expression** <regular-expression> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters. Refer to [“Using regular expressions to filter”](#) on page 1397.

The <address> <mask> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157. or that have a longer prefix (such as 209.157.22.) are displayed.

The **neighbor** <ip-addr> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbors <ip-addr> flap-statistics**.

[Table 241](#) shows the field definitions for the display output.

TABLE 241 Route flap dampening statistics

Field	Description
Total number of flapping routes	Total number of routes in the Layer 3 switch BGP4 route table that have changed state and thus have been marked as flapping routes.
Status code	Indicates the dampening status of the route, which can be one of the following: <ul style="list-style-type: none">• > – This is the best route among those in the BGP4 route table to the route destination.• d – This route is currently dampened, and thus unusable.• h – The route has a history of flapping and is unreachable now.• * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to the Layer 3 switch.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and thus be usable again.
Path	Shows the AS-path information for the route.

You also can display all the dampened routes by entering the **show ip bgp dampened-paths** command.

Clearing route flap dampening statistics

To clear route flap dampening statistics, use the following CLI method.

NOTE

Clearing the dampening statistics for a route does not change the dampening status of the route.

To clear all the route dampening statistics, enter the following command at any level of the CLI.

```
Brocade#clear ip bgp flap-statistics
```

Syntax: `clear ip bgp flap-statistics` [**regular-expression** <regular-expression> | <address> <mask> | **neighbor** <ip-addr>]

The parameters are the same as those for the **show ip bgp flap-statistics** command (except the **longer-prefixes** option is not supported). Refer to [“Displaying route flap dampening statistics”](#) on page 1419.

NOTE

The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. Refer to [“Displaying route flap dampening statistics”](#) on page 1419.

Generating traps for BGP

You can enable and disable SNMP traps for BGP. BGP traps are enabled by default.

To enable BGP traps after they have been disabled, enter the following command.

```
Brocade(config)#snmp-server enable traps bgp
```

Syntax: [**no**] **snmp-server enable traps bgp**

Use the **no** form of the command to disable BGP traps.

Displaying BGP4 information

You can display the following configuration information and statistics for the BGP4 protocol on the router:

- Summary BGP4 configuration information for the router
- Active BGP4 configuration information (the BGP4 information in the running-config)
- CPU utilization statistics
- Neighbor information
- Peer-group information
- Information about the paths from which BGP4 selects routes
- Summary BGP4 route information
- The router BGP4 route table
- Route flap dampening statistics
- Active route maps (the route map configuration information in the running-config)
- BGP4 graceful restart neighbor information

Displaying summary BGP4 information

You can display the local AS number, the maximum number of routes and neighbors supported, and some BGP4 statistics.

To view summary BGP4 information for the router, enter the **show ip bgp summary** command at any CLI prompt.

```
Brocade#show ip bgp summary
BGP4 Summary
Router ID: 101.0.0.1   Local AS Number : 4
Confederation Identifier : not configured
Confederation Peers: 4 5
Maximum Number of Paths Supported for Load Sharing : 1
Number of Neighbors Configured : 11
Number of Routes Installed : 2
Number of Routes Advertising to All Neighbors : 8
Number of Attribute Entries Installed : 6
Neighbor Address  AS#  State  Time      Rt:Accepted  Filtered  Sent  ToSend
1.2.3.4           200  ADMDN  0h44m56s  0            0         0     2
10.0.0.2          5    ADMDN  0h44m56s  0            0         0     0
10.1.0.2          5    ESTAB  0h44m56s  1            11        0     0
10.2.0.2          5    ESTAB  0h44m55s  1            0         0     0
10.3.0.2          5    ADMDN  0h25m28s  0            0         0     0
10.4.0.2          5    ADMDN  0h25m31s  0            0         0     0
10.5.0.2          5    CONN   0h 0m 8s  0            0         0     0
10.7.0.2          5    ADMDN  0h44m56s  0            0         0     0
100.0.0.1         4    ADMDN  0h44m56s  0            0         0     2
102.0.0.1         4    ADMDN  0h44m56s  0            0         0     2
150.150.150.150  0    ADMDN  0h44m56s  0            0         0     2
```

Table 242 lists the field definitions for the command output.

TABLE 242 BGP4 summary information

Field	Description
Router ID	The Layer 3 switch router ID.
Local AS Number	The BGP4 AS number the router is in.
Confederation Identifier	The AS number of the confederation the Layer 3 switch is in.
Confederation Peers	The numbers of the local autonomous systems contained in the confederation. This list matches the confederation peer list you configure on the Layer 3 switch.
Maximum Number of Paths Supported for Load Sharing	The maximum number of route paths across which the device can balance traffic to the same destination. The feature is enabled by default but the default number of paths is 1. You can increase the number from 2 through 4 paths. Refer to “Changing the maximum number of paths for BGP4 load sharing” on page 1366.
Number of Neighbors Configured	The number of BGP4 neighbors configured on this Layer 3 switch.
Number of Routes Installed	The number of BGP4 routes in the router BGP4 route table. To display the BGP4 route table, refer to “Displaying the BGP4 route table” on page 1441.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors.

TABLE 242 BGP4 summary information (Continued)

Field	Description
Number of Attribute Entries Installed	The number of BGP4 route-attribute entries in the router route-attributes table. To display the route-attribute table, refer to “Displaying BGP4 route-attribute entries” on page 1447.
Neighbor Address	The IP addresses of this router BGP4 neighbors.
AS#	The AS number.

TABLE 242 BGP4 summary information (Continued)

Field	Description
State	<p>The state of this router neighbor session with each neighbor. The states are from this router perspective of the session, not the neighbor perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each router:</p> <ul style="list-style-type: none"> • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND – The neighbor has been administratively shut down. Refer to “Administratively shutting down a session with a BGP4 neighbor” on page 1364. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. <p>NOTE: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.</p> <ul style="list-style-type: none"> • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE packets with the neighbor. If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. <p>NOTE: If you display information for the neighbor using the show ip bgp neighbors <ip-addr> command, the TCP receiver queue value will be greater than 0.</p> <p>Operational States: Additional information regarding the BGP operational states described above may be added as follows:</p> <ul style="list-style-type: none"> • (+) – is displayed if there is more BGP4 data in the TCP receiver queue. Note: If you display information for the neighbor using the show ip bgp neighbors <ip-addr> command, the TCP receiver queue value will be greater than 0. • (-) – indicates that the session has gone down and the software is clearing or removing routes. • (*) – indicates that the inbound or outbound policy is being updated for the peer. • (s) – indicates that the peer has negotiated restart, and the session is in a stale state. • (r) – indicates that the peer is restarting the BGP4 connection, through restart. • (^) – on the standby MP indicates that the peer is in the ESTABLISHED state and has received restart capability (in the primary MP). • (<) – indicates that the device is waiting to receive the “End of RIB” message the peer.
Time	The time that has passed since the state last changed.
Accepted	The number of routes received from the neighbor that this router installed in the BGP4 route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this router filtered out some of the routes received in the UPDATE messages.

TABLE 242 BGP4 summary information (Continued)

Field	Description
Filtered	The routes or prefixes that have been filtered out: <ul style="list-style-type: none"> • If soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4 route table) but retained in memory. • If soft reconfiguration is not enabled, this field shows the number of BGP4 routes that have been filtered out.
Sent	The number of BGP4 routes that the Layer 3 switch has sent to the neighbor.
ToSend	The number of routes the Layer 3 switch has queued to send to this neighbor.

Displaying the active BGP4 configuration

To view the active BGP4 configuration information contained in the running-config without displaying the entire running-config, use the following CLI method.

To display the device active BGP4 configuration, enter the **show ip bgp config** command at any level of the CLI.

```
Brocade#show ip bgp config
Current BGP configuration:
router bgp
address-filter 1 deny any any
as-path-filter 1 permit ^65001$
local-as 65002
maximum-paths 4
neighbor pgl peer-group
neighbor pgl remote-as 65001
neighbor pgl description "Brocade group 1"
neighbor pgl distribute-list out 1
neighbor 192.169.100.1 peer-group pgl
neighbor 192.169.101.1 peer-group pgl
neighbor 192.169.102.1 peer-group pgl
neighbor 192.169.201.1 remote-as 65101
neighbor 192.169.201.1 shutdown
neighbor 192.169.220.3 remote-as 65432
network 1.1.1.0 255.255.255.0
network 2.2.2.0 255.255.255.0
redistribute connected
```

Syntax: show ip bgp config

Displaying CPU utilization statistics

You can display CPU utilization statistics for BGP4 and other IP protocols.

To display CPU utilization statistics for BGP4 for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the **show process cpu** command at any level of the CLI.

Displaying BGP4 information

```
Brocade#show process cpu
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ARP              0.01        0.03        0.09        0.22         9
BGP            0.04       0.06       0.08       0.14       13
GVRP             0.00        0.00        0.00        0.00         0
ICMP             0.00        0.00        0.00        0.00         0
IP               0.00        0.00        0.00        0.00         0
OSPF             0.00        0.00        0.00        0.00         0
RIP              0.00        0.00        0.00        0.00         0
STP              0.00        0.00        0.00        0.00         0
VRRP             0.00        0.00        0.00        0.00         0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example.

```
Brocade#show process cpu
The system has only been up for 6 seconds.
Process Name    5Sec(%)    1Min(%)    5Min(%)    15Min(%)    Runtime(ms)
ARP              0.01        0.00        0.00        0.00         0
BGP              0.00        0.00        0.00        0.00         0
GVRP             0.00        0.00        0.00        0.00         0
ICMP             0.01        0.00        0.00        0.00         1
IP               0.00        0.00        0.00        0.00         0
OSPF             0.00        0.00        0.00        0.00         0
RIP              0.00        0.00        0.00        0.00         0
STP              0.00        0.00        0.00        0.00         0
VRRP             0.00        0.00        0.00        0.00         0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following.

```
Brocade#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name    Sec(%)    Time(ms)
ARP              0.00        0
BGP              0.00        0
GVRP             0.00        0
ICMP             0.01        1
IP               0.00        0
OSPF             0.00        0
RIP              0.00        0
STP              0.01        0
VRRP             0.00        0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: `show process cpu [<num>]`

The `<num>` parameter specifies the number of seconds and can be from 1 through 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying summary neighbor information

To display summary neighbor information, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 192.168.4.211 routes-summary
1   IP Address: 192.168.4.211
Routes Accepted/Installed:1,  Filtered/Kept:11,  Filtered:11
    Routes Selected as BEST Routes:1
        BEST Routes not Installed in IP Forwarding Table:0
    Unreachable Routes (no IGP Route for NEXTHop):0
    History Routes:0

NLRI's Received in Update Message:24,  Withdraws:0 (0),  Replacements:1
    NLRI's Discarded due to
        Maximum Prefix Limit:0,  AS Loop:0
        Invalid Nexthop:0,  Invalid Nexthop Address:0.0.0.0
        Duplicated Originator_ID:0,  Cluster_ID:0

Routes Advertised:0,  To be Sent:0,  To be Withdrawn:0
NLRI's Sent in Update Message:0,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
    Receiving Update Messages:0,  Accepting Routes(NLRI):0
    Attributes:0,  Outbound Routes(RIB-out):0
```

Syntax: `show ip bgp neighbors [<ip-addr>] | [routes-summary]`

[Table 243](#) lists the field definitions for the command output.

TABLE 243 BGP4 route summary information for a neighbor

Field	Description
IP Address	The IP address of the neighbor
Routes Received	How many routes the Layer 3 switch has received from the neighbor during the current BGP4 session: <ul style="list-style-type: none"> Accepted/Installed – Indicates how many of the received routes the Layer 3 switch accepted and installed in the BGP4 route table. Filtered/Kept – Indicates how many routes were filtered out, but were nonetheless retained in memory for use by the soft reconfiguration feature. Filtered – Indicates how many of the received routes were filtered out.
Routes Selected as BEST Routes	The number of routes that the Layer 3 switch selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the Layer 3 switch does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.

TABLE 243 BGP4 route summary information for a neighbor (Continued)

Field	Description
NLRIs Received in Update Message	<p>The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages:</p> <ul style="list-style-type: none"> Withdraws – The number of withdrawn routes the Layer 3 switch has received. Replacements – The number of replacement routes the Layer 3 switch has received.
NLRIs Discarded due to	<p>Indicates the number of times the Layer 3 switch discarded an NLRI for the neighbor due to the following reasons:</p> <ul style="list-style-type: none"> Maximum Prefix Limit – The Layer 3 switch configured maximum prefix amount had been reached. AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. Invalid Nexthop – The next hop value was not acceptable. Duplicated Originator_ID – The originator ID was the same as the local router ID. Cluster_ID – The cluster list contained the local cluster ID, or contained the local router ID (see above) if the cluster ID is not configured.
Routes Advertised	<p>The number of routes the Layer 3 switch has advertised to this neighbor:</p> <ul style="list-style-type: none"> To be Sent – The number of routes the Layer 3 switch has queued to send to this neighbor. To be Withdrawn – The number of NLRIs for withdrawing routes the Layer 3 switch has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	<p>The number of NLRIs for new routes the Layer 3 switch has sent to this neighbor in UPDATE messages:</p> <ul style="list-style-type: none"> Withdraws – The number of routes the Layer 3 switch has sent to the neighbor to withdraw. Replacements – The number of routes the Layer 3 switch has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	<p>Statistics for the times the Layer 3 switch has run out of BGP4 memory for the neighbor during the current BGP4 session:</p> <ul style="list-style-type: none"> Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. Accepting Routes(NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. Attributes – The number of times there was no memory for BGP4 attribute entries. Outbound Routes(RIB-out) – The number of times there was no memory to place a “best” route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying BGP4 neighbor information

To view BGP4 neighbor information including the values for all the configured parameters, enter the following command.

NOTE

The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

```

Brocade#show ip bgp neighbors 10.4.0.2
1  IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 100.0.0.1
    Description: neighbor 10.4.0.2
    State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
    PeerGroup: pgl
    Multihop-EBGP: yes, ttl: 1
    RouteReflectorClient: yes
    SendCommunity: yes
    NextHopSelf: yes
    DefaultOriginate: yes (default sent)
    MaximumPrefixLimit: 90000
    RemovePrivateAs: : yes
    RefreshCapability: Received
Route Filter Policies:
    Distribute-list: (out) 20
    Filter-list: (in) 30
    Prefix-list: (in) pfl
    Route-map: (in) setnp1 (out) setnp2
Messages:      Open      Update  KeepAlive  Notification  Refresh-Req
Sent          : 1        1        1           0             0
Received: 1      8        1           0             0
Last Update Time: NLRI      Withdraw      NLRI      Withdraw
                  Tx: 0h0m59s  ---          Rx: 0h0m59s  ---
Last Connection Reset Reason:Unknown
Notification Sent:      Unspecified
Notification Received: Unspecified
TCP Connection state: ESTABLISHED
Local host:  10.4.0.1, Local  Port: 179
Remote host: 10.4.0.2, Remote Port: 8053
ISentSeq:    52837276  SendNext:    52837392  TotUnAck:      0
TotSent:      116  ReTrans:      0  UnAckSeq:    52837392
IRcvSeq:    2155052043  RcvNext:    2155052536  SendWnd:      16384
TotalRcv:      493  DupliRcv:      0  RcvWnd:      16384
SendQue:      0  RcvQue:      0  CngstWnd:    1460

```

This example shows how to display information for a specific neighbor, by specifying the neighbor IP address with the command. None of the other display options are used; thus, all of the information is displayed for the neighbor. The number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the Layer 3 switch Transmission Control Block (TCB) for the TCP session between the Layer 3 switch and its neighbor. These fields are described in detail in section 3.2 of RFC 793, “Transmission Control Protocol Functional Specification”.

Syntax: `show ip bgp neighbors [<ip-addr> [advertised-routes [detail [<ip-addr>[/<mask-bits>]]]] | [attribute-entries [detail]] | [flap-statistics] | [last-packet-with-error] | [received prefix-filter] | [received-routes] | [routes [best]] | [detail [best]] | [not-installed-best] | [unreachable]] | [rib-out-routes [<ip-addr>/<mask-bits> | <ip-addr> <net-mask> | detail]] | [routes-summary]]`

The `<ip-addr>` option lets you narrow the scope of the command to a specific neighbor.

The **advertised-routes** option displays only the routes that the Layer 3 switch has advertised to the neighbor during the current BGP4 neighbor session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet's contents are displayed in decoded (human-readable) format.

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **received-routes** option lists all the route information received in route updates from the neighbor since the soft reconfiguration feature was enabled. Refer to [“Using soft reconfiguration”](#) on page 1452.

The **routes** option lists the routes received in UPDATE messages from the neighbor. You can specify the following additional options:

- **best** – Displays the routes received from the neighbor that the Layer 3 switch selected as the best routes to their destinations.
- **not-installed-best** – Displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
- **unreachable** – Displays the routes that are unreachable because the Layer 3 switch does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** – Displays detailed information for the specified routes. You can refine your information request by also specifying one of the options above (**best**, **not-installed-best**, or **unreachable**).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. You can display all the routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this Layer 3 switch from the neighbor
- Number of routes this Layer 3 switch filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor.

[Table 244](#) lists the field definitions for the command output.

TABLE 244 BGP4 neighbor information

Field	Description
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.
EBGP/IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: <ul style="list-style-type: none"> • EBGP – The neighbor is in another AS. • EBGP_Confed – The neighbor is a member of another sub-AS in the same confederation. • IBGP – The neighbor is in the same AS.

TABLE 244 BGP4 neighbor information (Continued)

Field	Description
RouterID	The neighbor router ID.
Description	The description you gave the neighbor when you configured it on the Layer 3 switch.
State	<p>The state of the router session with the neighbor. The states are from this router perspective of the session, not the neighbor perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each router:</p> <ul style="list-style-type: none"> • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND – The neighbor has been administratively shut down. Refer to “Administratively shutting down a session with a BGP4 neighbor” on page 1364. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. <p>NOTE: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.</p> <ul style="list-style-type: none"> • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE messages with the neighbor. If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. <p>NOTE: If you display information for the neighbor using the show ip bgp neighbors <ip-addr> command, the TCP receiver queue value will be greater than 0.</p>
Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this router sends keep alive messages to the neighbor. Refer to “Changing the Keep Alive Time and Hold Time” on page 1365.
HoldTime	The hold time, which specifies how many seconds the router will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead. Refer to “Changing the Keep Alive Time and Hold Time” on page 1365.
PeerGroup	The name of the peer group the neighbor is in, if applicable.
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.
MaximumPrefixLimit	Lists the maximum number of prefixes the Layer 3 switch will accept from this neighbor.

TABLE 244 BGP4 neighbor information (Continued)

Field	Description
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this Layer 3 switch has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, if configured.
Filter-list	Lists the filter list parameters, if configured.
Prefix-list	Lists the prefix list parameters, if configured.
Route-map	Lists the route map parameters, if configured.
Messages Sent	The number of messages this router has sent to the neighbor. The display shows statistics for the following message types: <ul style="list-style-type: none"> • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this router has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: <ul style="list-style-type: none"> • NLRIs • Withdraws

TABLE 244 BGP4 neighbor information (Continued)

Field	Description
Last Connection Reset Reason	<p>The reason the previous session with this neighbor ended. The reason can be one of the following.</p> <p>Reasons described in the BGP specifications:</p> <ul style="list-style-type: none"> • Message Header Error • Connection Not Synchronized • Bad Message Length • Bad Message Type • OPEN Message Error • Unsupported Version Number • Bad Peer AS Number • Bad BGP Identifier • Unsupported Optional Parameter • Authentication Failure • Unacceptable Hold Time • Unsupported Capability • UPDATE Message Error • Malformed Attribute List • Unrecognized Well-known Attribute • Missing Well-known Attribute • Attribute Flags Error • Attribute Length Error • Invalid ORIGIN Attribute • Invalid NEXT_HOP Attribute • Optional Attribute Error • Invalid Network Field • Malformed AS_PATH • Hold Timer Expired • Finite State Machine Error • Rcv Notification
Last Connection Reset Reason (cont.)	<p>Reasons specific to the Brocade implementation:</p> <ul style="list-style-type: none"> • Reset All Peer Sessions • User Reset Peer Session • Port State Down • Peer Removed • Peer Shutdown • Peer AS Number Change • Peer AS Confederation Change • TCP Connection KeepAlive Timeout • TCP Connection Closed by Remote • TCP Data Stream Error Detected

TABLE 244 BGP4 neighbor information (Continued)

Field	Description
Notification Sent	<p>If the router receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.</p> <p>Message Header Error:</p> <ul style="list-style-type: none"> • Connection Not Synchronized • Bad Message Length • Bad Message Type • Unspecified <p>Open Message Error:</p> <ul style="list-style-type: none"> • Unsupported Version • Bad Peer As • Bad BGP Identifier • Unsupported Optional Parameter • Authentication Failure • Unacceptable Hold Time • Unspecified <p>Update Message Error:</p> <ul style="list-style-type: none"> • Malformed Attribute List • Unrecognized Attribute • Missing Attribute • Attribute Flag Error • Attribute Length Error • Invalid Origin Attribute • Invalid NextHop Attribute • Optional Attribute Error • Invalid Network Field • Malformed AS Path • Unspecified <p>Hold Timer Expired</p> <p>Finite State Machine Error</p> <p>Cease</p> <p>Unspecified</p>
Notification Received	See above.

TABLE 244 BGP4 neighbor information (Continued)

Field	Description
TCP Connection state	<p>The state of the connection with the neighbor. The connection can have one of the following states:</p> <ul style="list-style-type: none"> • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the Layer 3 switch.
Local port	The TCP port the Layer 3 switch is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the Layer 3 switch.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the Layer 3 switch that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the Layer 3 switch retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.

TABLE 244 BGP4 neighbor information (Continued)

Field	Description
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying route information for a neighbor

You can display routes based on the following criteria:

- A summary of the routes for a specific neighbor.
- The routes received from the neighbor that the Layer 3 switch selected as the best routes to their destinations.
- The routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
- The routes that are unreachable because the Layer 3 switch does not have a valid RIP, OSPF, or static route to the next hop.
- Routes for a specific network advertised by the Layer 3 switch to the neighbor.
- The Routing Information Base (RIB) for a specific network advertised to the neighbor. You can display the RIB regardless of whether the Layer 3 switch has already sent it to the neighbor.

To display route information for a neighbor, use the following CLI methods.

Displaying summary route information

To display summary route information, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 10.1.0.2 routes-summary
1  IP Address: 10.1.0.2
Routes Accepted/Installed:1,  Filtered/Kept:11,  Filtered:11
  Routes Selected as BEST Routes:1
    BEST Routes not Installed in IP Forwarding Table:0
  Unreachable Routes (no IGP Route for NEXTHOP):0
  History Routes:0

NLRI's Received in Update Message:24,  Withdraws:0 (0),  Replacements:1
  NLRI's Discarded due to
    Maximum Prefix Limit:0,  AS Loop:0
    Invalid Nexthop:0,  Invalid Nexthop Address:0.0.0.0
    Duplicated Originator_ID:0,  Cluster_ID:0

Routes Advertised:0,  To be Sent:0,  To be Withdrawn:0
NLRI's Sent in Update Message:0,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
  Receiving Update Messages:0,  Accepting Routes(NLRI):0
  Attributes:0,  Outbound Routes(RIB-out):0
```

Table 245 lists the field definitions for the command output.

TABLE 245 BGP4 route summary information for a neighbor

Field	Description
Routes Received	How many routes the Layer 3 switch has received from the neighbor during the current BGP4 session: <ul style="list-style-type: none"> • Accepted/Installed – Indicates how many of the received routes the Layer 3 switch accepted and installed in the BGP4 route table. • Filtered – Indicates how many of the received routes the Layer 3 switch did not accept or install because they were denied by filters on the Layer 3 switch.
Routes Selected as BEST Routes	The number of routes that the Layer 3 switch selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the Layer 3 switch does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: <ul style="list-style-type: none"> • Withdraws – The number of withdrawn routes the Layer 3 switch has received. • Replacements – The number of replacement routes the Layer 3 switch has received.
NLRIs Discarded due to	Indicates the number of times the Layer 3 switch discarded an NLRI for the neighbor due to the following reasons: <ul style="list-style-type: none"> • Maximum Prefix Limit – The Layer 3 switch configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • Invalid Nexthop – The next hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local router ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local router ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the Layer 3 switch has advertised to this neighbor: <ul style="list-style-type: none"> • To be Sent – The number of routes the Layer 3 switch has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the Layer 3 switch has queued up to send to this neighbor in UPDATE messages.

TABLE 245 BGP4 route summary information for a neighbor (Continued)

Field	Description
NLRIs Sent in Update Message	<p>The number of NLRIs for new routes the Layer 3 switch has sent to this neighbor in UPDATE messages:</p> <ul style="list-style-type: none"> • Withdraws – The number of routes the Layer 3 switch has sent to the neighbor to withdraw. • Replacements – The number of routes the Layer 3 switch has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	<p>Statistics for the times the Layer 3 switch has run out of BGP4 memory for the neighbor during the current BGP4 session:</p> <ul style="list-style-type: none"> • Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes(NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4 attribute entries. • Outbound Routes(RIB-out) – The number of times there was no memory to place a “best” route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying advertised routes

To display the routes the Layer 3 switch has advertised to a specific neighbor for a specific network, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 192.168.4.211 advertised-routes
      There are 2 routes advertised to neighbor 192.168.4.211
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
      Network      Next Hop      Metric      LocPrf      Weight      Status
1       102.0.0.0/24  192.168.2.102    12                32768      BL
2       200.1.1.0/24  192.168.2.102     0                32768      BL
```

You also can enter a specific route, as in the following example.

```
Brocade#show ip bgp neighbors 192.168.4.211 advertised 200.1.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
      Network      Next Hop      Metric      LocPrf      Weight      Status
1       200.1.1.0/24  192.168.2.102     0                32768      BL
```

Syntax: `show ip bgp neighbors <ip-addr> advertised-routes [<ip-addr>/<prefix>]`

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the `show ip bgp` display.

Displaying the best routes

To display the routes received from a specific neighbor that are the “best” routes to their destinations, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 192.168.4.211 routes best
```

Syntax: `show ip bgp neighbors <ip-addr> routes best`

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the `show ip bgp` display.

Displaying the best routes that were nonetheless not installed in the IP route table

To display the BGP4 routes received from a specific neighbor that are the “best” routes to their destinations but are not installed in the Layer 3 switch IP route table, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 192.168.4.211 routes not-installed-best
```

Each of the displayed routes is a valid path to its destination, but the Layer 3 switch received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The Layer 3 switch always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: `show ip bgp neighbors <ip-addr> routes not-installed-best`

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the `show ip bgp` display.

Displaying the routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 192.168.4.211 routes unreachable
```

Syntax: `show ip bgp neighbors <ip-addr> routes unreachable`

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the `show ip bgp` display.

Displaying the Adj-RIB-Out for a neighbor

To display the Layer 3 switch current BGP4 Routing Information Base (Adj-RIB-Out) for a specific neighbor and a specific destination network, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 192.168.4.211 rib-out-routes 192.168.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
      Prefix      Next Hop      Metric      LocPrf      Weight Status
1      200.1.1.0/24      0.0.0.0          0          101      32768 BL
```

The Adj-RIB-Out contains the routes that the Layer 3 switch either has most recently sent to the neighbor or is about to send to the neighbor.

Syntax: `show ip bgp neighbors <ip-addr> rib-out-routes [<ip-addr>/<prefix>]`

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the `show ip bgp` display.

Displaying peer group information

You can display configuration information for peer groups.

To display peer-group information, enter a command such as the following at the Privileged EXEC level of the CLI.

```
Brocade#show ip bgp peer-group pg1
1  BGP peer-group is pg
   Description: peer group abc
       SendCommunity: yes
       NextHopSelf: yes
       DefaultOriginate: yes
   Members:
       IP Address: 192.168.10.10, AS: 65111
```

Syntax: `show ip bgp peer-group [<peer-group-name>]`

Only the parameters that have values different from their defaults are listed.

Displaying summary route information

To display summary statistics for all the routes in the Layer 3 switch BGP4 route table, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp routes summary
Total number of BGP routes (NLRIs) Installed      : 20
Distinct BGP destination networks                 : 20
Filtered BGP routes for soft reconfig              : 100178
Routes originated by this router                   : 2
Routes selected as BEST routes                    : 19
BEST routes not installed in IP forwarding table   : 1
Unreachable routes (no IGP route for NEXTHOP)     : 1
IBGP routes selected as best routes                : 0
EBGP routes selected as best routes                : 17
```

Syntax: `show ip bgp routes summary`

[Table 246](#) lists the field definitions for the command output.

TABLE 246 BGP4 summary route information

Field	Description
Total number of BGP routes (NLRIs) Installed	The number of BGP4 routes the Layer 3 switch has installed in the BGP4 route table.
Distinct BGP destination networks	The number of destination networks the installed routes represent. The BGP4 route table can have multiple routes to the same network.
Filtered BGP routes for soft reconfig	The number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained. For information about soft reconfiguration, refer to “Using soft reconfiguration” on page 1452.
Routes originated by this router	The number of routes in the BGP4 route table that this Layer 3 switch originated.
Routes selected as BEST routes	The number of routes in the BGP4 route table that this Layer 3 switch has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	The number of BGP4 routes that are the best BGP4 routes to their destinations but were not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable routes (no IGP route for NEXTHOP)	The number of routes in the BGP4 route table whose destinations are unreachable because the next hop is unreachable.

TABLE 246 BGP4 summary route information (Continued)

Field	Description
IBGP routes selected as best routes	The number of “best” routes in the BGP4 route table that are IBGP routes.
EBGP routes selected as best routes	The number of “best” routes in the BGP4 route table that are EBGP routes.

Displaying the BGP4 route table

BGP4 uses filters you define as well as the algorithm described in [“How BGP4 selects a path for a route”](#) on page 1344 to determine the preferred route to a destination. BGP4 sends only the preferred route to the router IP table. However, if you want to view all the routes BGP4 knows about, you can display the BGP4 table using either of the following methods.

To view the BGP4 route table, enter the following command.

```
Brocade#show ip bgp routes
Total number of BGP Routes: 97371
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
      Prefix          Next Hop      Metric      LocPrf      Weight      Status
1      3.0.0.0/8       192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 701 80
2      4.0.0.0/8       192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 1
3      4.60.212.0/22    192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8       192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 3356 7170 1455
5      8.8.1.0/24      192.168.4.106      0            100         0          BE
      AS_PATH: 65001
```

Syntax: `show ip bgp routes` `[[network] <ip-addr>]` `<num>` `[age <secs>]` `[as-path-access-list <num>]` `[best]` `[cidr-only]` `[community <num>]` `[no-export]` `[no-advertise]` `[internet]` `[local-as]` `[community-access-list <num>]` `[community-list <num>]` `[detail <option>]` `[filter-list <num, num,...>]` `[next-hop <ip-addr>]` `[no-best]` `[not-installed-best]` `[prefix-list <string>]` `[regular-expression <regular-expression>]` `[route-map <map-name>]` `[summary]` `[unreachable]`

The `<ip-addr>` option displays routes for a specific network. The **network** keyword is optional. You can enter the network address without entering “network” in front of it.

The `<num>` option specifies the table entry with which you want the display to start. For example, if you want to list entries beginning with table entry 100, specify 100.

The **age** `<secs>` parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** `<num>` parameter filters the display using the specified AS-path ACL.

The **best** parameter displays the routes received from the neighbor that the Layer 3 switch selected as the best routes to their destinations.

The **cidr-only** option lists only the routes whose network masks do not match their class network length.

The **community** option lets you display routes for a specific community. You can specify **local-as**, **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of 1 through 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** *<num>* parameter filters the display using the specified community ACL.

The **community-list** option lets you display routes that match a specific community filter.

The **detail** option lets you display more details about the routes. You can refine your request by also specifying one of the other display options after the detail keyword.

The **filter-list** option displays routes that match a specific address filter list.

The **next-hop** *<ip-addr>* option displays the routes for a given next-hop IP address.

The **no-best** option displays the routes for which none of the routes to a given prefix were selected as the best route. The IP route table does not contain a BGP4 route for any of the routes listed by the command.

The **not-installed-best** option displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes).

The **prefix-list** *<string>* parameter filters the display using the specified IP prefix list.

The **regular-expression** *<regular-expression>* option filters the display based on a regular expression. Refer to [“Using regular expressions to filter”](#) on page 1397.

The **route-map** *<map-name>* parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map set statements.

The **summary** option displays summary information for the routes.

The **unreachable** option displays the routes that are unreachable because the Layer 3 switch does not have a valid RIP, OSPF, or static route to the next hop.

Displaying the best BGP4 routes

To display all the BGP4 routes in the Layer 3 switch BGP4 route table that are the best routes to their destinations, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp routes best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
      Prefix           Next Hop           Metric      LocPrf      Weight Status
1      3.0.0.0/8        192.168.4.106           100          0          BE
      AS_PATH: 65001 4355 701 80
2      4.0.0.0/8        192.168.4.106           100          0          BE
      AS_PATH: 65001 4355 1
3      4.60.212.0/22     192.168.4.106           100          0          BE
      AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8        192.168.4.106           100          0          BE
      AS_PATH: 65001 4355 3356 7170 1455
5      9.2.0.0/16       192.168.4.106           100          0          BE
      AS_PATH: 65001 4355 701
```


Syntax: show ip bgp routes best

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the **show ip bgp** display.

Displaying the best BGP4 routes that are not in the IP route table

When the Layer 3 switch has multiple routes to a destination from different sources (such as BGP4, OSPF, RIP, or static routes), the Layer 3 switch selects the route with the lowest administrative distance as the best route, and installs that route in the IP route table.

To display the BGP4 routes that are the “best” routes to their destinations but are not installed in the Layer 3 switch IP route table, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp routes not-installed-best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
        E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
        Prefix          Next Hop          Metric      LocPrf      Weight Status
1          192.168.4.0/24  192.168.4.106    0           100         0      bE
        AS_PATH: 65001
```

Each of the displayed routes is a valid path to its destination, but the Layer 3 switch received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The Layer 3 switch always selects the path with the lowest administrative distance to install in the IP route table.

Notice that the route status in this example is the new status, “b”. Refer to [Table 247](#) on page 1444 for a description.

Syntax: show ip bgp routes not-installed-best

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the **show ip bgp** display.

NOTE

To display the routes that the Layer 3 switch has selected as the best routes and installed in the IP route table, display the IP route table using the **show ip route** command.

Displaying BGP4 routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp routes unreachable
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
        H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
        Prefix          Next Hop          Metric      LocPrf      Weight Status
1          8.8.8.0/24  192.168.5.1      0           101         0
        AS_PATH: 65001 4355 1
```

Syntax: show ip bgp routes unreachable

For information about the fields in this display, refer to [Table 247](#) on page 1444. The fields in this display also appear in the **show ip bgp** display.

Displaying information for a specific route

To display BGP4 network information by specifying an IP address within the network, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp 9.3.4.0
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
      Network      Next Hop      Metric LocPrf Weight Path
*>  9.3.4.0/24      192.168.4.106      100    0      65001 4355 1 1221 ?
      Last update to IP routing table: 0h11m38s, 1 path(s) installed:
      Gateway      Port
      192.168.2.1      2/1
      Route is advertised to 1 peers:
      20.20.20.2(65300)
```

Syntax: `show ip bgp [route] <ip-addr>/<prefix> [longer-prefixes] | <ip-addr>`

If you use the **route** option, the display for the information is different, as shown in the following example.

```
Brocade#show ip bgp route 9.3.4.0
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
      Prefix      Next Hop      Metric      LocPrf      Weight Status
1      9.3.4.0/24      192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 1 1221
      Last update to IP routing table: 0h12m1s, 1 path(s) installed:
      Gateway      Port
      192.168.2.1      2/1
      Route is advertised to 1 peers:
      20.20.20.2(65300)
```

These displays show the following information.

TABLE 247 BGP4 network information

Field	Description
Number of BGP Routes matching display condition	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command output. NOTE: This field appears only if you <i>do not</i> enter the route option.
Prefix	The network address and prefix.
Next Hop	The next-hop router for reaching the network from the Layer 3 switch.
Metric	The value of the route MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 through 4294967295.

TABLE 247 BGP4 network information (Continued)

Field	Description
Weight	The value that this router associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight.
Path	The route AS path. NOTE: This field appears only if you <i>do not</i> enter the route option.
Origin code	A character the display uses to indicate the route origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command output. NOTE: This field appears only if you <i>do not</i> enter the route option.
Status	<p>The route status, which can be one or more of the following:</p> <ul style="list-style-type: none"> • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. <p>NOTE: If the “b” is shown in lowercase, the software was not able to install the route in the IP route table.</p> <ul style="list-style-type: none"> • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this Layer 3 switch. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with “B”. <p>NOTE: If the “m” is shown in lowercase, the software was not able to install the route in the IP route table.</p> <ul style="list-style-type: none"> • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. <p>NOTE: This field appears only if you enter the route option.</p>

Displaying route details

Here is an example of the information displayed when you use the **detail** option. In this example, the information for one route is shown.

```

Brocade#show ip bgp routes detail
Total number of BGP Routes: 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
1 Prefix: 10.5.0.0/24, Status: BME, Age: 0h28m28s
NEXT_HOP: 201.1.1.2, Learned from Peer: 10.1.0.2 (5)
LOCAL_PREF: 101, MED: 0, ORIGIN: igp, Weight: 10
AS_PATH: 5
Adj_RIB_out count: 4, Admin distance 20

```

These displays show the following information.

TABLE 248 BGP4 route information

Field	Description
Total number of BGP Routes	The number of BGP4 routes.
Status codes	A list of the characters the display uses to indicate the route status. The status code is appears in the left column of the display, to the left of each route. The status codes are described in the command output.
Prefix	The network prefix and mask length.
Status	<p>The route status, which can be one or more of the following:</p> <ul style="list-style-type: none"> • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. <p>NOTE: If the “b” is shown in lowercase, the software was not able to install the route in the IP route table.</p> <ul style="list-style-type: none"> • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 switch received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this Layer 3 switch. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with “B”. <p>NOTE: If the “m” is shown in lowercase, the software was not able to install the route in the IP route table.</p> <ul style="list-style-type: none"> • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
Age	The last time an update occurred.
Next_Hop	The next-hop router for reaching the network from the Layer 3 switch.
Learned from Peer	The IP address of the neighbor that sent this route.
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 through 4294967295.
MED	The route metric. If the route does not have a metric, this field is blank.
Origin	<p>The source of the route information. The origin can be one of the following:</p> <ul style="list-style-type: none"> • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. <p>When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.</p>

TABLE 248 BGP4 route information (Continued)

Field	Description
Weight	The value that this router associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight.
Atomic	Whether network information in this route has been aggregated <i>and</i> this aggregation has resulted in information loss. NOTE: Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The router that originated this aggregator.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the Layer 3 switch learned the route.
Admin Distance	The administrative distance of the route.
Adj_RIB_out	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.

Displaying BGP4 route-attribute entries

The route-attribute entries table lists the sets of BGP4 attributes stored in the router memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the router typically has fewer route attribute entries than routes. To display the route-attribute entries table, use one of the following methods.

To display the IP route table, enter the following command.

```
Brocade#show ip bgp attribute-entries
```

Syntax: show ip bgp attribute-entries

Here is an example of the information displayed by this command. A zero value indicates that the attribute is not set.

```
Brocade#show ip bgp attribute-entries
      Total number of BGP Attribute Entries: 7753
1      Next Hop :192.168.11.1      Metric :0      Origin:IGP
      Originator:0.0.0.0      Cluster List:None
      Aggregator:AS Number :0      Router-ID:0.0.0.0      Atomic:FALSE
      Local Pref:100      Communities:Internet
      AS Path : (65002) 65001 4355 2548 3561 5400 6669 5548
2      Next Hop :192.168.11.1      Metric :0      Origin:IGP
      Originator:0.0.0.0      Cluster List:None
      Aggregator:AS Number :0      Router-ID:0.0.0.0      Atomic:FALSE
      Local Pref:100      Communities:Internet
      AS Path : (65002) 65001 4355 2548
```

Table 249 lists the field definitions for the command output.

TABLE 249 BGP4 route-attribute entries information

Field	Description
Total number of BGP Attribute Entries	The number of routes contained in this router BGP4 route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: <ul style="list-style-type: none"> EGP – The routes with this set of attributes came to BGP through EGP. IGP – The routes with this set of attributes came to BGP through IGP. INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: <ul style="list-style-type: none"> AS Number shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. Router-ID shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: <ul style="list-style-type: none"> TRUE – Indicates information loss has occurred FALSE – Indicates no information loss has occurred NOTE: Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The autonomous systems through which routes with this set of attributes have passed. The local AS is shown in parentheses.

Displaying the routes BGP4 has placed in the IP route table

The IP route table indicates the routes it has received from BGP4 by listing “BGP” as the route type.

To display the IP route table, enter the following command.

```
Brocade#show ip route
```

Syntax: `show ip route [<ip-addr> | <num> | bgp | ospf | rip]`

Here is an example of the information displayed by this command. Notice that most of the routes in this example have type “B”, indicating that their source is BGP4.

```

Brocade#show ip route
Total number of IP routes: 50834
B:BGP D:Directly-Connected O:OSPF R:RIP S:Static
  Network Address  NetMask          Gateway          Port      Cost    Type
  3.0.0.0          255.0.0.0        192.168.13.2    1/1       0       B
  4.0.0.0          255.0.0.0        192.168.13.2    1/1       0       B
  9.20.0.0         255.255.128.0    192.168.13.2    1/1       0       B
  10.1.0.0         255.255.0.0      0.0.0.0         1/1       1       D
  10.10.11.0       255.255.255.0    0.0.0.0         2/24      1       D
  12.2.97.0        255.255.255.0    192.168.13.2    1/1       0       B
  12.3.63.0        255.255.255.0    192.168.13.2    1/1       0       B
  12.3.123.0       255.255.255.0    192.168.13.2    1/1       0       B
  12.5.252.0       255.255.254.0    192.168.13.2    1/1       0       B
  12.6.42.0        255.255.254.0    192.168.13.2    1/1       0       B
remaining 50824 entries not shown...

```

Displaying route flap dampening statistics

To display route dampening statistics or all the dampened routes, enter the following command at any level of the CLI.

```

Brocade#show ip bgp flap-statistics
Total number of flapping routes: 414
  Status Code  >:best d:damped h:history *:valid
  Network      From          Flaps Since      Reuse      Path
h> 192.50.206.0/23 166.90.213.77 1 0 :0 :13 0 :0 :0 65001 4355 1 701
h> 203.255.192.0/20 166.90.213.77 1 0 :0 :13 0 :0 :0 65001 4355 1 7018
h> 203.252.165.0/24 166.90.213.77 1 0 :0 :13 0 :0 :0 65001 4355 1 7018
h> 192.50.208.0/23 166.90.213.77 1 0 :0 :13 0 :0 :0 65001 4355 1 701
h> 133.33.0.0/16 166.90.213.77 1 0 :0 :13 0 :0 :0 65001 4355 1 701
*> 204.17.220.0/24 166.90.213.77 1 0 :1 :4 0 :0 :0 65001 4355 701 62

```

Syntax: `show ip bgp flap-statistics [regular-expression <regular-expression> | <address> <mask> [longer-prefixes] | neighbor <ip-addr> | filter-list <num>...]`

The **regular-expression** <regular-expression> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters. Refer to [“Using regular expressions to filter”](#) on page 1397.

The <address> <mask> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

The **neighbor** <ip-addr> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbors <ip-addr> flap-statistics**.

The **filter-list** <num> parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filters are displayed.

[Table 250](#) lists the field definitions for the command output.

TABLE 250 Route flap dampening statistics

Field	Description
Total number of flapping routes	The total number of routes in the Layer 3 switch BGP4 route table that have changed state and thus have been marked as flapping routes.
Status code	Indicates the dampening status of the route, which can be one of the following: <ul style="list-style-type: none"> • > – This is the best route among those in the BGP4 route table to the route destination. • d – This route is currently dampened, and thus unusable. • h – The route has a history of flapping and is unreachable now. • * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to the Layer 3 switch.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and thus be usable again.
Path	Shows the AS-path information for the route.

You also can display all the dampened routes by entering the following command.

show ip bgp dampened-paths.

Displaying the active route map configuration

To view the device active route map configuration (contained in the running-config) without displaying the entire running-config, enter the following command at any level of the CLI.

```
Brocade#show route-map
route-map permitnet4 permit 10
  match ip address prefix-list plist1
route-map permitnet1 permit 1
  match ip address prefix-list plist2
route-map setcomm permit 1
  set community 1234:2345 no-export
route-map test111 permit 111
  match address-filters 11
  set community 11:12 no-export
route-map permit1122 permit 12
  match ip address 11
route-map permit1122 permit 13
  match ip address std_22
```

This example shows that the running-config contains six route maps. Notice that the match and set statements within each route map are listed beneath the command for the route map itself. In this simplified example, each route map contains only one match or set statement.

To display the active configuration for a specific route map, enter a command such as the following, which specifies a route map name.


```
Brocade#show route-map setcomm
route-map setcomm permit 1
set community 1234:2345 no-export
```

This example shows the active configuration for a route map called “setcomm”.

Syntax: `show route-map [<map-name>]`

Displaying BGP4 graceful restart neighbor information

Use the **show ip bgp neighbors** command to display BGP4 restart information for BGP4 neighbors.

```
Brocade# show ip bgp neighbors
Total number of BGP Neighbors: 6
1  IP Address: 50.50.50.10, AS: 20 (EBGP), RouterID: 10.10.10.20
State: ESTABLISHED, Time: 0h0m18s, KeepAliveTime: 60, HoldTime: 180
KeepAliveTimer Expire in 34 seconds, HoldTimer Expire in 163 seconds
Minimum Route Advertisement Interval: 0 seconds
RefreshCapability: Received
GracefulRestartCapability: Received
  Restart Time 120 sec, Restart bit 0
  afi/safi 1/1, Forwarding bit 0
GracefulRestartCapability: Sent
  Restart Time 120 sec, Restart bit 0
  afi/safi 1/1, Forwarding bit 1
Messages:      Open      Update  KeepAlive Notification Refresh-Req
```

The text in bold is the BGP4 restart information for the specified neighbor.

Syntax: `show ip bgp neighbors`

Updating route information and resetting a neighbor session

The following sections describe ways to update route information with a neighbor, reset the session with a neighbor, and close a session with a neighbor.

Whenever you change a policy (ACL, route map, and so on) that affects the routes that the Layer 3 switch learns from a BGP4 neighbor or peer group of neighbors, you must enter a command to place the changes into effect. The changes take place automatically, but only affect new route updates. To make changes retroactive for routes received or sent before the changes were made, you need to enter a clear command.

You can update the learned routes using either of the following methods:

- Request the complete BGP4 route table from the neighbor or peer group. You can use this method if the neighbor supports the refresh capability (RFCs 2842 and 2858).
- Clear (reset) the session with the neighbor or peer group. This is the only method you can use if the neighbor does not support the refresh capability.

Each of these methods is effective, but can be disruptive to the network. The first method adds overhead while the Layer 3 switch learns and filters the neighbor or group entire route table, while the second method adds more overhead while the devices re-establish their BGP4 sessions.

You also can clear and reset the BGP4 routes that have been installed in the IP route table. Refer to [“Clearing and resetting BGP4 routes in the IP route table”](#) on page 1458.

Using soft reconfiguration

The *soft reconfiguration* feature places policy changes into effect without resetting the BGP4 session. Soft reconfiguration does not request the neighbor or group to send its entire BGP4 table, nor does the feature reset the session with the neighbor or group. Instead, the soft reconfiguration feature stores all the route updates received from the neighbor or group. When you request a soft reset of inbound routes, the software performs route selection by comparing the policies against the stored route updates, instead of requesting the neighbor BGP4 route table or resetting the session with the neighbor.

When you enable the soft reconfiguration feature, it sends a refresh message to the neighbor or group if the neighbor or group supports dynamic refresh. Otherwise, the feature resets the neighbor session. This step is required to ensure that the soft reconfiguration feature has a complete set of updates to use, and occurs only once, when you enable the feature. The feature accumulates all the route updates from the neighbor, eliminating the need for additional refreshes or resets when you change policies in the future.

To use soft reconfiguration:

- Enable the feature.
- Make the policy changes.
- Apply the changes by requesting a soft reset of the inbound updates from the neighbor or group.

Use the following CLI methods to configure soft configuration, apply policy changes, and display information for the updates that are filtered out by the policies.

Enabling soft reconfiguration

To configure a neighbor for soft reconfiguration, enter a command such as the following.

```
Brocade(config-bgp-router)#neighbor 10.10.200.102 soft-reconfiguration inbound
```

This command enables soft reconfiguration for updates received from 10.10.200.102. The software dynamically refreshes or resets the session with the neighbor, then retains all route updates from the neighbor following the reset.

Syntax: [no] neighbor <ip-addr> | <peer-group-name> soft-reconfiguration inbound

NOTE

The syntax related to soft reconfiguration is shown. For complete command syntax, refer to [“Adding BGP4 neighbors”](#) on page 1354.

Placing a policy change into effect

To place policy changes into effect, enter a command such as the following.

```
Brocade(config-bgp-router)#clear ip bgp neighbor 10.10.200.102 soft in
```

This command updates the routes by comparing the route policies against the route updates that the Layer 3 switch has stored. The command does not request additional updates from the neighbor or otherwise affect the session with the neighbor.

Syntax: clear ip bgp neighbor <ip-addr> | <peer-group-name> soft in

NOTE

If you do not specify “in”, the command applies to both inbound and outbound updates.

NOTE

The syntax related to soft reconfiguration is shown. For complete command syntax, refer to [“Dynamically refreshing routes”](#) on page 1455.

Displaying the filtered routes received from the neighbor or peer group

When you enable soft reconfiguration, the Layer 3 switch saves all updates received from the specified neighbor or peer group. This includes updates that contain routes that are filtered out by the BGP4 route policies in effect on the Layer 3 switch. To display the routes that have been filtered out, enter the following command at any level of the CLI.

```
Brocade#show ip bgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
      Prefix          Next Hop      Metric      LocPrf      Weight Status
1      3.0.0.0/8       192.168.4.106
      AS_PATH: 65001 4355 701 80
2      4.0.0.0/8       192.168.4.106
      AS_PATH: 65001 4355 1
3      4.60.212.0/22   192.168.4.106
      AS_PATH: 65001 4355 701 1 189
```

The routes displayed by the command are the routes that the Layer 3 switch BGP4 policies filtered out. The Layer 3 switch did not place the routes in the BGP4 route table, but did keep the updates. If a policy change causes these routes to be permitted, the Layer 3 switch does not need to request the route information from the neighbor, but instead uses the information in the updates.

Syntax: `show ip bgp filtered-routes [<ip-addr>] | [as-path-access-list <num>] | [detail] | [prefix-list <string>]`

The *<ip-addr>* parameter specifies the IP address of the destination network.

The **as-path-access-list** *<num>* parameter specifies an AS-path ACL. Only the routes permitted by the AS-path ACL are displayed.

The **detail** parameter displays detailed information for the routes. (The example above shows summary information.) You can specify any of the other options after **detail** to further refine the display request.

The **prefix-list** *<string>* parameter specifies an IP prefix list. Only the routes permitted by the prefix list are displayed.

NOTE

The syntax for displaying filtered routes is shown. For complete command syntax, refer to [“Displaying the BGP4 route table”](#) on page 1441.

Displaying all the routes received from the neighbor

To display all the route information received in route updates from a neighbor since you enabled soft reconfiguration, enter a command such as the following at any level of the CLI.

```
Brocade#show ip bgp neighbors 192.168.4.106 received-routes
      There are 97345 received routes from neighbor 192.168.4.106
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
      Prefix          Next Hop      Metric      LocPrf      Weight Status
1      3.0.0.0/8       192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 701 80
2      4.0.0.0/8       192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 1
3      4.60.212.0/22    192.168.4.106      100          0          BE
      AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8       192.168.4.106      100          0          BE
```

Syntax: `show ip bgp neighbors <ip-addr> received-routes [detail]`

The **detail** parameter displays detailed information for the routes. The example above shows summary information.

NOTE

The syntax for displaying received routes is shown. For complete command syntax, refer to [“Displaying BGP4 neighbor information”](#) on page 1428.

NOTE

The `show ip bgp neighbors <ip-addr> received-routes` syntax supported in previous software releases is changed to the following syntax: `show ip bgp neighbors <ip-addr> routes`.

Dynamically requesting a route refresh from a BGP4 neighbor

You can easily apply changes to filters that control BGP4 routes received from or advertised to a neighbor, without resetting the BGP4 session between the Layer 3 switch and the neighbor. For example, if you add, change, or remove a BGP4 address filter that denies specific routes received from a neighbor, you can apply the filter change by requesting a route refresh from the neighbor. If the neighbor also supports dynamic route refreshes, the neighbor resends its Adj-RIB-Out, its table of BGP4 routes. Using the route refresh feature, you do not need to reset the session with the neighbor.

The route refresh feature is based on the following specifications:

- RFC 2842. This RFC specifies the Capability Advertisement, which a BGP4 router uses to dynamically negotiate a capability with a neighbor.
- RFC 2858 for Multi-protocol Extension.

NOTE

The Brocade implementation of dynamic route refresh supports negotiation of IP version 4 unicasts only.

- RFC 2918, which describes the dynamic route refresh capability

The dynamic route refresh capability is enabled by default and cannot be disabled. When the Layer 3 switch sends a BGP4 OPEN message to a neighbor, the Layer 3 switch includes a Capability Advertisement to inform the neighbor that the Layer 3 switch supports dynamic route refresh.

NOTE

The option for dynamically refreshing routes received from a neighbor requires the neighbor to support dynamic route refresh. If the neighbor does not support this feature, the option does not take effect and the software displays an error message. The option for dynamically re-advertising routes to a neighbor does not require the neighbor to support dynamic route refresh.

To use the dynamic refresh feature, use either of the following methods.

Dynamically refreshing routes

The following sections describe how to dynamically refresh BGP4 routes to place new or changed filters into effect.

To request a dynamic refresh of all routes from a neighbor, enter a command such as the following.

```
Brocade(config-bgp-router)#clear ip bgp neighbor 192.168.1.170 soft in
```

This command asks the neighbor to send its BGP4 table (Adj-RIB-Out) again. The Layer 3 switch applies its filters to the incoming routes and adds, modifies, or removes BGP4 routes as necessary.

Syntax: `clear ip bgp neighbor all | <ip-addr> | <peer-group-name> | <as-num> [soft-outbound | soft [in | out]]`

The **all** | *<ip-addr>* | *<peer-group-name>* | *<as-num>* option specifies the neighbor. The *<ip-addr>* parameter specifies a neighbor by its IP interface with the Layer 3 switch. The *<peer-group-name>* specifies all neighbors in a specific peer group. The *<as-num>* parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor.

The **soft [in | out]** parameter specifies whether you want to refresh the routes received from the neighbor or sent to the neighbor:

- **soft in** does one of the following:
 - If you enabled soft reconfiguration for the neighbor or peer group, **soft in** updates the routes by comparing the route policies against the route updates that the Layer 3 switch has stored. Soft reconfiguration does not request additional updates from the neighbor or otherwise affect the session with the neighbor. Refer to [“Using soft reconfiguration”](#) on page 1452.
 - If you did not enable soft reconfiguration, **soft in** requests the neighbor entire BGP4 route table (Adj-RIB-Out), then applies the filters to add, change, or exclude routes.
 - If a neighbor does not support dynamic refresh, **soft in** resets the neighbor session.
- **soft out** updates all outbound routes, then sends the Layer 3 switch entire BGP4 route table (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters.

If you do not specify **in** or **out**, the Layer 3 switch performs both options.

NOTE

The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor. The **soft out** parameter updates all outbound routes, then sends the Layer 3 switch entire BGP4 route table (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters. Use **soft-outbound** if only the outbound policy is changed.

To dynamically resend all the Layer 3 switch BGP4 routes to a neighbor, enter a command such as the following.

```
Brocade(config-bgp-router)#clear ip bgp neighbor 192.168.1.170 soft out
```

This command applies its filters for outgoing routes to the Layer 3 switch BGP4 route table (Adj-RIB-Out), changes or excludes routes accordingly, then sends the resulting Adj-RIB-Out to the neighbor.

NOTE

The Brocade Layer 3 switch does not automatically update outbound routes using a new or changed outbound policy or filter when a session with the neighbor goes up or down. Instead, the Layer 3 switch applies a new or changed policy or filter when a route is placed in the outbound queue (Adj-RIB-Out).

To place a new or changed outbound policy or filter into effect, you must enter a **clear ip bgp neighbor** command regardless of whether the neighbor session is up or down. You can enter the command without optional parameters or with the **soft out** or **soft-outbound** option. Either way, you must specify a parameter for the neighbor (<ip-addr>, <as-num>, <peer-group-name>, or **all**).

Displaying dynamic refresh information

You can use the **show ip bgp neighbors** command to display information for dynamic refresh requests. For each neighbor, the display lists the number of dynamic refresh requests the Layer 3 switch has sent to or received from the neighbor and indicates whether the Layer 3 switch received confirmation from the neighbor that the neighbor supports dynamic route refresh.

The RefreshCapability field indicates whether this Layer 3 switch has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability. The statistics in the Message Sent and Message Received rows under Refresh-Req indicate how many dynamic refreshes have been sent to and received from the neighbor. The statistic is cumulative across sessions.

```

Brocade#show ip bgp neighbors 10.4.0.2
1  IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 100.0.0.1
    Description: neighbor 10.4.0.2
    State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
    PeerGroup: pgl
    Mutihop-EBGP: yes, ttl: 1
    RouteReflectorClient: yes
    SendCommunity: yes
    NextHopSelf: yes
    DefaultOriginate: yes (default sent)
    MaximumPrefixLimit: 90000
    RemovePrivateAs: : yes
    RefreshCapability: Received
    Route Filter Policies:
        Distribute-list: (out) 20
        Filter-list: (in) 30
        Prefix-list: (in) pfl
        Route-map: (in) setnp1 (out) setnp2
    Messages:
        Open      Update  KeepAlive  Notification  Refresh-Req
        Sent      : 1      1          1            0              0
        Received: 1      8          1            0              0
    Last Update Time: NLRI      Withdraw      NLRI      Withdraw
                     Tx: 0h0m59s  ---          Rx: 0h0m59s  ---
    Last Connection Reset Reason: Unknown
    Notification Sent:      Unspecified
    Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
    Byte Sent: 115, Received: 492
    Local host: 10.4.0.1, Local Port: 179
    Remote host: 10.4.0.2, Remote Port: 8053
    ISentSeq: 52837276 SendNext: 52837392 TotUnAck: 0
    TotSent: 116 ReTrans: 0 UnAckSeq: 52837392
    IRcvSeq: 2155052043 RcvNext: 2155052536 SendWnd: 16384
    TotalRcv: 493 DupliRcv: 0 RcvWnd: 16384
    SendQueue: 0 RcvQueue: 0 CngstWnd: 1460

```

Closing or resetting a neighbor session

You can close a neighbor session or resend route updates to a neighbor.

If you make changes to filters or route maps and the neighbor does not support dynamic route refresh, use the following methods to ensure that neighbors contain only the routes you want them to contain:

- If you close a neighbor session, the Layer 3 switch and the neighbor clear all the routes they learned from each other. When the Layer 3 switch and neighbor establish a new BGP4 session, they exchange route tables again. Use this method if you want the Layer 3 switch to relearn routes from the neighbor and resend its own route table to the neighbor.
- If you use the soft-outbound option, the Layer 3 switch compiles a list of all the routes it would normally send to the neighbor at the beginning of a session. However, before sending the updates, the Brocade Layer 3 switch also applies the filters and route maps you have configured to the list of routes. If the filters or route maps result in changes to the list of routes, the Layer 3 switch sends updates to advertise, change, or even withdraw routes on the

neighbor as needed. This ensures that the neighbor receives only the routes you want it to contain. Even if the neighbor already contains a route learned from the Layer 3 switch that you later decided to filter out, using the soft-outbound option removes that route from the neighbor.

You can specify a single neighbor or a peer group.

To close a neighbor session and thus flush all the routes exchanged by the Layer 3 switch and the neighbor, enter the following command.

```
Brocade#clear ip bgp neighbor all
```

Syntax: `clear ip bgp neighbor all | <ip-addr> | <peer-group-name> | <as-num> [soft-outbound | soft [in | out]]`

The `all` | `<ip-addr>` | `<peer-group-name>` | `<as-num>` option specifies the neighbor. The `<ip-addr>` parameter specifies a neighbor by its IP interface with the Layer 3 switch. The `<peer-group-name>` specifies all neighbors in a specific peer group. The `<as-num>` parameter specifies all neighbors within the specified AS. The `all` parameter specifies all neighbors.

To resend routes to a neighbor without closing the neighbor session, enter a command such as the following.

```
Brocade#clear ip bgp neighbor 10.0.0.1 soft out
```

Clearing and resetting BGP4 routes in the IP route table

To clear BGP4 routes from the IP route table and reset the routes, enter a command such as the following.

```
Brocade#clear ip bgp routes
```

Syntax: `clear ip bgp routes [<ip-addr>/<prefix-length>]`

NOTE

The `clear ip bgp routes` command has the same effect as the `clear ip route` command, but applies only to routes that come from BGP4.

Clearing traffic counters

You can clear the counters (reset them to 0) for BGP4 messages. To do so, use one of the following methods.

To clear the BGP4 message counter for all neighbors, enter the following command.

```
Brocade#clear ip bgp traffic
```

Syntax: `clear ip bgp traffic`

To clear the BGP4 message counter for a specific neighbor, enter a command such as the following.

```
Brocade#clear ip bgp neighbor 10.0.0.1 traffic
```

To clear the BGP4 message counter for all neighbors within a peer group, enter a command such as the following.

```
Brocade#clear ip bgp neighbor PeerGroup1 traffic
```


Syntax: `clear ip bgp neighbor all | <ip-addr> | <peer-group-name> | <as-num> traffic`

The **all** | *<ip-addr>* | *<peer-group-name>* | *<as-num>* option specifies the neighbor. The *<ip-addr>* parameter specifies a neighbor by its IP interface with the Layer 3 switch. The *<peer-group-name>* specifies all neighbors in a specific peer group. The *<as-num>* parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

Clearing route flap dampening statistics

To clear route flap dampening statistics, use the following CLI method.

NOTE

Clearing the dampening statistics for a route does not change the dampening status of the route.

To clear all the route dampening statistics, enter the following command at any level of the CLI.

```
Brocade#clear ip bgp flap-statistics
```

Syntax: `clear ip bgp flap-statistics [regular-expression <regular-expression> | <address> <mask> | neighbor <ip-addr>]`

The parameters are the same as those for the **show ip bgp flap-statistics** command (except the **longer-prefixes** option is not supported). Refer to “[Displaying route flap dampening statistics](#)” on page 1419.

NOTE

The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. Refer to “[Displaying route flap dampening statistics](#)” on page 1419.

Removing route flap dampening

You can un-suppress routes by removing route flap dampening from the routes. The Layer 3 switch allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI.

```
Brocade#clear ip bgp damping
```

Syntax: `clear ip bgp damping [<ip-addr> <ip-mask>]`

The *<ip-addr>* parameter specifies a particular network.

The *<ip-mask>* parameter specifies the network mask.

To un-suppress a specific route, enter a command such as the following.

```
Brocade#clear ip bgp damping 209.157.22.0 255.255.255.0
```

This command un-suppresses only the routes for network 209.157.22.0/24.

Clearing diagnostic buffers

The Layer 3 switch stores the following BGP4 diagnostic information in buffers:

- The first 400 bytes of the last packet that contained an error
- The last NOTIFICATION message either sent or received by the Layer 3 switch

To display these buffers, use options with the **show ip bgp neighbors** command. Refer to [“Displaying BGP4 neighbor information”](#) on page 1428.

This information can be useful if you are working with Brocade Technical Support to resolve a problem. The buffers do not identify the system time when the data was written to the buffer. If you want to ensure that diagnostic data in a buffer is recent, you can clear the buffers. You can clear the buffers for a specific neighbor or for all neighbors.

If you clear the buffer containing the first 400 bytes of the last packet that contained errors, all the bytes are changed to zeros. The Last Connection Reset Reason field of the BGP neighbor table also is cleared.

If you clear the buffer containing the last NOTIFICATION message sent or received, the buffer contains no data.

You can clear the buffers for all neighbors, for an individual neighbor, or for all the neighbors within a specific peer group.

To clear these buffers for neighbor 10.0.0.1, enter the following commands.

```
Brocade#clear ip bgp neighbor 10.0.0.1 last-packet-with-error
Brocade#clear ip bgp neighbor 10.0.0.1 notification-errors
```

Syntax: **clear ip bgp neighbor** **all** | *<ip-addr>* | *<peer-group-name>* | *<as-num>*
last-packet-with-error | **notification-errors**

The **all** | *<ip-addr>* | *<peer-group-name>* | *<as-num>* option specifies the neighbor. The *<ip-addr>* parameter specifies a neighbor by its IP interface with the Layer 3 switch. The *<peer-group-name>* specifies all neighbors in a specific peer group. The *<as-num>* parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

IP Multicast Traffic Reduction on Brocade FastIron X Series switches

Table 251 lists the individual Brocade FastIron X Series switches and the IP multicast traffic reduction features they support. These features are supported in the Layer 2, base Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 251 Supported IP multicast reduction features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
IGMP v1/v2 Snooping Global	Yes	For details about IP multicast traffic reduction on FastIron WS and Brocade FCX Series and Brocade ICX 6610 Series switches, refer to "IGMP snooping configuration" on page 1497.			Yes
IGMP v3 Snooping Global	Yes (*,G)				Yes
IGMP v1/v2/v3 Snooping per VLAN	Yes				Yes
IGMP v2/v3 Fast Leave (membership tracking)	Yes				Yes
PIM-SM V2 Snooping	Yes				Yes

IGMP snooping overview

When a device processes a multicast packet, by default, it broadcasts the packets to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to the CPU. This behavior causes some clients to receive unwanted traffic.

IGMP snooping provides multicast containment by forwarding traffic to only the ports that have IGMP receivers for a specific multicast group (destination address). A device maintains the IGMP group membership information by processing the IGMP reports and leave messages, so traffic can be forwarded to ports receiving IGMP reports.

An IPv4 multicast address is a destination address in the range of 224.0.0.0 to 239.255.255.255. Addresses of 224.0.0.X are reserved. Because packets destined for these addresses may require VLAN flooding, devices do not snoop in the reserved range. Data packets destined to addresses in the reserved range are flooded to the entire VLAN by hardware, and mirrored to the CPU. Multicast data packets destined for the non-reserved range of addresses are snooped. A client must send IGMP reports in order to receive traffic. If an application outside the reserved range requires VLAN flooding, the user must configure a static group that applies to the entire VLAN.

An IGMP device's responsibility is to broadcast general queries periodically, and to send group queries when receiving a leave message, to confirm that none of the clients on the port still want specific traffic before removing the traffic from the port. IGMP V2 lets clients specify what group (destination address) will receive the traffic but not to specify the source of the traffic. IGMP V3 is for source-specific multicast traffic, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An IGMP V3 device port state could be INCLUDE or EXCLUDE, and there are different types of group records for client reports.

The interfaces respond to general or group queries by sending a membership report that contains one or more of the following records associated with a specific group:

- Current-state record that indicates from which sources the interface wants to receive and not receive traffic. This record contains the source address of interfaces and whether or not traffic will be included (IS_IN) or not excluded (IS_EX) from this source.
- Filter-mode-change record. If the interface state changes from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if the interface state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- An IGMP V2 leave report is equivalent to a TO_IN (empty) record in IGMP V3. This record means that no traffic from this group will be received regardless of the source.
- An IGMP V2 group report is equivalent to an IS_EX (empty) record in IGMP V3. This record means that all traffic from this group will be received regardless of source.
- Source-list-change record. If the interface wants to add or remove traffic sources from its membership report, the report can contain an ALLOW record, which includes a list of new sources from which the interface wishes to receive traffic. It can also contain a BLOCK record, which lists the current traffic sources from which the interface wants to stop receiving traffic.

IGMP protocols provide a method for clients and a device to exchange messages, and let the device build a database indicating which port wants what traffic. The protocols do not specify forwarding methods. They require IGMP snooping or multicast protocols such as PIM or DVMRP to handle packet forwarding. PIM or DVMRP can route multicast packets within and outside a VLAN, while IGMP snooping can switch packets only within a VLAN.

If a VLAN is not IGMP snooping-enabled, it floods multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, IGMP packets are trapped to the CPU. Data packets are mirrored to the CPU in addition to being VLAN flooded. The CPU then installs hardware resources, so that subsequent data packets can be switched to desired ports in hardware without going to the CPU. If there is no client report or port to queriers for a data stream, the hardware resource drops it.

MAC-based implementation on FastIron X Series devices

On both switch and router software images, IGMP snooping is MAC-based. This differs from IGMP snooping on the BigIron router images, which match on both IP source and group (S,G) entries programmed in the Layer 4 CAM. In contrast, the FastIron X Series images match on Layer 2 destination MAC address entries (*,G).

When Layer 2 CAM is used, traffic is switched solely based on the destination MAC address. Consequently, traffic of the same group coming to the same port, regardless of its source, is switched in the same way. In addition, the lowest 23 bits of the group address are mapped to a MAC address. In this way, multiple groups (for example, 224.1.1.1 and 225.1.1.1) have the same

MAC address. Groups having the same MAC address are switched to the same destination ports, which are the superset of individual group output ports. Thus, the use of Layer 2 CAM might cause unwanted packets to be sent to some ports. However, the switch generally needs far less Layer 2 CAM than it does Layer 4 CAM, which is required for each stream with a different source and group.

Queriers and non-queriers

An IGMP snooping-enabled Brocade device can be configured as a querier (active) or non-querier (passive). An IGMP querier sends queries; a non-querier listens for IGMP queries and forwards them to the entire VLAN. VLANs can be independently configured to be queriers or non-queriers. If a VLAN has a connection to a PIM- or DVMRP-enabled port on another router, the VLAN must be configured as a non-querier. When multiple IGMP snooping devices are connected together, and there is no connection to a PIM- or DVMRP-enabled port, one of the devices must be configured as a querier. If multiple devices are configured as queriers, after these devices exchange queries, then all except the winner stop sending queries. The device with the lowest address becomes the querier. Although the system will work when multiple devices are configured as queriers, Brocade recommends that only one device (preferably the one with the traffic source) is configured as a querier.

The non-queriers always forward multicast data traffic and IGMP messages to router ports which receive IGMP queries or PIM or DVMRP hellos. Brocade recommends that you configure the device with the data traffic source (server) as a querier. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether there are any clients on the querier.

NOTE

In a topology of one or more connecting devices, at least one device must be running PIM or DVMRP configured as active. Otherwise, none of the devices can send out queries, and traffic cannot be forwarded to clients.

VLAN-specific configuration

IGMP snooping can be enabled on some VLANs or on all VLANs. Each VLAN can be independently configured to be a querier or non-querier and can be configured for IGMP V2 or IGMP V3. In general, the **ip multicast** commands apply globally to all VLANs except those configured with VLAN-specific multicast commands. The VLAN-specific multicast commands supersede the global **ip multicast** commands.

IGMP snooping can be configured for IGMP V2 or IGMP V3 on individual ports of a VLAN. An interface or router sends the queries and reports that include its IGMP version specified on it. The version configuration only applies to sending queries. The snooping device recognizes and processes IGMP V2 and IGMP V3 packets regardless of the version configuration.

To avoid version deadlock, an interface retains its version configuration even when it receives a report with a lower version.

Tracking and fast leave

Brocade devices support fast leave for IGMP V2, and tracking and fast leave for IGMP V3. Fast leave stops the traffic immediately when the port receives a leave message. Tracking traces all IGMP V3 clients. Refer to [“Enabling IGMP V3 membership tracking and fast leave for the VLAN”](#) on page 1474 and [“Enabling fast leave for IGMP V2”](#) on page 1475.

Support for IGMP snooping and Layer 3 multicast routing together on the same device

The Brocade device supports global Layer 2 IP multicast traffic reduction (IGMP snooping) and Layer 3 multicast routing (DVMRP or PIM-Sparse or PIM-Dense) together on the same device in the full Layer 3 software image, as long as the Layer 2 feature configuration is at the VLAN level.

Refer to [“IP multicast protocols and IGMP snooping on the same device”](#) on page 1606.

Configuration notes and feature limitations for IGMP snooping and Layer 3 multicast routing

The following details apply to FastIron X Series devices:

- Layer 2 IGMP snooping is automatically enabled with Layer 3 multicast routing. If Layer 3 multicast routing is enabled on your system, do not attempt to enable Layer 2 IGMP snooping.
- The default IGMP version is V2.
- A user can configure the maximum numbers of group address entries.
- An IGMP device can be configured to rate-limit the forwarding IGMP V2 membership reports to queriers.
- The device supports static groups. The device acts as a proxy to send IGMP reports for the static groups when receiving queries.
- A user can configure static router ports to force all multicast traffic to these specific ports.
- If a VLAN has a connection to a PIM- or DVMRP-enabled port on another router, the VLAN must be configured as a non-querier (passive). When multiple snooping devices connect together and there is no connection to PIM or DVMRP ports, one device must be configured as a querier (active). If multiple devices are configured as active (queriers), only one will keep sending queries after exchanging queries.
- The querier must configure an IP address to send out queries.
- IGMP snooping requires hardware resource. Hardware resource is installed only when there is data traffic. If resource is inadequate, the data stream without a resource is mirrored to the CPU in addition to being VLAN flooded, which can cause high CPU usage. Brocade recommends that you avoid global enabling of snooping unless necessary.
- IGMP snooping requires clients to send membership reports in order to receive data traffic. If a client application does not send reports, you must configure static groups on the snooping VLAN to force traffic to client ports. Note that servers (traffic sources) are not required to send IGMP memberships.
- FastIron X Series devices support VSRP together with IGMP snooping on the same interface.
- When VSRP or VSRP-aware is configured on a VLAN, the VLAN will support IGMP snooping version 2 only. IGMP version 3 will not be supported on the VLAN.
- High CPU utilization occurs when IGMP Snooping and PIM or DVMRP routing are enabled simultaneously on FastIron X Series devices. With IGMP snooping and PIM or DVMRP Routing enabled simultaneously on a given system, IP Multicast data packets received in the snooping VLAN(s) are forwarded to client ports via the hardware; however, copies of these packets will also be received and dropped by the CPU.

PIM SM traffic snooping overview

When multiple PIM sparse routers connect through a snooping-enabled device, the Brocade device always forwards multicast traffic to these routers. For example, PIM sparse routers R1, R2, and R3 connect through a device. Assume R2 needs traffic, and R1 sends it to the device, which forwards it to both R2 and R3, even though R3 does not need it. A PIM SM snooping-enabled device listens to join and prune messages exchanged by PIM sparse routers, and stops traffic to the router that sends prune messages. This allows the device to forward the data stream to R2 only.

PIM SM traffic snooping requires IGMP snooping to be enabled on the device. IGMP snooping configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for PIM SM join and prune messages sent from one PIM SM router to another through the device.

Application examples of PIM SM traffic snooping

[Figure 170](#) shows an example application of the PIM SM traffic snooping feature. In this example, a device is connected through an IP router to a PIM SM group source that is sending traffic for two PIM SM groups. The device also is connected to a receiver for each of the groups.

When PIM SM traffic snooping is enabled, the device starts listening for PIM SM join and prune messages and IGMP group membership reports. Until the device receives a PIM SM join message or an IGMP group membership report, the device forwards IP multicast traffic out all ports. Once the device receives a join message or group membership report for a group, the device forwards subsequent traffic for that group only on the ports from which the join messages or IGMP reports were received.

In this example, the router connected to the receiver for group 239.255.162.1 sends a join message toward the group source. Because PIM SM traffic snooping is enabled on the device, the device examines the join message to learn the group ID, then makes a forwarding entry for the group ID and the port connected to the receiver router. The next time the device receives traffic for 239.255.162.1 from the group source, the device forwards the traffic only on port 5/1, because that is the only port connected to a receiver for the group.

Notice that the receiver for group 239.255.162.69 is directly connected to the device. As a result, the device does not see a join message on behalf of the client. However, because IGMP snooping also is enabled, the device uses the IGMP group membership report from the client to select the port for forwarding traffic to group 239.255.162.69 receivers.

The IGMP snooping feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even though the device never sees a join message for the receiver for group 239.255.162.69, the device nonetheless learns about the receiver and forwards group traffic to the receiver.

The device stops forwarding IP multicast traffic on a port for a group if the port receives a prune message for the group.

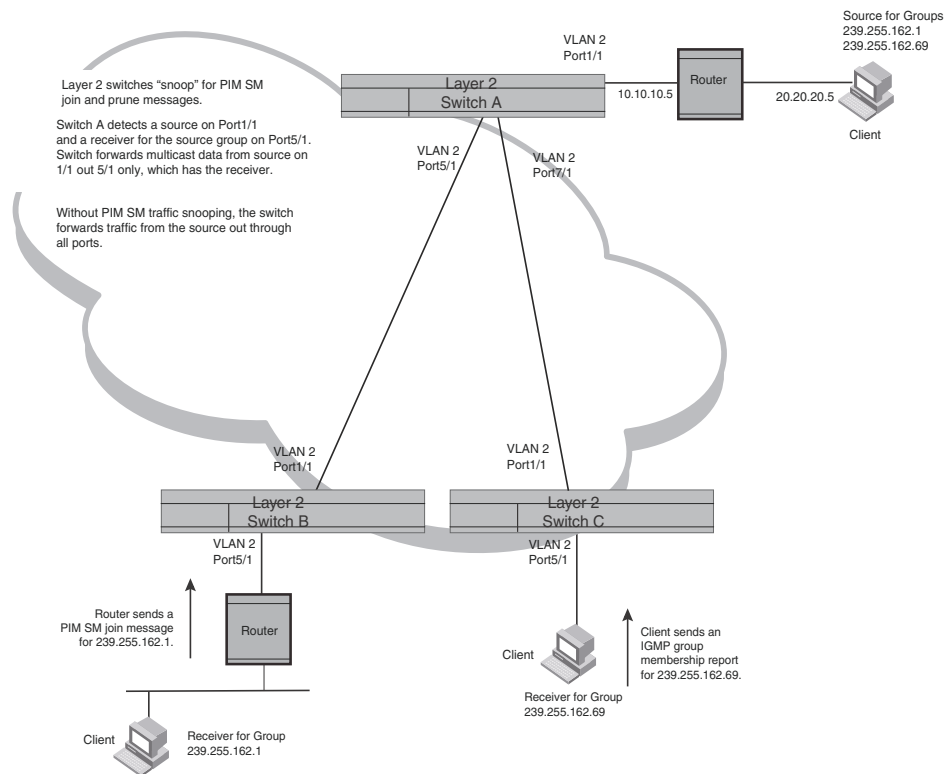
Notice that the ports connected to the source and the receivers are all in the same port-based VLAN on the device. This is required for the PIM SM snooping feature. The devices on the edge of the Global Ethernet cloud are configured for IGMP snooping and PIM SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

The following figure shows another example application for PIM SM traffic snooping. This example shows devices on the edge of a Global Ethernet cloud (a Layer 2 Packet over SONET cloud). Assume that each device is attached to numerous other devices such as other Layer 2 Switches and Layer 3 Switches (routers).

NOTE

This example assumes that the devices are actually Brocade devices running Layer 2 Switch software.

FIGURE 170 PIM SM traffic reduction in Global Ethernet environment



The devices on the edge of the Global Ethernet cloud are configured for IGMP snooping and PIM SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

Configuration notes and limitations for PIM SM snooping

- PIM SM snooping applies only to PIM SM version 2 (PIM SM V2).
- PIM SM traffic snooping is supported in the Layer 2, base Layer 3, and full Layer 3 code.
- IGMP snooping must be enabled on the device that will be running PIM SM snooping. The PIM SM traffic snooping feature requires IGMP snooping.

NOTE

Use the passive mode of IGMP snooping instead of the active mode. The passive mode assumes that a router is sending group membership queries as well as join and prune messages on behalf of receivers. The active mode configures the device to send group membership queries.

- All the device ports connected to the source and receivers or routers must be in the same port-based VLAN.
- The PIM SM snooping feature assumes that the group source and the device are in different subnets and communicate through a router. The source must be in a different IP subnet than the receivers. A PIM SM router sends PIM join and prune messages on behalf of a multicast group receiver only when the router and the source are in different subnet. When the receiver and source are in the same subnet, they do not need the router in order to find one another. They find one another directly within the subnet.

The device forwards all IP multicast traffic by default. Once you enable IGMP snooping and PIM SM traffic snooping, the device initially blocks all PIM SM traffic instead of forwarding it. The device forwards PIM SM traffic to a receiver only when the device receives a join message from the receiver. Consequently, if the source and the downstream router are in the same subnet, and PIM SM traffic snooping is enabled, the device blocks the PIM SM traffic and never starts forwarding the traffic. This is because the device never receives a join message from the downstream router for the group. The downstream router and group find each other without a join message because they are in the same subnet.

NOTE

If the “route-only” feature is enabled on a Layer 3 Switch, PIM SM traffic snooping will not be supported.

IGMP snooping configuration

Configuring IGMP snooping on a Brocade device consists of the following global, VLAN-specific, and port-specific tasks:

Global IGMP snooping tasks

Perform the following global tasks:

- [“Configuring the IGMP V3 snooping software resource limits”](#)
- [“Enabling IGMP snooping globally on the device”](#)
- [“Configuring the global IGMP mode”](#)
- [“Configuring the global IGMP version”](#)
- [“Modifying the age interval for group membership entries”](#)
- [“Modifying the query interval \(active IGMP snooping mode only\)”](#)
- [“Modifying the maximum response time”](#)
- [“Configuring report control” \(rate limiting\)](#)
- [“Modifying the wait time before stopping traffic when receiving a leave message”](#)
- [“Modifying the multicast cache age time”](#)
- [“Enabling or disabling error and warning messages”](#)

VLAN-specific IGMP snooping tasks

Perform the following VLAN-specific tasks:

- “Configuring the IGMP mode for a VLAN” (active or passive)
- “Disabling IGMP snooping on a VLAN”
- “Configuring the IGMP version for a VLAN”
- “Configuring static router ports.”
- “Turning off static group proxy”
- “Enabling IGMP V3 membership tracking and fast leave for the VLAN”
- “Enabling fast leave for IGMP V2”
- “Enabling fast convergence”

Port-specific IGMP snooping tasks

Perform the following port-specific tasks:

- “Disabling transmission and receipt of IGMP packets on a port”
- “Configuring the IGMP version for individual ports in a VLAN”

Configuring the IGMP V3 snooping software resource limits

By default, FastIron X Series devices support up to 512 IGMP snooping multicast cache (mcache) entries and a maximum of 8K IGMP group addresses. If necessary, you can change the default values using the procedures in this section.

About IGMP snooping mcache entries and group addresses

An IGMP snooping group address entry is created when an IGMP join message is received for a group. An IGMP snooping mcache entry is created when data traffic is received for that group. Each mcache entry represents one data stream, and multiple mcache entries (up to 32) can share the same hardware (MAC) address entry. The egress port list for the mcache entry is obtained from the IGMP group address entry. If there is no existing IGMP group address entry when an mcache entry is created, data traffic for that multicast group is dropped in hardware. If there is an existing IGMP group address entry when an mcache is created, data traffic for that multicast group is switched in hardware.

Changing the maximum number of supported IGMP snooping mcache entries

When IGMP snooping is enabled, by default, the system supports up to 512 IGMP snooping mcache entries. If necessary, you can change the maximum number of IGMP snooping cache entries supported on the device. To do so, enter the **system-max igmp-snoop-mcache** command.

```
BrocadeBrocade(config)# system-max igmp-snoop-mcache 2000
```

Syntax: [no] **system-max igmp-snoop-mcache** <num>

where <num> is a value ranges from 256 through 8192. The default is 512.

Setting the maximum number of IGMP group addresses

When IGMP snooping is enabled, by default, FastIron X Series devices support up to 4096 of IGMP group addresses, and the configurable range is from 256 through 8192. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limit are not processed. Enter the **system-max igmp-max-group-addr** command to define the maximum number of IGMP group addresses.

```
Brocade(config)# system-max igmp-max-group-addr 1600
```

Syntax: [no] **system-max igmp-max-group-addr** <num>

The <num> variable is a value ranging from 256 through 8192 and the default is 4096.

Enabling IGMP snooping globally on the device

When you globally enable IGMP snooping, you can specify IGMP V2 or IGMP V3. The **ip multicast version** command enables IGMP V3.

```
Brocade(config)# ip multicast version 3
```

Syntax: [no] **ip multicast version 2 | 3**

If you do not specify a version number, IGMP V2 is assumed.

Configuration notes for Layer 3 devices

- If Layer 3 multicast routing is enabled on your system, do not attempt to enable Layer 2 IGMP snooping. Layer 2 IGMP snooping is automatically enabled with Layer 3 multicast routing.
- If the “route-only” feature is enabled on the Layer 3 Switch, then IP multicast traffic reduction will not be supported.
- IGMP snooping is not supported on the default VLAN of Layer 3 Switches.

Configuring the IGMP mode

You can configure active or passive IGMP modes on the Brocade device. The default mode is passive. If you specify an IGMP mode for a VLAN, it overrides the global setting.

- **Active** - When active IGMP mode is enabled, a Brocade device actively sends out IGMP queries to identify multicast groups on the network, and makes entries in the IGMP table based on the group membership reports it receives.

NOTE

Routers in the network generally handle this operation. Use the active IGMP mode only when the device is in a stand-alone Layer 2 Switched network with no external IP multicast router attachments. In this case, enable the active IGMP mode on only one of the devices and leave the other devices configured for passive IGMP mode.

- **Passive** - When passive IGMP mode is enabled, it forwards reports to the router ports which receive queries. IGMP snooping in the passive mode does not send queries. However, it forwards queries to the entire VLAN.

Configuring the global IGMP mode

To globally set the IGMP mode to active, enter the following command.

```
Brocade(config)# ip multicast active
```

Syntax: [no] ip multicast [active | passive]

If you do not enter either *active* or *passive*, the passive mode is assumed.

Configuring the IGMP mode for a VLAN

If you specify an IGMP mode for a VLAN, it overrides the global setting.

To set the IGMP mode for VLAN 20 to active, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast active
```

Syntax: [no] multicast active | passive

Configuring the IGMP version

Use the procedures in this section to specify the IGMP version.

Configuring the global IGMP version

To globally specify IGMP V2 or IGMP V3, refer to [“Enabling IGMP snooping globally on the device”](#) on page 1469.

Configuring the IGMP version for a VLAN

You can specify the IGMP version for a VLAN. For example, the following commands configure VLAN 20 to use IGMP V3.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast version 3
```

Syntax: [no] multicast version 2 | 3

If no IGMP version is specified, then the globally-configured IGMP version is used. If an IGMP version is specified for individual ports, those ports use that version, instead of the VLAN version.

Configuring the IGMP version for individual ports in a VLAN

You can specify the IGMP version for individual ports in a VLAN. For example, the following commands configure ports 4, 5, and 6 to use IGMP V3. The other ports either use the IGMP version specified with the multicast version command, or the globally-configured IGMP version.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast port-version 3 ethernet 2/4 to 2/6
```

Syntax: [no] multicast port-version 2 | 3 ethernet <port> [ethernet <port> | to <port>]

Specify the <port> variable in one of the following formats:

- FSX 800 and FSX 1600 chassis devices – <slotnum/portnum>
- FESX compact switches – <portnum>

To specify a list of ports, enter each port as **ethernet <port>** followed by a space. For example, **ethernet 1/24 ethernet 6/24 ethernet 8/17**

To specify a range of ports, enter the first port in the range as **ethernet <port>** followed by the last port in the range. For example, **ethernet 1/1 to 1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 1/1 to 1/8 ethernet 1/24 ethernet 6/24 ethernet 8/17**.

Disabling IGMP snooping on a VLAN

When IGMP snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands cause IGMP snooping to be disabled for VLAN 20. This setting overrides the global setting.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast disable-multicast-snoop
```

Syntax: [no] multicast disable-multicast-snoop

Disabling transmission and receipt of IGMP packets on a port

When a VLAN is snooping-enabled, all IGMP packets are trapped to the CPU without hardware VLAN flooding. The CPU can block IGMP packets to and from a multicast-disabled port, and does not add it to the output interfaces of hardware resources. This prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic from these groups is VLAN flooded, including to disabled ports. Traffic from disabled ports cannot be blocked in hardware, and is switched in the same way as traffic from enabled ports.

This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

To disable transmission and receipt of IGMP packets on a port, enter the following commands.

```
Brocade(config)# interface ethernet 3
Brocade(config-if-e1000-3)# ip-multicast-disable
```

The above commands disable IGMP snooping on port 1/5 but does not affect the state of IGMP on other ports.

Syntax: [no] ip-multicast-disable

Modifying the age interval for group membership entries

When the device receives a group membership report, it makes an entry for that group in the IGMP group table. The age interval specifies how long the entry can remain in the table before the device receives another group membership report. When multiple devices connect together, all devices must be configured for the same age interval, which must be at least twice the length of the query interval, so that missing one report won't stop traffic. Non-querier age intervals must be the same as the age interval of the querier.

To modify the age interval, enter the following command.

```
Brocade(config)# ip multicast age-interval 280
```

Syntax: [no] ip multicast age-interval <interval>

The <interval> parameter specifies the aging time. You can specify a value from 20 through 7200 seconds. The default is 260 seconds.

Modifying the query interval (active IGMP snooping mode only)

If IP multicast traffic reduction is set to active mode, you can modify the query interval to specify how often the device sends general queries. When multiple queriers connect together, they must all be configured with the same query interval.

To modify the query interval, enter the following command.

```
Brocade(config)# ip multicast query-interval 120
```

Syntax: [no] ip multicast query-interval <interval>

The <interval> parameter specifies the time between queries. You can specify a value from 10 through 3600 seconds. The default is 125 seconds.

Modifying the maximum response time

The maximum response time is the number of seconds that a client can wait before responding to a query sent by the switch.

To change the maximum response time, enter the following command.

```
Brocade(config)# ip multicast max-response-time 5
```

Syntax: [no] ip multicast max-response-time <interval>

For <interval>, enter a value from 1 through 10 seconds. The default is 10 seconds.

Configuring report control

A device in passive mode forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding within the same group to no more than once every 10 seconds. This rate-limiting does not apply to the first report answering a group-specific query.

NOTE

This feature applies to IGMP V2 only. The leave messages are not rate limited.

IGMP V2 membership reports of the same group from different clients are considered to be the same and are rate-limited.

Use the **ip multicast report-control** command to alleviate report storms from many clients answering the upstream router query.

```
Brocade(config)# ip multicast report-control
```

Syntax: [no] ip multicast report-control

The original command, **ip igmp-report-control**, has been renamed to **ip multicast report-control**. The original command is still accepted; however, it is renamed when you issue a **show configuration** command.

Modifying the wait time before stopping traffic when receiving a leave message

You can define the wait time before stopping traffic to a port when a leave message is received. The device sends group-specific queries once per second to ask if any client in the same port still needs this group. Due to internal timer granularity, the actual wait time is between n and $(n+1)$ seconds (n is the configured value).

```
Brocade(config)# ip multicast leave-wait-time 1
```

Syntax: [no] **ip multicast leave-wait-time** *<num>*

<num> is the number of seconds from 1 through 5. The default is 2 seconds.

Modifying the multicast cache age time

You can set the time for an mcache to age out when it does not receive traffic. The traffic is hardware switched. One minute before aging out an mcache, the device mirrors a packet of this mcache to CPU to reset the age. If no data traffic arrives within one minute, this mcache is deleted. A lower value quickly removes resources consumed by idle streams, but it mirrors packets to CPU often. A higher value is recommended only data streams are continually arriving.

```
Brocade(config)# ip multicast mcache-age 180
```

Syntax: [no] **ip multicast mcache-age** *<num>*

<num> is the number of seconds from 60 through 3600. The default is 60 seconds.

Enabling or disabling error and warning messages

The device prints error or warning messages when it runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate-limited. You can turn off these messages by entering the following command.

```
Brocade(config)# ip multicast verbose-off
```

Syntax: [no] **ip multicast verbose-off**

Configuring static router ports

The Brocade device forwards all multicast control and data packets to router ports which receive queries. Although router ports are learned, you can force multicast traffic to specified ports even though these ports never receive queries. To configure static router ports, enter the following commands.

```
Brocade(config)#vlan 70
Brocade(config-vlan-70)# multicast router-port ethernet 4 to 5 ethernet 8
```

Syntax: [no] **multicast router-port ethernet** *<port>* [**ethernet** *<port>* | **to** *<port>*]

Specify the `<port>` variable in one of the following formats:

- FSX 800 and FSX 1600 chassis devices – `<slotnum/portnum>`
- FESX compact switches – `<portnum>`

To specify a list of ports, enter each port as **ethernet** `<port>` followed by a space. For example, **ethernet 1/24 ethernet 6/24 ethernet 8/17**

To specify a range of ports, enter the first port in the range as **ethernet** `<port>` followed by the last port in the range. For example, **ethernet 1/1 to 1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 1/1 to 1/8 ethernet 1/24 ethernet 6/24 ethernet 8/17**.

Turning off static group proxy

If a device has been configured for static groups, it acts as a proxy and sends membership reports for the static groups when it receives general or group-specific queries. When a static group configuration is removed, it is deleted from the active group table immediately. However, leave messages are not sent to the querier, and the querier must age out the group. Proxy activity can be turned off. The default is on. To turn proxy activity off for VLAN 20, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast proxy-off
```

Syntax: `[no] multicast proxy-off`

Enabling IGMP V3 membership tracking and fast leave for the VLAN

IGMP V3 gives clients membership tracking and fast leave capability. In IGMP V2, only one client on an interface needs to respond to a router's queries. This can leave some clients invisible to the router, making it impossible to track the membership of all clients in a group. When a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before it stops the traffic. You can configure the wait time using the **ip multicast leave-wait-time** command.

IGMP V3 requires every client to respond to queries, allowing the device to track all clients. When tracking is enabled, and an IGMP V3 client sends a leave message and there is no other client, the device immediately stops forwarding traffic to the interface. This feature requires the entire VLAN be configured for IGMP V3 with no IGMP V2 clients. If a client does not send a report during the specified group membership time (the default is 260 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can only track group membership; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each receives traffic streams from different sources. Client A receives a stream from (source_1, group1) and Client B receives a stream from (source_2, group1). The device still waits for the configured leave-wait-time before it stops the traffic because these two clients are in the same group. If the clients are in different groups, then the waiting period is not applied and traffic is stopped immediately.

To enable the tracking and fast leave feature for VLAN 20, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast tracking
```

Syntax: [no] multicast tracking

The membership tracking and fast leave features are supported for IGMP V3 only. If any port or any client is not configured for IGMP V3, then the multicast tracking command is ignored.

Enabling fast leave for IGMP V2

When a device receives an IGMP V2 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic. When fast-leave-v2 is configured, and when the device receives a leave message, it immediately stops forwarding to that port. The device does not send group specific-queries. You must ensure that no snooping-enabled ports have multiple clients. When two devices connect together, the querier must not be configured for fast-leave-v2, because the port might have multiple clients through the non-querier. The number of queries, and the waiting period (in seconds) can be configured using the **ip multicast leave-wait-time** command. The default is 2 seconds.

To configure fast leave for IGMP V2, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast fast-leave-v2
```

Syntax: [no] multicast fast-leave-v2

Enabling fast convergence

In addition to sending periodic general queries, an active device sends general queries when it detects a new port. However, because the device does not recognize the other device's port up event, multicast traffic might still require up to the query-interval time to resume after a topology change. Fast convergence allows the device to listen to topology change events in Layer 2 protocols such as spanning tree, and then send general queries to shorten the convergence time.

If the Layer 2 protocol cannot detect a topology change, fast convergence may not work in some cases. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this optimization, rather than a topology change. In this example, other devices will not receive topology change notifications, and will be unable to send queries to speed up the convergence. Fast convergence works well with the regular spanning tree protocol in this case.

To enable fast-convergence, enter the following commands.

```
Brocade(config)# vlan 70
Brocade(config-vlan-70)# multicast fast-convergence
```

Syntax: multicast fast-convergence

PIM SM snooping configuration

Configuring PIM SM snooping on a Brocade device consists of the following global and VLAN-specific tasks.

Global PIM SM snooping task

Perform the following global tasks:

- [“Enabling or disabling PIM SM snooping”](#)

VLAN-specific PIM SM snooping tasks

Perform the following VLAN-specific tasks:

- [“Enabling PIM SM snooping on a VLAN”](#)
- [“Disabling PIM SM snooping on a VLAN”](#)

Enabling or disabling PIM SM snooping

Use PIM SM snooping only in topologies where multiple PIM sparse routers connect through a device. PIM SM snooping does not work on a PIM dense mode router which does not send join messages and traffic to PIM dense ports is stopped. A PIM SM snooping-enabled device displays a warning if it receives PIM dense join or prune messages.

To enable PIM sparse snooping globally, enter the **ip pimsm-snooping** command.

```
Brocade(config)# ip pimsm-snooping
```

This command enables PIM SM traffic snooping. The PIM SM traffic snooping feature assumes that the network has routers that are running PIM SM.

NOTE

The device must be in passive mode before it can be configured for PIM SM snooping.

To disable the feature, enter the **no ip pimsm-snooping** command.

```
Brocade(config)# no ip pimsm-snooping
```

If you also want to disable IP multicast traffic reduction, enter the **no ip multicast** command.

```
Brocade(config)# no ip multicast
```

Syntax: [no]ip pimsm-snooping

Enabling PIM SM snooping on a VLAN

You can enable PIM SM snooping for a specific VLAN. For example, the following commands enable PIM SM snooping on VLAN 20.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast pimsm-snooping
```

Syntax: [no] multicast pimsm-snooping

Disabling PIM SM snooping on a VLAN

When PIM SM snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands disable PIM SM snooping for VLAN 20. This setting overrides the global setting.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast disable-pimsm-snoop
```

Syntax: [no] multicast disable-pimsm-snoop

IGMP snooping show commands

This section describes the **show** commands for IGMP snooping.

Displaying the IGMP snooping configuration

To display the global IGMP snooping configuration, enter the **show ip multicast** command at any level of the CLI.

```
Brocade# show ip multicast
Summary of all vlans. Please use "sh ip mu vlan <vlan-id>" for details
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260
VL10: cfg V3, vlan cfg passive, , pimsm (vlan cfg), 1 grp, 0 (SG) cache, no rtr
port
```

To display the IGMP snooping information for a specific VLAN, enter the following command.

```
Brocade# show ip multicast vlan 10
Version=3, Intervals: Query=10, Group Age=260, Max Resp=10, Other Qr=30
VL10: cfg V3, vlan cfg passive, , pimsm (vlan cfg), 3 grp, 1 (SG) cache, no rtr
port,
  e2      has      3 groups, non-QR (passive), default V3
  **** Warning! has V2 client (life=240),
    group: 239.0.0.3, life = 240
    group: 224.1.1.2, life = 240
    group: 224.1.1.1, life = 240

  e4      has      0 groups, non-QR (passive), default V3
```

Syntax: show ip multicast vlan [<vlan-id>]

If you do not specify a <vlan-id>, information for all VLANs is displayed.

The following table describes the information displayed by the **show ip multicast vlan** command.

Field	Description
Version	The global IGMP version. In this example, the device is configured for IGMP version 2.
Query	How often a querier sends a general query on the interface. In this example, the general queries are sent every 125 seconds.
Group Age	The number of seconds membership groups can be members of this group before aging out.
Max Resp	The maximum number of seconds a client waits before replying to a query.

Field	Description
Other Qr	How long it took a switch with a lower IP address to become a new querier. This value is 2 x Query + Max Resp.
cfg	The IGMP version for the specified VLAN. In this example, VL10: cfg V3 indicates that VLAN 10 is configured for IGMP V3.
vlan cfg	The IGMP configuration mode, which is either passive or active.
pimsm	Indicates that PIM SM is enabled on the VLAN.
rtr port	The router ports, which are the ports receiving queries.

Displaying IGMP snooping errors

To display information about possible IGMP errors, enter the **show ip multicast error** command.

```
Brocade# show ip multicast error
snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: **show ip multicast error**

The following table describes the output from the **show ip multicast error** command.

Field	Description
SW processed pkt	The number of multicast packets processed by IGMP snooping.
up-time	The time since the IGMP snooping is enabled.

Displaying IGMP group information

To display information about IGMP groups, enter the **show ip multicast group** command.

```
Brocade# show ip multicast group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 3 groups, 4 group-port, tracking_enabled
      group      p-port  ST    QR    life mode  source
1      224.1.1.2    1/33   no   yes   120  EX    0
2      224.1.1.1    1/33   no   yes   120  EX    0
3      226.1.1.1    1/35   yes  yes   100  EX    0
4      226.1.1.1    1/33   yes  yes   100  EX    0
```

In this example, an IGMP V2 group is in EXCLUDE mode with a source of 0. The group only excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed IGMP group information for a specific group, enter the **show ip multicast group detail** command.

```

Brocade# show ip multicast group 226.1.1.1 detail
Display group 226.1.1.1 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 2 group-port, tracking_enabled
      group      p-port  ST    QR    life mode    source
1      226.1.1.1    1/35   yes   yes   120  EX      0
      group: 226.1.1.1, EX, permit 0 (source, life):
      life=120, deny 0:
      group      p-port  ST    QR    life mode    source
2      226.1.1.1    1/33   yes   yes   120  EX      0
      group: 226.1.1.1, EX, permit 0 (source, life):
      life=120, deny 0:

```

If the tracking and fast leave features are enabled, you can display the list of clients that belong to a particular group by entering the following command.

```

Brocade# show ip multicast group 224.1.1.1 tracking
Display group 224.1.1.1 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 1 group-port, tracking_enabled
      group      p-port  ST    QR    life mode    source
*** Note: has 1 static groups to the entire vlan, not displayed here
1      224.1.1.1    1/33   no    yes   100  EX      0
      receive reports from 1 clients: (age)
      (2.2.100.2 60)

```

Syntax: `show ip multicast group [<group-address> [detail] [tracking]]`

If you want a report for a specific multicast group, enter that group's address for `<group-address>`.

Enter detail to display the source list of a specific VLAN.

Enter tracking for information on interfaces that have tracking enabled.

The following table describes the information displayed by the `show ip multicast group` command.

Field	Description
group	The address of the group (destination address in this case, 224.1.1.1)
p-port	The physical port on which the group membership was received.
ST	Yes indicates that the IGMP group was configured as a static group; No means the address was learned from reports.
QR	Yes means the port is a querier port; No means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the device.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE mode if it does not receive an "IS_EX" or "TO_EX" message during a certain period of time. The default is 260 seconds. There is no life displayed in INCLUDE mode.
mode	Indicates current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If an interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.
source	Identifies the source list that will be included or excluded on the interface. For example, if an IGMP V2 group is in EXCLUDE mode with a source of 0, the group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

Displaying IGMP snooping mcache information

The IGMP snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the **show ip multicast mcache** command.

```
Brocade# show ip multicast mcache
Example: (S G) cnt=: cnt is number of SW processed packets
        OIF: e1/22 TR(1/32,1/33), TR is trunk, e1/32 primary, e1/33 output
        vlan 10, 1 caches. use 1 VIDX
1      (10.10.10.2 239.0.0.3) cnt=0
        OIF: tag e2
        age=2s up-time=2s change=2s vidx=8191 (ref-cnt=1)
```

Syntax: show ip multicast mcache

The following table describes the output of the **show ip multicast mcache** command.

Field	Description
(source group)	Source and group addresses of this data stream. (* group) means match group only; (source group) means match both.
cnt	The number of packets processed in software. Packets are switched in hardware, which increases this number slowly.
OIF	The output interfaces. If entire vlan is displayed, this indicates that static groups apply to the entire VLAN.
age	The mcache age. The mcache will be reset to 0 if traffic continues to arrive, otherwise the mcache will be aged out when it reaches the time defined by the ip multicast mcache-age command.
uptime	The up time of this mcache in seconds.
vidx	Vidx specifies output port list index. Range is from 4096 through 8191
ref-cnt	The vidx is shared among mcaches having the same output interfaces. Ref-cnt indicates the number of mcaches using this vidx.

Displaying software resource usage for VLANs

To display information about the software resources used, enter the **show ip multicast resource** command.

```
Brocade# show ip multicast resource
          alloc in-use avail get-fail    limit  get-mem  size init
igmp group      256     1    255         0   32000      1    16  256
igmp phy port   1024     1   1023         0  200000      1    22 1024
... entries deleted ...
snoop mcache entry  128     2    126         0    8192      3    56  128
total pool memory 109056 bytes
has total 2 forwarding hash
VIDX sharing hash : size=2      anchor=997  2nd-hash=no  fast-trav=no
Available vidx: 4060. IGMP/MLD use 2
```

Syntax: show ip multicast resource

The following table describes the output displayed by the **show ip multicast resource** command.

Field	Description
alloc	The allocated number of units.
in-use	The number of units which are currently being used.
avail	The number of available units.
get-fail	This displays the number of resource failures. NOTE: It is important to pay attention to this field.
limit	The upper limit of this expandable field. The limit of <code>multicast group</code> is configured by the <code>system-max igmp-max-group-addr</code> command. The limit of <code>snoop mcache entry</code> is configured by the <code>system-max multicast-snoop-mcache</code> command.
get-mem	The number of memory allocation. This number must continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. More memory may be allocated if resources run out.
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying the status of IGMP snooping traffic

To display status information for IGMP snooping traffic, enter the **show ip multicast traffic** command.

```

Brocade# show ip multicast traffic
IGMP snooping: Total Recv: 22, Xmit: 26
Q: query, Qry: general Q, G-Qry: group Q, GSQry: group-source Q, Mbr: member
Recv      QryV2      QryV3      G-Qry      GSQry      MbrV2      MbrV3      Leave
VL1        0          0          0          0          4          0          0
VL70       18          0          0          0          0          0          0
Recv      IsIN      IsEX      ToIN      ToEX      ALLOW      BLOCK      Pkt-Err
VL1        0          4          0          0          0          0          0
VL70       0          0          0          0          0          0          0

Send      QryV2      QryV3      G-Qry      GSQry      MbrV2      MbrV3
VL1        0          0          8          0          0          0
VL70       0          0          0          0          0          18
VL70  pimsm-snooping, Hello: 12, Join/Prune: 9

```

Syntax: show ip multicast traffic

The following table describes the information displayed by the **show ip multicast traffic** command.

Field	Description
Q	Query
Qry	General Query
QryV2	Number of general IGMP V2 queries received or sent.
QryV3	Number of general IGMP V3 queries received or sent.
G-Qry	Number of group-specific queries received or sent.

Field	Description
GSQry	Number of group source-specific queries received or sent.
Mbr	The membership report.
MbrV2	The IGMP V2 membership report.
MbrV3	The IGMP V3 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times that additional source addresses were allowed on the interface.
BLK	Number of times that sources were removed from an interface.
Pkt-Err	Number of packets having errors, such as checksum.
Pism-snooping hello, join, prune	Number of PIM sparse hello, join, and prune packets

Displaying querier information

You can use the **show ip multicast vlan** command to display the querier information for a VLAN. This command displays the VLAN interface status and if there is any other querier present with the lowest IP address. The following list provides the combinations of querier possibilities:

- Active Interface with no other querier present
- Passive Interface with no other querier present
- Active Interface with other querier present
- Passive Interface with other querier present

Displaying the active interface with no other querier present

The following example shows the output in which the VLAN interface is active and no other querier is present with the lowest IP address.

```
Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260
VL10: dft V2, vlan cfg active, 0 grp, 0 (*G) cache, no rtr port,

    1/1/16 has    0 groups,
This interface is Querier
default V2

    1/1/24 has    0 groups,
This interface is Querier
default V2

    2/1/16 has    0 groups,
This interface is Querier
default V2

    2/1/24 has    0 groups,
This interface is Querier
```



```

default V2

    3/1/1   has    0 groups,
This interface is Querier
default V2

    3/1/4   has    0 groups,
This interface is Querier
default V2

```

Syntax: `show ip multicast vlan <vlan-id>`

If you do not specify a <vlan-id>, information for all VLANs is displayed.

Displaying the passive interface with no other querier present

The following example shows the output in which the VLAN interface is passive and no other querier is present with the lowest IP address.

```

Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260

VL10: dft V2, vlan cfg passive, 0 grp, 0 (*G) cache, no rtr port,

    1/1/16  has    0 groups,
This interface is non-Querier (passive)
default V2

    1/1/24  has    0 groups,
This interface is non-Querier (passive)
default V2

    2/1/16  has    0 groups,
This interface is non-Querier (passive)
default V2

    2/1/24  has    0 groups,
This interface is non-Querier (passive)
default V2

    3/1/1   has    0 groups,
This interface is non-Querier (passive)
default V2

    3/1/4   has    0 groups,
This interface is non-Querier (passive)
default V2

```

Displaying the active Interface with other querier present

The following example shows the output in which the VLAN interface is active and another querier is present with the lowest IP address.

```

Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260

VL10: dft V2, vlan cfg active, 7 grp, 6 (*G) cache, rtr ports,
      router ports: 2/1/24(260) 5.5.5.5, 3/1/4(260) 8.8.8.8,

```

IGMP snooping show commands

```
1/1/16 has 4 groups,
This interface is Querier
default V2
  group: 226.6.6.6, life = 240
  group: 228.8.8.8, life = 240
  group: 230.0.0.0, life = 240
  group: 224.4.4.4, life = 240

1/1/24 has 1 groups,
This interface is Querier
default V2
  group: 228.8.8.8, life = 240

2/1/16 has 4 groups,
This interface is Querier
default V2
  group: 226.6.6.6, life = 240
  group: 228.8.8.8, life = 240
  group: 230.0.0.0, life = 240
  group: 224.4.4.4, life = 240

2/1/24 has 2 groups,
This interface is non-Querier
Querier is 5.5.5.5
Age is 0
Max response time is 100
default V2
  **** Warning! has V3 (age=0) nbrs
  group: 234.4.4.4, life = 260
  group: 226.6.6.6, life = 260

3/1/1 has 4 groups,
This interface is Querier
default V2
  group: 238.8.8.8, life = 260
  group: 228.8.8.8, life = 260
  group: 230.0.0.0, life = 260
  group: 224.4.4.4, life = 260

3/1/4 has 1 groups,
This interface is non-Querier
Querier is 8.8.8.8
Age is 0
Max response time is 100
default V2
  **** Warning! has V3 (age=0) nbrs
  group: 236.6.6.6, life = 260
```

Displaying the passive interface with other querier present

The following example shows the output in which the VLAN interface is passive and another querier is present with the lowest IP address.

```
Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260

VL10: dft V2, vlan cfg passive, 7 grp, 6 (*G) cache, rtr ports,
  router ports: 2/1/24(260) 5.5.5.5, 3/1/4(260) 8.8.8.8,
```

```

1/1/16 has 4 groups,
This interface is non-Querier (passive)
default V2
  group: 226.6.6.6, life = 260
  group: 228.8.8.8, life = 260
  group: 230.0.0.0, life = 260
  group: 224.4.4.4, life = 260

1/1/24 has 1 groups,
This interface is non-Querier (passive)
default V2
  group: 228.8.8.8, life = 260

2/1/16 has 4 groups,
This interface is non-Querier (passive)
default V2
  group: 226.6.6.6, life = 260
  group: 228.8.8.8, life = 260
  group: 230.0.0.0, life = 260
  group: 224.4.4.4, life = 260

2/1/24 has 2 groups,
This interface is non-Querier (passive)
Querier is 5.5.5.5
Age is 0
Max response time is 100
default V2
  **** Warning! has V3 (age=0) nbrs
  group: 234.4.4.4, life = 260
  group: 226.6.6.6, life = 260

3/1/1 has 4 groups,
This interface is non-Querier (passive)
default V2
  group: 238.8.8.8, life = 260
  group: 228.8.8.8, life = 260
  group: 230.0.0.0, life = 260
  group: 224.4.4.4, life = 260

3/1/4 has 1 groups,
This interface is non-Querier (passive)
Querier is 8.8.8.8
Age is 0
Max response time is 100
default V2
  **** Warning! has V3 (age=0) nbrs
  group: 236.6.6.6, life = 260

```

PIM SM snooping show commands

This section shows how to display information about PIM SM snooping, including:

- [“Displaying PIM SM snooping information”](#)
- [“Displaying PIM SM snooping information on a Layer 2 switch”](#)
- [“Displaying PIM SM snooping information for a specific group or source group pair”](#)

Displaying PIM SM snooping information

To display PIM SM snooping information, enter the **show ip multicast pimsm-snooping** command.

```
Brocade# show ip multicast pimsm-snooping
vlan 1, has 2 caches.
1    (* 230.1.1.1) has 1 pim join ports out of 1 OIF
    1 (age=60)
    1 has 1 src: 20.20.20.66(60)
2    (* 230.2.2.2) has 1 pim join ports out of 1 OIF
    1 (age=60)
    1 has 1 src: 20.20.20.66(60)
```

This output shows the number of PIM join OIF out of the total OIF. The join or prune messages are source-specific. In this case, If the mcache is in (* G), the display function will also print the traffic source information.

Syntax: **show ip multicast pimsm-snooping** [*<vlan-id>*]

Use the *<vlan-id>* parameter to display PIM SM snooping information for a specific VLAN.

Displaying PIM SM snooping information on a Layer 2 switch

You can display PIM SM snooping information for all groups by entering the following command at any level of the CLI on a Layer 2 Switch.

```

Brocade# show ip multicast pimsm-snooping vlan 100
VLAN ID 100, total 3 entries
PIMSM Neighbor list:
    1.100.100.12      : 3/3 expire 120 s
    1.100.100.10      : 3/2 expire 170 s
    1.100.100.7       : 3/1 expire 160 s
1   Group: 224.0.1.22, fid 08ac, NO cam
    Forwarding Port: 3/3
    PIMv2 Group Port: 3/3
    (Source, Port) list: 1 entries
2   Group: 239.255.162.2, fid 08aa, cam 8
    Forwarding Port: 3/1 3/2
    PIMv2 Group Port: 3/1 3/2
    (Source, Port) list: 3 entries
3   Group: 239.255.163.2, fid 08a9, cam 10
    Forwarding Port: 3/1 3/2
    PIMv2 Group Port: 3/1 3/2
    (Source, Port) list: 3 entries
VLAN ID 4008, total 0 entries
PIMSM Neighbor list:

```

Syntax: `show ip mulitcast pimsm-snooping vlan <vlan-id>`

Enter the ID of the VLAN for the **vlan** *<vlan-id>* parameter.

If you want to display PIM SM snooping information for one source or one group, enter a command as in the following example. The command also displays the (source, port) list of the group.

```

Brocade# show ip multicst pimsm-snooping 239.255.163.2
Show pimsm snooping group 239.255.163.2 in all vlan
VLAN ID 100
  Group: 239.255.163.2, fid 08a9, cam 10
    Forwarding Port: 3/1 3/2
    PIMv2 Group Port: 3/1 3/2
    (Source, Port) list: 3 entries
      1   192.168.176.44, age=0, port: 3/2
      2   158.158.158.158, age=0, port: 3/1
      3   1.1.7.1, age=0, port: 3/2

```

Syntax: `show ip multicst pimsm-snooping <group-address> | <source-address>`

If the address you entered is within the range of source addresses, then the router treats it as the source address. Likewise, if the address falls in the range of group addresses, then the router assumes that you are requesting a report for that group.

The following table describes the information displayed by the **show ip multicast pimsm-snooping** command.

Field	Description
VLAN ID	The port-based VLAN to which the information listed below apply and the number of members in the VLAN.
PIM SM Neighbor list	The PIM SM routers that are attached to the Layer 2 Switch ports. The value following “expires” indicates how many seconds the Layer 2 Switch will wait for a hello message from the neighbor before determining that the neighbor is no longer present and removing the neighbor from the list.

Field	Description
Multicast Group	The IP address of the multicast group. NOTE: The fid and camindex values are used by Brocade Technical Support for troubleshooting.
Forwarding Port	The ports attached to the group receivers. A port is listed here when it receives a join message for the group, an IGMP membership report for the group, or both.
PIMv2 Group Port	The ports on which the Layer 2 Switch has received PIM SM join messages for the group.
Source, Port list	The IP address of each PIM SM source and the Layer 2 Switch ports connected to the receivers of the source.

Displaying PIM SM snooping information for a specific group or source group pair

To display PIM SM snooping information for a specific group, enter the following command at any level of the CLI.

```
Brocade# show ip multicast pimsm-snooping 230.1.1.1
Show pimsm snooping group 230.1.1.1 in all vlans
vlan 10, has 2 caches.
1 (*230.1.1.1) has 1 pim join ports out of 1 OIF
  1(age=120)
    1 has 1 src:20.20.20.66(120)
```

To display PIM SM snooping information for a specific (source, group) pair, enter the following command at any level of the CLI.

```
Brocade# show ip multicast pimsm-snooping 230.2.2.2 20.20.20.66
Show pimsm snooping source 20.20.20.66, group 230.2.2.2 in all vlans
vlan 10: (*230.2.2.2) has 1 pim join ports out of 2 OIF
  1(age=0)
    1 has 1 src:20.20.20.66(0)
```

Syntax: `show ip multicast pimsm-snooping <group-address> [<source-ip-address>]`

The Brocade device determines which address is the group address and which one is the source address based on the ranges that the address fall into. If the address is within the range of source addresses, then the router treats it as the source address. Likewise, if the address falls in the range of group addresses, then the router assumes it is a group address.

The following table describes the information displayed by the **show ip multicast pimsm-snooping** command.

Field	Description
vlan	The VLAN membership ID of the source.
port	The port on which the source is sending traffic. In this example, the port number is 1.
age	The age of the port, in seconds.
src	The source address and age. The age (number of seconds) is indicated in brackets immediately following the source.

Clear commands for IGMP snooping

The clear IGMP snooping commands must be used only in troubleshooting conditions, or to recover from errors.

Clearing the IGMP mcache

To clear the mcache on all VLANs, enter the **clear ip multicast mcache** command.

```
Brocade# clear ip multicast mcache
```

Syntax: clear ip multicast mcache

Clearing the mcache on a specific VLAN

To clear the mcache on a specific VLAN, enter the following command.

```
Brocade# clear ip multicast vlan 10 mcache
```

Syntax: clear ip multicast vlan <vlan-id> mcache

The <vlan-id> parameter specifies the specific VLAN in which the mcache needs to be cleared.

Clearing traffic on a specific VLAN

To clear the traffic counters on a specific VLAN, enter the following command.

```
Brocade# clear ip multicast vlan 10 traffic
```

Syntax: clear ip multicast vlan <vlan-id> traffic

The <vlan-id> parameter specifies the specific VLAN in which traffic counters needs to be cleared.

Clearing IGMP counters on VLANs

To clear IGMP snooping on error and traffic counters for all VLANs, enter the **clear ip multicast counters** command.

```
Brocade# clear ip multicast counters
```

Syntax: clear ip multicast counters

Clear commands for IGMP snooping

IP Multicast Traffic Reduction for FastIron WS and Brocade FCX and ICX Switches

Table 252 lists the individual Brocade FastIron switches and the IP multicast traffic reduction features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 252 Supported IP multicast reduction features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
IGMP v1/v2 Snooping Global	For details about IP multicast traffic reduction on FastIron X Series switches, refer to “IGMP snooping configuration” on page 1467.	Yes	Yes	Yes	Yes
IGMP v3 Snooping Global		Yes (S,G)	Yes (S,G)	Yes (S,G)	Yes
IGMP v1/v2/v3 Snooping per VLAN		Yes	Yes	Yes	Yes
IGMP v2/v3 Fast Leave (membership tracking)		Yes	Yes	Yes	Yes
PIM-SM V2 Snooping		Yes	Yes	Yes	Yes
Multicast static group traffic filtering (for snooping scenarios)		Yes	Yes	Yes	Yes

IGMP snooping overview

When a device processes a multicast packet, by default, the device broadcasts the packets to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to the CPU. This behavior causes some clients to receive unwanted traffic.

IGMP snooping provides multicast containment by forwarding traffic to only the ports that have IGMP receivers for a specific multicast group (destination address). A device maintains the IGMP group membership information by processing the IGMP reports and leave messages, so traffic can be forwarded to ports receiving IGMP reports.

An IPv4 multicast address is a destination address in the range of 224.0.0.0 to 239.255.255.255. Addresses of 224.0.0.X are reserved. Because packets destined for these addresses may require VLAN flooding, devices do not do snooping in the reserved range. Data packets destined to addresses in reserved range are flooded to the entire VLAN by hardware, and mirrored to the CPU. Multicast data packets destined for the non-reserved range of addresses are snooped. A client must send IGMP reports in order to receive traffic. If an application outside the reserved range requires VLAN flooding, the user must configure a static group that applies to the entire VLAN. In addition, a static group with the drop option can discard multicast data packets to a specified group in hardware, including addresses in the reserved range.

An IGMP device is responsible for broadcasting general queries periodically, and sending group queries when it receives a leave message, to confirm that none of the clients on the port still want specific traffic before removing the traffic from the port. IGMPv2 lets clients specify what group (destination address) will receive the traffic but not to specify the source of the traffic. IGMPv3 is for source-specific multicast traffic, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An IGMPv3 device port state could be INCLUDE or EXCLUDE, and there are different types of group records for client reports.

The interfaces respond to general or group queries by sending a membership report that contains one or more of the following records associated with a specific group:

- Current-state record that indicates from which sources the interface wants to receive and not receive traffic. This record contains the source address of interfaces and whether or not traffic will be included (IS_IN) or not excluded (IS_EX) from this source.
- Filter-mode-change record. If the interface state changes from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if the interface state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- An IGMPv2 leave report is equivalent to a TO_IN (empty) record in IGMPv3. This record means that no traffic from this group will be received regardless of the source.
- An IGMPv2 group report is equivalent to an IS_EX (empty) record in IGMPv3. This record means that all traffic from this group will be received regardless of source.
- Source-list-change record. If the interface wants to add or remove traffic sources from its membership report, the report can contain an ALLOW record, which includes a list of new sources from which the interface wishes to receive traffic. It can also contain a BLOCK record, which lists the current traffic sources from which the interface wants to stop receiving traffic.

IGMP protocols provide a method for clients and a device to exchange messages, and let the device build a database indicating which port wants what traffic. The protocols do not specify forwarding methods. They require IGMP snooping or multicast protocols such as PIM or DVMRP to handle packet forwarding. PIM and DVMRP can route multicast packets within and outside a VLAN, while IGMP snooping can switch packets only within a VLAN. Currently, FWS and FCX devices do not support multicast routing.

If a VLAN is not IGMP snooping-enabled, it floods multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, IGMP packets are trapped to the CPU. Data packets are mirrored to the CPU in addition to being VLAN flooded. The CPU then installs hardware resources, so that subsequent data packets can be switched to desired ports in hardware without going to the CPU. If there is no client report or port to queriers for a data stream, the hardware resource drops it. The hardware can either match the group address only (* G), or both the source and group (S G) of the data stream. If any IGMPv3 is configured in any port of a VLAN, this VLAN uses (S G) match; otherwise, it uses (* G). This is 32-bit IP address matching, not 23-bit multicast MAC address 01-00-5e-xx-xx-xx matching.

Brocade FWS and FCX devices have 16K of hardware resources allocated to MAC learning, IGMP, and MLD snooping. If a data packet does not match any of these resources, it might be sent to the CPU, which increases the CPU burden. This can happen if the device runs out of hardware resource, or is unable to install resources for a specific matching address due to hashing collision. The hardware hashes addresses into 16K entries, with some addresses hashed into the same entry. If the collision number in an entry is more than the hardware chain length, the resource cannot be installed. The chain length can be configured using the **hash-chain-length** command.

```
Brocade(config)# hash-chain-length 8
```

Syntax: [no] hash-chain-length <num>

The `<num>` value can be 4, 8, 16, or 32. Any other value is truncated to the closest lower ceiling. For example, a value of 15 is changed to 8. The default hash chain length is 4. A chain length of more than 4 may affect line rate switching.

NOTE

For this command to take effect, you must save the configuration and reload the switch.

The hardware resource limit applies only to the VLANs where IGMP snooping is enabled. Multicast streams are switched in hardware without using any pre-installed resources in a VLAN where snooping is not enabled.

FastIron devices support up to 8K of IGMP groups, which are produced by client membership reports.

Configuration notes for IGMP snooping

- Servers (traffic sources) are not required to send IGMP memberships.
- The default IGMP version is V2.
- Hardware resource is installed only when there is data traffic. If a VLAN is configured for IGMPv3, the hardware matches (S G), otherwise it matches (* G).
- A user can configure the maximum numbers of groups and hardware switched data streams.
- The device supports static groups that apply to the entire VLAN, or to just a few ports. The device acts as a proxy to send IGMP reports for the static groups when receiving queries. The static group has a drop option to discard multicast data packets in hardware.
- A user can configure static router ports to force all multicast traffic to these specific ports.
- The devices support fast leave for IGMPv2. Fast leave stops traffic immediately when the port receives a leave message.
- The devices support tracking and fast leave for IGMPv3, tracking all IGMPv3 clients. If the only client on a port leaves, traffic is stopped immediately.
- An IGMP device can be configured as a querier (active) or non-querier (passive). Queriers send queries. Non-queriers listen for queries and forward them to the entire VLAN.
- Every VLAN can be independently configured to be a querier or a non-querier.
- If a VLAN has a connection to a PIM- or DVMRP-enabled port on another router, this VLAN must be configured as a non-querier (passive). When multiple snooping devices connect together and there is no connection to PIM or DVMRP ports, one device must be configured as a querier (active). If multiple devices are configured as active (queriers), only one will keep sending queries after exchanging queries.
- An IGMP device can be configured to rate-limit the forwarding IGMPv2 membership reports to queriers.
- The querier must configure an IP address to send out queries.
- When VSRP or VSRP-aware is configured on a VLAN, the VLAN will support IGMP snooping version 2 only. IGMP version 3 will not be supported on the VLAN.
- When OSPF/PIM/VRRP is configured on a VLAN on the FCX, the VLAN will support IGMP snooping version 1 and version 2. IGMP snooping version 3 will not be supported on the VLAN.

The implementation allows snooping on some VLANs or all VLANs. Each VLAN can independently enable or disable IGMP, or configure V2 or V3. In general, global configuration commands **ip multicast** apply to every VLAN except those that have local multicast configurations (which supersede the global configuration). IGMP also allows independent configuration of individual ports in a VLAN for either IGMPv2 or IGMPv3. Configuring a specific version on a port or a VLAN only applies to the device's sent queries. The device always processes client reports of any version regardless of the configured version.

IGMP snooping requires hardware resources. If resources are inadequate, the data stream without a resource is mirrored to CPU in addition to being VLAN flooded, which can cause high CPU usage. Brocade recommends that you avoid global enabling of snooping unless necessary.

When any port in a VLAN is configured for IGMPv3, the VLAN matches both source and group (S G) in hardware switching. If no ports are configured for IGMPv3, the VLAN matches group only (* G). Matching (S G) requires more hardware resources than matching (* G) when there are multiple servers sharing the same group. For example, two data streams from different sources to the same group require two (S G) entries in IGMPv3, but only one (* G) in IGMPv2. To conserve resources, IGMPv3 must be used only in source-specific applications. When VLANs are independently configured for versions, some VLANs can match (* G) while others match (S G).

IGMP snooping requires clients to send membership reports in order to receive data traffic. If a client application does not send reports, you must configure static groups to force traffic to client ports. A static group can apply to only some ports or to the entire VLAN.

Querier and non-querier configuration

An IGMP snooping-enabled device can be configured as a querier (active) or non-querier (passive). An IGMP querier sends queries; a non-querier listens for IGMP queries and forwards them to the entire VLAN. Also, VLANs can be independently configured to be queriers or non-queriers. If a VLAN has a connection to a PIM- or DVMRP-enabled port on another router, the VLAN must be configured as a non-querier. When multiple IGMP snooping devices are connected together, and there is no connection to a PIM- or DVMRP-enabled port, one of the devices must be configured as a querier. If multiple devices are configured as queriers, after these devices exchange queries, then all except the winner stop sending queries. The device with the lowest address becomes the querier. Although the system will work when multiple devices are configured as queriers, Brocade recommends that only one device (preferably the one with the traffic source) is configured as a querier.

The non-queriers always forward multicast data traffic and IGMP messages to router ports which receive IGMP queries or PIM or DVMRP hellos. Brocade recommends that you configure the device with the data traffic source (server) as a querier. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether there are any clients on the querier.

NOTE

In a topology of one or more connecting devices, at least one device must be running PIM or DVMRP configured as active. Otherwise, none of the devices can send out queries, and traffic cannot be forwarded to clients.

VLAN specific configuration

You can configure IGMP snooping on some VLANs or on all VLANs. Each VLAN can be independently enabled or disabled for IGMP snooping, and can be configured for IGMPv2 or IGMPv3. In general, the **ip multicast** commands apply globally to all VLANs except those configured with VLAN-specific multicast commands. The VLAN-specific multicast commands supersede the global **ip multicast** commands.

IGMPv2 with IGMPv3

IGMP snooping can be configured for IGMPv2 or IGMPv3 on individual ports on a VLAN. An interface or router sends the queries and reports that include its IGMP version specified on it. The version configuration only applies to sending queries. The snooping device recognizes and processes IGMPv2 and IGMPv3 packets regardless of the version configuration.

To avoid version deadlock, an interface retains its version configuration even when it receives a report with a lower version.

PIM SM traffic snooping overview

When multiple PIM sparse routers connect through a snooping-enabled device, the device always forwards multicast traffic to these routers. For example, PIM sparse routers R1, R2, and R3 connect through a device. Assume R2 needs traffic, and R1 sends it to the device, which forwards it to both R2 and R3, even though R3 does not need it. A PIM snooping-enabled device listens to join and prune messages exchanged by PIM sparse routers, and stops traffic to the router that sends prune messages. This allows the device to forward the data stream to R2 only.

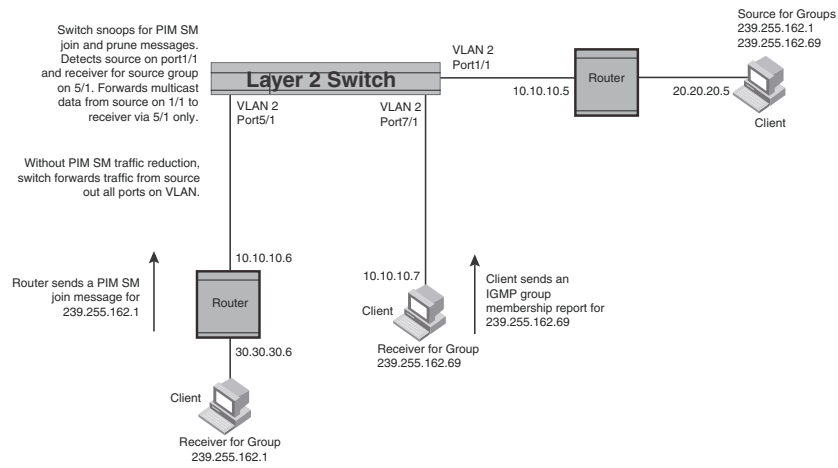
PIM SM traffic snooping requires IP multicast traffic reduction to be enabled on the device. IP multicast traffic reduction configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for PIM SM join and prune messages sent from one PIM SM router to another through the device.

NOTE

This feature applies only to PIM SM version 2 (PIM V2).

Application example of PIM SM traffic snooping

[Figure 171](#) shows an example application of the PIM SM traffic snooping feature. In this example, a device is connected through an IP router to a PIM SM group source that is sending traffic for two PIM SM groups. The device also is connected to a receiver for each of the groups.

FIGURE 171 PIM SM traffic reduction in an enterprise network**NOTE**

IP address 239.192.0.0/14 must be used for IPv4 Organization Local Scope.

When PIM SM traffic snooping is enabled, the device starts listening for PIM SM join and prune messages and IGMP group membership reports. Until the device receives a PIM SM join message or an IGMP group membership report, the device forwards IP multicast traffic out all ports. Once the device receives a join message or group membership report for a group, the device forwards subsequent traffic for that group only on the ports from which the join messages or IGMP reports were received.

In this example, the router connected to the receiver for group 239.255.162.1 sends a join message toward the group source. Because PIM SM traffic snooping is enabled on the device, the device examines the join message to learn the group ID, then makes a forwarding entry for the group ID and the port connected to the receiver router. The next time the device receives traffic for 239.255.162.1 from the group source, the device forwards the traffic only on port 5/1, because that is the only port connected to a receiver for the group.

Notice that the receiver for group 239.255.162.69 is directly connected to the device. As a result, the device does not see a join message on behalf of the client. However, since IP multicast traffic reduction also is enabled, the device uses the IGMP group membership report from the client to select the port for forwarding traffic to group 239.255.162.69 receivers.

The IP multicast traffic reduction feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even though the device never sees a join message for the receiver for group 239.255.162.69, the device nonetheless learns about the receiver and forwards group traffic to the receiver.

The device stops forwarding IP multicast traffic on a port for a group if the port receives a prune message for the group.

Notice that the ports connected to the source and the receivers are all in the same port-based VLAN on the device. This is required for the PIM SM snooping feature. The devices on the edge of the Global Ethernet cloud are configured for IP multicast traffic reduction and PIM SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

IGMP snooping configuration

To configure IGMP snooping on FWS, FCX, and ICX devices, you need to perform the following global and VLAN-specific tasks.

Global IGMP snooping tasks

Perform the following global tasks:

- [“Configuring the hardware and software resource limits”](#) on page 1498
- [“Enabling or disabling transmission and receipt of IGMP packets on a port”](#) on page 1498
- [“Configuring the global IGMP mode”](#) on page 1498 (Must be enabled for IGMP snooping)
- [“Modifying the age interval”](#) on page 1499
- [“Modifying the query interval \(active IGMP snooping mode only\)”](#) on page 1499
- [“Configuring the global IGMP version”](#) on page 1499
- [“Configuring report control”](#) on page 1499 (rate limiting)
- [“Modifying the wait time before stopping traffic when receiving a leave message”](#) on page 1500
- [“Modifying the multicast cache age time”](#) on page 1500
- [“Enabling or disabling error and warning messages”](#) on page 1500
- [“Enabling or disabling PIM sparse snooping”](#) on page 1500

VLAN-specific IGMP snooping tasks

Perform the following VLAN-specific tasks:

- [“Configuring the IGMP mode for a VLAN”](#) on page 1501 (active or passive)
- [“Disabling IGMP snooping for the VLAN”](#) on page 1501
- [“Disabling PIM sparse mode snooping for the VLAN”](#) on page 1501
- [“Configuring the IGMP version for the VLAN”](#) on page 1502
- [“Configuring the IGMP version for individual ports”](#) on page 1502
- [“Configuring static groups to the entire VLAN or to specific ports”](#) on page 1502
- [“Configuring static router ports”](#) on page 1503
- [“Turning off static group proxy”](#) on page 1503
- [“Enabling IGMPv3 membership tracking and fast leave for the VLAN”](#) on page 1503
- [“Configuring fast leave for IGMPv2”](#) on page 1504
- [“Enabling fast convergence”](#) on page 1504

Configuring the hardware and software resource limits

The system supports up to 8K of hardware-switched multicast streams. The configurable range is from 256 through 8192 with a default of 512. However, for ICX 6430 devices, the range is from 256 through 1024, and the default is 256. Enter the **system-max igmp-snoop-mcache** command to define the maximum number of IGMP snooping cache entries.

```
Brocade(config)# system-max igmp-snoop-mcache 8000
```

Syntax: [no] **system-max igmp-snoop-mcache** <num>

The system supports up to 32K of groups. The configurable range is from 256 through 32768 and the default is 8192. However, for ICX 6430, the range is from 256 through 1024, and the default is 256, while for ICX 6450, the default is 4096. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limits are not processed. Enter the **system-max igmp-max-group-addr** command to define the maximum number of IGMP group addresses.

```
Brocade(config)# system-max igmp-max-group-addr 1600
```

Syntax: [no] **system-max igmp-max-group-addr** <num>

Enabling or disabling transmission and receipt of IGMP packets on a port

When a VLAN is snooping-enabled, all IGMP packets are trapped to CPU without hardware VLAN flooding. The CPU can block IGMP packets to and from a multicast-disabled port, and does not add it to the output interfaces of hardware resources. This prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic from these groups is VLAN flooded, including to disabled ports. Traffic from disabled ports cannot be blocked in hardware, and is switched in the same way as traffic from enabled ports.

This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

```
Brocade(config)# interface ethernet 0/1/3
Brocade(config-if-e1000-0/1/3)# ip-multicast-disable
```

Syntax: [no] **ip-multicast-disable**

Configuring the global IGMP mode

You can configure active or passive IGMP modes on an FWS, FCX, or ICX device. The default mode is passive. If you specify an IGMP mode for a VLAN, it overrides the global setting.

- **Active** - When active IGMP mode is enabled, an FWS or FCX device actively sends out IGMP queries to identify multicast groups on the network, and makes entries in the IGMP table based on the group membership reports it receives.
- **Passive** - When passive IGMP mode is enabled, it forwards reports to the router ports which receive queries. IGMP snooping in the passive mode does not send queries. However, it forwards queries to the entire VLAN.

To globally set the IGMP mode to active, enter the **ip multicast active** command.

```
Brocade(config)# ip multicast active
```

Syntax: [no] **ip multicast** [active | passive]

If you do not enter either *active* or *passive*, the passive mode is assumed.

Modifying the age interval

When the device receives a group membership report, it makes an entry for that group in the IGMP group table. The age interval specifies how long the entry can remain in the table before the device receives another group membership report. When multiple devices connect together, all devices must be configured for the same age interval, which must be at least twice the length of the query interval, so that missing one report won't stop traffic. Non-querier age intervals must be the same as the age interval of the querier.

To modify the age interval, enter the **ip multicast age-interval** command.

```
Brocade(config)# ip multicast age-interval 280
```

Syntax: [no] **ip multicast age-interval** <interval>

The <interval> parameter specifies the aging time. You can specify a value from 20 through 7200 seconds. The default is 140 seconds.

Modifying the query interval (active IGMP snooping mode only)

For a device with an active IGMP mode, you can modify the query interval to specify how often the device sends general queries. When multiple queriers connect together, they must all be configured with the same query interval.

To modify the query interval, enter the **ip multicast query-interval** command.

```
Brocade(config)# ip multicast query-interval 120
```

Syntax: [no] **ip multicast query-interval** <interval>

The <interval> parameter specifies the time between queries. You can specify a value from 10 through 3600 seconds. The default is 125 seconds.

Configuring the global IGMP version

You can globally specify IGMPv2 or IGMPv3 for the device. The default is IGMPv2. For example, the **ip multicast version 3** command causes the device to use IGMPv3.

```
Brocade(config)# ip multicast version 3
```

Syntax: [no] **ip multicast version 2 | 3**

You can also optionally specify the IGMP version for individual VLANs, or individual ports within VLANs. When no IGMP version is specified for a VLAN, the global IGMP version is used. When an IGMP version is specified for individual ports within a VLAN, the ports use that version, instead of the VLAN version or the global version. The default is IGMPv2.

Configuring report control

A device in passive mode forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding within the same group to no more than once every 10 seconds. This rate-limiting does not apply to the first report answering a group-specific query.

NOTE

This feature applies to IGMPv2 only. The leave messages are not rate limited.

IGMPv2 membership reports of the same group from different clients are considered to be the same and are rate-limited.

Use the **ip multicast report-control** command to alleviate report storms from many clients answering the upstream router query.

```
Brocade(config)# ip multicast report-control
```

Syntax: [no] **ip multicast report-control**

The original command, **ip igmp-report-control**, has been renamed to **ip multicast report-control**. The original command is still accepted; however, it is renamed when you issue a **show configuration** command.

Modifying the wait time before stopping traffic when receiving a leave message

You can define the wait time before stopping traffic to a port when a leave message is received. The device sends group-specific queries once per second to ask if any client in the same port still needs this group. The value range is from 1 through 5, and the default is 2. Due to internal timer granularity, the actual wait time is between n and $(n+1)$ seconds (n is the configured value).

```
Brocade(config)# ip multicast leave-wait-time 1
```

Syntax: [no] **ip multicast leave-wait-time** <num>

Modifying the multicast cache age time

You can set the time for an mcache to age out when it does not receive traffic. The traffic is hardware switched. One minute before aging out an mcache, the device mirrors a packet of this mcache to CPU to reset the age. If no data traffic arrives within one minute, this mcache is deleted. A lower value quickly removes resources consumed by idle streams, but it mirrors packets to CPU often. A higher value is recommended only data streams are continually arriving. The range is from 60 through 3600 seconds, and the default is 60 seconds.

```
Brocade(config)# ip multicast mcache-age 180
```

Syntax: [no] **ip multicast mcache-age** <num>

Enabling or disabling error and warning messages

The device prints error or warning messages when it runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate-limited. You can turn off these messages by entering the **ip multicast verbose-off** command.

```
Brocade(config)# ip multicast verbose-off
```

Syntax: [no] **ip multicast verbose-off**

Enabling or disabling PIM sparse snooping

PIM snooping must be used only in topologies where multiple PIM sparse routers connect through a device. PIM snooping does not work on a PIM dense mode router which does not send join messages, and traffic to PIM dense ports is stopped. A PIM snooping-enabled device displays a warning if it receives PIM dense join or prune messages. Configure PIM sparse snooping by entering the **ip pimsm-snooping** command.

```
Brocade(config)# ip pimsm-snooping
```

Syntax: [no] ip pimsm-snooping

NOTE

The device must be in passive mode before it can be configured for PIM snooping.

Configuring the IGMP mode for a VLAN

You can configure a VLAN to use the active or passive IGMP mode. The default mode is passive. The setting specified for the VLAN overrides the global setting:

- **Active** - An active IGMP mode device actively sends out IGMP queries to identify multicast groups on the network, and makes entries in the IGMP table based on the group membership reports received.
- **Passive** - A passive IGMP mode device forwards reports to the router ports which receive queries. IGMP snooping in the passive mode forwards queries to the entire VLAN, but it does not send queries.

To set the IGMP mode for VLAN 20 to active, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast active
```

Syntax: [no] multicast active | passive

Disabling IGMP snooping for the VLAN

When IGMP snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands cause IGMP snooping to be disabled for VLAN 20. This setting overrides the global setting.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast disable-igmp-snoop
```

Syntax: [no] multicast disable-igmp-snoop

Enabling PIM sparse mode snooping for the VLAN

You can enable PIM snooping for a specific VLAN. For example, the following commands enable PIM snooping on VLAN 20.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast pimsm-snooping
```

Syntax: [no] multicast pimsm-snooping

Disabling PIM sparse mode snooping for the VLAN

When PIM snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands disable PIM snooping for VLAN 20. This setting overrides the global setting.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast disable-pimsm-snoop
```

Syntax: [no] multicast disable-pimsm-snoop

Configuring the IGMP version for the VLAN

You can specify the IGMP version for a VLAN. For example, the following commands configure VLAN 20 to use IGMPv3.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast version 3
```

Syntax: [no] multicast version 2 | 3

If no IGMP version is specified, then the globally-configured IGMP version is used. If an IGMP version is specified for individual ports, those ports use that version, instead of the VLAN version.

Configuring the IGMP version for individual ports

You can specify the IGMP version for individual ports in a VLAN. For example, the following commands configure ports 0/1/4, 0/1/5, 0/1/6 and 0/2/1 to use IGMPv3. The other ports either use the IGMP version specified with the multicast version command, or the globally-configured IGMP version.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast port-version 3 ethernet 0/2/1 ethernet 0/1/4
to 0/1/6
```

Syntax: [no] multicast port-version 2 | 3 <port-numbers>

Configuring static groups to the entire VLAN or to specific ports

A snooping-enabled VLAN cannot forward multicast traffic to ports that do not receive IGMP membership reports. If clients cannot send reports, you can configure a static group which applies to the entire VLAN or only to specific ports. The static group allows packets to be forwarded to the static group ports even though they have no client membership reports. The static group to the entire VLAN is used in VLAN flooding, which consumes less hardware resource than the static group to ports.

The static group drop option discards data traffic to a group in hardware. The group can be any multicast group including groups in the reserved range of 224.0.0.X. The drop option does not apply to IGMP packets, which are always trapped to CPU when snooping is enabled. The drop option applies to the entire VLAN, and cannot be configured for a port list. When the drop option is not specified, the group must exist outside the reserved range.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast static-group 224.1.1.1 count 2 ethernet 0/1/3
ethernet 0/1/5 to 0/1/7
Brocade(config-vlan-20)# multicast static-group 239.1.1.1 count 3 drop
Brocade(config-vlan-20)# multicast static-group 239.1.1.1
```

Syntax: [no] multicast static-group <ipv4-address> [count <num>] [<port-numbers> | drop]

The <ipv4-address> parameter is the address of the multicast group.

The count is optional, which allows a contiguous range of groups. Omitting the count <num> is equivalent to the count being 1.

If no <port-numbers> are entered, the static groups apply to the entire VLAN.

Configuring static router ports

FastIron Stackable devices forward all multicast control and data packets to router ports which receive queries. Although router ports are learned, you can force multicast traffic to specified ports even though these ports never receive queries. To configure static router ports, enter the following commands.

```
Brocade(config)# vlan 70
Brocade(config-vlan-70)# multicast router-port ethernet 0/1/4 to 0/1/5 ethernet
0/1/8
```

Syntax: [no] multicast router-port ethernet <port> [ethernet <port> | to <port>]

Specify the <port> variable in the format <stack-unit/slotnum/portnum>.

To specify a list of ports, enter each port as **ethernet <port>** followed by a space. For example, **ethernet 0/1/4 ethernet 0/1/5 ethernet 0/1/8**

To specify a range of ports, enter the first port in the range as **ethernet <port>** followed by the last port in the range. For example, **ethernet 0/1/1 to 0/1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 0/1/1 to 0/1/8 ethernet 0/1/24 ethernet 0/2/2 ethernet 0/2/4**.

Turning off static group proxy

If a device has been configured for static groups, it acts as a proxy and sends membership reports for the static groups when it receives general or group-specific queries. When a static group configuration is removed, it is deleted from active group table immediately. However, leave messages are not sent to the querier, and the querier must age the group out. Proxy activity can be turned off. The default is on. To turn proxy activity off for VLAN 20, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast proxy-off
```

Syntax: [no] multicast proxy-off

Enabling IGMPv3 membership tracking and fast leave for the VLAN

IGMPv3 gives clients membership tracking and fast leave capability. In IGMPv2, only one client on an interface needs to respond to a router's queries. This can leave some clients invisible to the router, making it impossible to track the membership of all clients in a group. When a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before it stops the traffic. You can configure the wait time using the **ip multicast leave-wait-time** command.

IGMPv3 requires every client to respond to queries, allowing the device to track all clients. When tracking is enabled, and an IGMPv3 client sends a leave message and there is no other client, the device immediately stops forwarding traffic to the interface. This feature requires the entire VLAN be configured for IGMPv3 with no IGMPv2 clients. If a client does not send a report during the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can only track group membership; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each receives traffic streams from different sources. Client A receives a stream from (source_1, group1) and Client B receives a stream from (source_2, group1). The device still waits for the configured leave-wait-time before it stops the traffic because these two clients are in the same group. If the clients are in different groups, then the waiting period is not applied and traffic is stopped immediately.

To enable the tracking and fast leave feature for VLAN 20, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast tracking
```

Syntax: [no] multicast tracking

The membership tracking and fast leave features are supported for IGMPv3 only. If any port or any client is not configured for IGMPv3, then the multicast tracking command is ignored.

Configuring fast leave for IGMPv2

When a device receives an IGMPv2 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic. When fast-leave-v2 is configured, and when the device receives a leave message, it immediately stops forwarding to that port. The device does not send group specific-queries. You must ensure that no snooping-enabled ports have multiple clients. When two devices connect together, the querier must not be configured for fast-leave-v2, because the port might have multiple clients through the non-querier. The number of queries, and the waiting period (in seconds) can be configured using the **ip multicast leave-wait-time** command. The default is 2 seconds.

To configure fast leave for IGMPv2, enter the following commands.

```
Brocade(config)# vlan 20
Brocade(config-vlan-20)# multicast fast-leave-v2
```

Syntax: [no] multicast fast-leave-v2

Enabling fast convergence

In addition to sending periodic general queries, an active device sends general queries when it detects a new port. However, because the device does not recognize the other device's port up event, multicast traffic might still require up to the query-interval time to resume after a topology change. Fast convergence allows the device to listen to topology change events in Layer 2 protocols such as spanning tree, and then send general queries to shorten the convergence time.

If the Layer 2 protocol cannot detect a topology change, fast convergence may not work in some cases. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this optimization, rather than a topology change. In this example, other devices will not receive topology change notifications, and will be unable to send queries to speed up the convergence. Fast convergence works well with the regular spanning tree protocol in this case.

To enable fast-convergence, enter the following commands.

```
Brocade(config)# vlan 70
Brocade(config-vlan-70)# multicast fast-convergence
```

Syntax: multicast fast-convergence

Displaying IGMP snooping information

This section describes the **show** commands for IGMP snooping.

Displaying IGMP errors

To display information about possible IGMP errors, enter the **show ip multicast error** command.

```
Brocade# show ip multicast error
snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: **show ip multicast error**

The following table describes the output from the **show ip multicast error** command.

Field	Description
SW processed pkt	The number of multicast packets processed by IGMP snooping.
up-time	The time since the IGMP snooping is enabled.

Displaying IGMP group information

To display information about IGMP groups, enter the **show ip multicast group** command.

```
Brocade# show ip multicast group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 3 groups, 4 group-port, tracking_enabled
      group      p-port  ST    QR    life mode    source
1      224.1.1.2    0/1/33 no    yes   120 EX      0
2      224.1.1.1    0/1/33 no    yes   120 EX      0
3      226.1.1.1    0/1/35 yes   yes   100 EX      0
4      226.1.1.1    0/1/33 yes   yes   100 EX      0
```

In this example, an IGMPv2 group is in EXCLUDE mode with a source of 0. The group only excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed IGMP group information, enter the **show ip multicast group detail** command.

```
Brocade# show ip multicast group 226.1.1.1 detail
Display group 226.1.1.1 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 2 group-port, tracking_enabled
      group      p-port  ST    QR    life mode    source
1      226.1.1.1    0/1/35 yes   yes   120 EX      0
  group: 226.1.1.1, EX, permit 0 (source, life):
    life=120, deny 0:
      group      p-port  ST    QR    life mode    source
2      226.1.1.1    0/1/33 yes   yes   120 EX      0
  group: 226.1.1.1, EX, permit 0 (source, life):
    life=120, deny 0:
```

If the tracking and fast leave features are enabled, you can display the list of clients that belong to a particular group by entering the following command.

```
Brocade# show ip multicast group 224.1.1.1 tracking
Display group 224.1.1.1 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 1 group-port, tracking_enabled
      group      p-port  ST    QR    life mode    source
*** Note: has 1 static groups to the entire vlan, not displayed here
1      224.1.1.1      0/1/33  no    yes    100  EX      0
      receive reports from 1 clients: (age)
      (2.2.100.2 60)
```

Syntax: `show ip multicast group [<group-address> [detail] [tracking]]`

If you want a report for a specific multicast group, enter that group's address for `<group-address>`.

Enter detail to display the source list of a specific VLAN.

Enter tracking for information on interfaces that have tracking enabled.

The following table describes the information displayed by the **show ip multicast group** command.

Field	Description
group	The address of the group (destination address in this case, 224.1.1.1)
p-port	The physical port on which the group membership was received.
ST	Yes indicates that the IGMP group was configured as a static group; No means the address was learned from reports.
QR	Yes means the port is a querier port; No means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the device.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE mode if it does not receive an "IS_EX" or "TO_EX" message during a certain period of time. The default is 140 seconds. There is no life displayed in INCLUDE mode.
mode	Indicates current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If an interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.
source	Identifies the source list that will be included or excluded on the interface. For example, if an IGMPv2 group is in EXCLUDE mode with a source of 0, the group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

Displaying IGMP snooping mcache information

The IGMP snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the **show ip multicast mcache** command.

```
Brocade# show ip multicast mcache
Example: (S G) cnt=: cnt: SW proc. count
        OIF: 0/1/22 TR(0/1/32,0/1/33), TR is trunk, 0/1/32 primary, 0/1/33 output
vlan 1, 1 caches. use 1 VIDX
1      (1.2.10.102 225.1.1.1) cnt=46
        OIF: 0/1/4
        age=0m up-time=45m vidx=4130 (ref-cnt=1)
vlan 70, 1 caches. use 1 VIDX
1      (* 226.1.2.3) cnt=69
        OIF: 0/1/14
        age=0m up-time=59m vidx=4129 (ref-cnt=1)
```

Syntax: show ip multicast mcache

The following table describes the output of the **show ip multicast mcache** command.

Field	Description
(source group)	Source and group addresses of this data stream. (* group) means match group only; (source group) means match both.
cnt	The number of packets processed in software. Packets are switched in hardware, which increases this number slowly.
OIF	The output interfaces. If entire vlan is displayed, this indicates that static groups apply to the entire VLAN.
age	The mcache age. The mcache will be reset to 0 if traffic continues to arrive, otherwise the mcache will be aged out when it reaches the time defined by the ip multicast mcache-age command.
uptime	The up time of this mcache in minutes.
vidx	Vidx specifies output port list index. Range is from 4096 to 8191
ref-cnt	The vidx is shared among mcaches having the same output interfaces. Ref-cnt indicates the number of mcaches using this vidx.

Displaying PIM sparse snooping information

PIM sparse mode snooping allows a device to listen for join or prune messages exchanged between PIM routers, which helps reduce unwanted traffic. To display PIM snooping information, enter the **show ip multicast pimsm-snooping** command.

```
Brocade# show ip multicast pimsm-snooping
vlan 1, has 1 caches.
1      (1.2.10.102 225.1.1.1) has 0 pim join ports out of 1 OIF
vlan 70, has 1 caches.
1      (* 226.1.2.3) has 2 pim join ports out of 2 OIF
        0/1/14 (age=60), 0/1/13 (age=60),
        0/1/14 has 1 src: 1.1.30.99(60)
        0/1/13 has 1 src: 1.1.30.99(60)
```

This output shows the number of OIF due to PIM out of the total OIF. The join or prune messages are source-specific. In this case, if the mcache is in (* G), the display function will also print the traffic source information.

Displaying software resource usage for VLANs

To display information about the software resources used, enter the **show ip multicast resource** command.

```
Brocade# show ip multicast resource
          alloc  in-use  avail get-fail    limit  get-mem  size init
igmp group          0      0      0         0    6000        0   16  256
igmp phy port        0      0      0         0 200000        0   22 1024
igmp exist phy port  256   107   149         0 1589248    107   29  256
igmp G/GS query      0      0      0         0  29696         0   12  128
igmp v3 source        0      0      0         0 500000         0   10 2000
igmp v3 tracking      0      0      0         0 118784         0   10  512
igmp glb sorted list  0      0      0         0 500000         0    8 2000
igmp generic link    128    12   116         0  29696        12    8  128
igmp generic multi-l 256     0   256         0  59392        12   16  256
igmp VIF             64    12    52         0  14848        12  170   64
snoop mcache entry    0      0      0         0    512         0  289  128
snoop group hash OIF  0      0      0         0 200000        0   12 2000
snoop group hash      0      0      0         0  59392         0   16  256
snoop router port     0      0      0         0  14848         0   15   64
snoop OIF            0      0      0         0  59392         0   23  256
IGMP snoop hitless s  0      0      0         0  29696         0   17  128
dynamic link hash     0      0      0         0  14848         0   20   64
snoop MCT OIF list    0      0      0         0 200000        0    8 2000
total pool memory 23424 bytes
has total 0 forwarding hash
VIDX sharing ha no no 997      0      0      0      0      0
Available vidx: 4003. IGMP/MLD use 0
```

Syntax: show ip multicast resource

The following table describes the output from the **show ip multicast resource** command.

Field	Description
alloc	The allocated number of units.
in-use	The number of units which are currently being used.
avail	The number of available units.
get-fail	This displays the number of resource failures. NOTE: It is important to pay attention to this field.
limit	The upper limit of this expandable field. The limit of multicast group is configured by the system-max igmp-max-group-addr command. The limit of snoop mcache entry is configured by the system-max multicast-snoop-mcache command.
get-mem	The number of memory allocation. This number should continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. More memory can be allocated if resources run out.
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying status of IGMP snooping traffic

To display status information for IGMP snooping traffic, enter the **show ip multicast traffic** command.

```
Brocade# show ip multicast traffic
IGMP snooping: Total Recv: 22, Xmit: 26
Q: query, Qry: general Q, G-Qry: group Q, GSQry: group-source Q, Mbr: member
Recv      QryV2      QryV3      G-Qry      GSQry      MbrV2      MbrV3      Leave
VL1        0         0         0         0         4         0         0
VL70       18         0         0         0         0         0         0
Recv      IsIN      IsEX      ToIN      ToEX      ALLOW      BLOCK      Pkt-Err
VL1        0         4         0         0         0         0         0
VL70       0         0         0         0         0         0         0

Send      QryV2      QryV3      G-Qry      GSQry      MbrV2      MbrV3
VL1        0         0         8         0         0         0
VL70       0         0         0         0         0         18
VL70  pimsm-snooping, Hello: 12, Join/Prune: 9
```

Syntax: show ip multicast traffic

The following table describes the information displayed by the **show ip multicast traffic** command.

Field	Description
Q	Query
Qry	General Query
QryV2	Number of general IGMPv2 queries received or sent.
QryV3	Number of general IGMPv3 queries received or sent.
G-Qry	Number of group-specific queries received or sent.
GSQry	Number of group source-specific queries received or sent.
Mbr	The membership report.
MbrV2	The IGMPv2 membership report.
MbrV3	The IGMPv3 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times that additional source addresses were allowed on the interface.
BLK	Number of times that sources were removed from an interface.
Pkt-Err	Number of packets having errors, such as checksum.
Pimsm-snooping hello, join, prune	Number of PIM sparse hello, join, and prune packets

Displaying IGMP snooping information by VLAN

You can display IGMP snooping information for all VLANs or for a specific VLAN. For example, to display IGMP snooping information for VLAN 70, enter the **show ip multicast vlan number** command.

```
Brocade# show ip multicast vlan 70
version=2, query-t=30, group-aging-t=140, max-resp-t=3, other-qr-present-t=63
VL70: dft V2, vlan cfg passive, , pimsm (vlan cfg), track, 0 grp, 1 (*G) cache,
rtr ports,
  router ports: 0/1/13(140) 1.1.70.3, 0/1/20(180) 1.1.70.2, 0/1/14(180)
0/1/13 has 0 groups, non-QR (passive), default V2
0/1/14 has 0 groups, non-QR (passive), default V2
0/1/20 has 0 groups, non-QR (passive), default V2
```

Syntax: **show ip multicast vlan** [**<vlan-id>**]

If you do not specify a **<vlan-id>**, information for all VLANs is displayed.

The following table describes the information displayed by the **show ip multicast vlan** command.

Field	Description
version	The IGMP version number
query-t	How often a querier sends a general query on the interface.
group-aging-t	The number of seconds membership groups can be members of this group before aging out.
rtr-port	The router ports which are the ports receiving queries. The display <code>router ports: 0/1/13(140) 1.1.70.3</code> means port 0/1/13 has a querier with 1.1.70.3 address, and a remaining life of 140 seconds.
max-resp-t	The maximum number of seconds a client waits before it replies to the query.
non-QR	Indicates that the port is a non-querier.
QR	Indicates that the port is a querier.
dft	The IGMP version for the specified VLAN. In this example, VL70: dft V2 indicates that the default IGMP version V2 is set for VLAN 70.

Displaying querier information

You can use the **show ip multicast vlan** command to display the querier information for a VLAN. This command displays the VLAN interface status and if there is any other querier present with the lowest IP address. The following list provides the combinations of querier possibilities:

- Active interface with no other querier present
- Passive interface with no other querier present
- Active interface with other querier present
- Passive interface with other querier present

Active interface with no other querier present

The following example shows the output in which the VLAN interface is active and no other querier is present with the lowest IP address.

```
Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260
VL10: dft V2, vlan cfg active, 0 grp, 0 (*G) cache, no rtr port,

    1/1/16 has    0 groups,
This interface is Querier
default V2

    1/1/24 has    0 groups,
This interface is Querier
default V2

    2/1/16 has    0 groups,
This interface is Querier
default V2

    2/1/24 has    0 groups,
This interface is Querier
default V2

    3/1/1  has    0 groups,
This interface is Querier
default V2

    3/1/4  has    0 groups,
This interface is Querier
default V2
```

Syntax: `show ip multicast vlan <vlan-id>`

If you do not specify a <vlan-id>, information for all VLANs is displayed.

Passive interface with no other querier present

The following example shows the output in which the VLAN interface is passive and no other querier is present with the lowest IP address.

```
Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260
VL10: dft V2, vlan cfg passive, 0 grp, 0 (*G) cache, no rtr port,

    1/1/16 has    0 groups,
This interface is non-Querier (passive)
default V2

    1/1/24 has    0 groups,
This interface is non-Querier (passive)
default V2

    2/1/16 has    0 groups,
This interface is non-Querier (passive)
default V2

    2/1/24 has    0 groups,
```

Displaying IGMP snooping information

```
This interface is non-Querier (passive)
default V2
```

```
3/1/1 has 0 groups,
This interface is non-Querier (passive)
default V2
```

```
3/1/4 has 0 groups,
This interface is non-Querier (passive)
default V2
```

Active interface with other querier present

The following example shows the output in which the VLAN interface is active and another querier is present with the lowest IP address.

```
Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260
```

```
VL10: dft V2, vlan cfg active, 7 grp, 6 (*G) cache, rtr ports,
router ports: 2/1/24(260) 5.5.5.5, 3/1/4(260) 8.8.8.8,
```

```
1/1/16 has 4 groups,
This interface is Querier
default V2
group: 226.6.6.6, life = 240
group: 228.8.8.8, life = 240
group: 230.0.0.0, life = 240
group: 224.4.4.4, life = 240
```

```
1/1/24 has 1 groups,
This interface is Querier
default V2
group: 228.8.8.8, life = 240
```

```
2/1/16 has 4 groups,
This interface is Querier
default V2
group: 226.6.6.6, life = 240
group: 228.8.8.8, life = 240
group: 230.0.0.0, life = 240
group: 224.4.4.4, life = 240
```

```
2/1/24 has 2 groups,
This interface is non-Querier
Querier is 5.5.5.5
Age is 0
Max response time is 100
default V2
**** Warning! has V3 (age=0) nbrs
group: 234.4.4.4, life = 260
group: 226.6.6.6, life = 260
```

```
3/1/1 has 4 groups,
This interface is Querier
default V2
group: 238.8.8.8, life = 260
group: 228.8.8.8, life = 260
group: 230.0.0.0, life = 260
```

```

group: 224.4.4.4, life = 260

3/1/4 has 1 groups,
This interface is non-Querier
Querier is 8.8.8.8
Age is 0
Max response time is 100
default V2
**** Warning! has V3 (age=0) nbrs
group: 236.6.6.6, life = 260

```

Passive interface with other querier present

The following example shows the output in which the VLAN interface is passive and another querier is present with the lowest IP address.

```

Brocade# show ip multicast vlan 10
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260

```

```

VL10: dft V2, vlan cfg passive, 7 grp, 6 (*G) cache, rtr ports,
router ports: 2/1/24(260) 5.5.5.5, 3/1/4(260) 8.8.8.8,

```

```

1/1/16 has 4 groups,
This interface is non-Querier (passive)
default V2
group: 226.6.6.6, life = 260
group: 228.8.8.8, life = 260
group: 230.0.0.0, life = 260
group: 224.4.4.4, life = 260

```

```

1/1/24 has 1 groups,
This interface is non-Querier (passive)
default V2
group: 228.8.8.8, life = 260

```

```

2/1/16 has 4 groups,
This interface is non-Querier (passive)
default V2
group: 226.6.6.6, life = 260
group: 228.8.8.8, life = 260
group: 230.0.0.0, life = 260
group: 224.4.4.4, life = 260

```

```

2/1/24 has 2 groups,
This interface is non-Querier (passive)
Querier is 5.5.5.5
Age is 0
Max response time is 100
default V2
**** Warning! has V3 (age=0) nbrs
group: 234.4.4.4, life = 260
group: 226.6.6.6, life = 260

```

```

3/1/1 has 4 groups,
This interface is non-Querier (passive)
default V2
group: 238.8.8.8, life = 260
group: 228.8.8.8, life = 260
group: 230.0.0.0, life = 260

```

```
group: 224.4.4.4, life = 260

3/1/4 has 1 groups,
This interface is non-Querier (passive)
Querier is 8.8.8.8
Age is 0
Max response time is 100
default V2
**** Warning! has V3 (age=0) nbrs

group: 236.6.6.6, life = 260
```

Clear IGMP snooping commands

The clear IGMP snooping commands must be used only in troubleshooting conditions, or to recover from errors.

Clearing IGMP counters on VLANs

To clear IGMP snooping on error and traffic counters for all VLANs, enter the **clear ip multicast counters** command.

```
Brocade# clear ip multicast counters
```

Syntax: clear ip multicast counters

Clearing IGMP mcache

To clear the mcache on all VLANs, enter the **clear ip multicast mcache** command.

```
Brocade# clear ip multicast mcache
```

Syntax: clear ip multicast mcache

Clearing mcache on a specific VLAN

To clear the mcache on a specific VLAN, enter the **clear ip multicast vlan <vlan-id> mcache** command.

```
Brocade# clear ip multicast vlan 10 mcache
```

Syntax: clear ip multicast vlan <vlan-id> mcache

The <vlan-id> parameter specifies the specific VLAN in which to clear the mcache.

Clearing traffic on a specific VLAN

To clear the traffic counters on a specific VLAN, enter the **clear ip multicast vlan <vlan-id> traffic** command.

```
Brocade# clear ip multicast vlan 10 traffic
```

Syntax: clear ip multicast vlan <vlan-id> traffic

The <vlan-id> parameter specifies the specific VLAN in which to clear the traffic counters.

IP Multicast Protocols

Table 253 lists the individual Brocade FastIron switches and the IP multicast features they support. These features are supported in the full Layer 3 software image only.

TABLE 253 Supported IP multicast features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Internet Group Management Protocol (IGMP) V1, V2, and V3 (for multicast routing scenarios)	Yes	No	Yes	Yes	No
IGMPv3 fast leave (for routing)	Yes	No	Yes	Yes	No
Protocol Independent Multicast Dense mode (PIM-DM) V1 (draft-ietf-pim-dm-05) and V2 (draft-ietf-pim-v2-dm-03)	Yes	No	Yes	Yes	No
Protocol Independent Multicast Sparse mode (PIM-SM) V2 (RFC 2362)	Yes	No	Yes	Yes	No
PIM passive	Yes	No	Yes	Yes	No
Distance Vector Multicast Routing Protocol (DVMRP) V2 (RFC 1075)	Yes	No	No	No	No
IP tunneling for multicast traffic	Yes	No	No	No	No
Anycast RP	Yes	No	No	No	No
Multicast Source Discovery Protocol (MSDP)	Yes	No	No	No	No
IGMP proxy	Yes	No	Yes	Yes	No
Passive multicast route insertion (PMRI)	Yes	No	Yes	Yes	No
Disable CPU processing for select multicast groups	Yes	No	No	No	No
Static multicast routes	Yes	No	No	No	No
Multicast trace route	Yes	No	No	No	No
IP multicast and IGMP snooping on the same device	Yes	No	Yes	Yes	No
ACLs to control multicast features	Yes	No	Yes	Yes	No
Static multicast groups	Yes	No	Yes	Yes	No

This chapter describes how to configure Brocade Layer 3 switches for Protocol Independent Multicast (PIM) and Distance Vector Multicast Routing Protocol (DVMRP).

NOTE

Each multicast protocol uses IGMP. IGMP is automatically enabled on an interface when you configure PIM or DVMRP and is disabled on the interface if you disable PIM or DVMRP.

NOTE

This chapter applies only to IP multicast routing. To configure Layer 2 multicast features, refer to [Chapter 34, “IP Multicast Traffic Reduction on Brocade FastIron X Series switches”](#) and [Chapter 35, “IP Multicast Traffic Reduction for FastIron WS and Brocade FCX and ICX Switches”](#).

IP multicast overview

Multicast protocols allow a group or channel to be accessed over different networks by multiple stations (clients) for the receipt and transmit of multicast data.

Distribution of stock quotes, video transmissions such as news services and remote classrooms, and video conferencing are all examples of applications that use multicast routing.

IPv4 multicast group addresses

In IPv4 Multicast, host groups are identified by Class D addresses, i.e., those with “1110” as their higher-order four bits. In Internet standard “dotted decimal” notation, these group addresses range from 224.0.0.0 to 239.255.255.255. However, the IANA IPv4 Multicast Address Registry (referencing RFC 3171) stipulates that the range 224.0.0.0 through 224.0.0.255 should not be used for regular multicasting applications.

“The range of addresses between 224.0.0.0 and 224.0.0.255, inclusive, is reserved for the use of routing protocols and other low-level topology discovery or maintenance protocols, such as gateway discovery and group membership reporting. Multicast routers should not forward any multicast datagram with destination addresses in this range, regardless of its TTL.”

Mapping of IPv4 Multicast group addresses to Ethernet MAC addresses

The IANA owns a block of Ethernet MAC addresses for Multicast usage that are in the range 0100.5e00.0000 through 0100.5e7f.ffff. For a given IPv4 Multicast group, there is a simple way of obtaining the appropriate Ethernet Destination MAC address that must be used in Layer 2 encapsulation. This is defined in RFC 1112, as follows:

“An IP host group address is mapped to an Ethernet multicast address by placing the low-order 23-bits of the IP address into the low-order 23 bits of the Ethernet multicast address 01-00-5E-00-00-00 (hex). Because there are 28 significant bits in an IP host group address, more than one host group address may map to the same Ethernet multicast address.”

NOTE

Since there are 5 bits in the IPv4 Group address that are not used in the mapping, there is a possibility for up to 32 IPv4 Multicast Groups to use the same Ethernet Destination MAC address. Taking this into account along with the reserved IPv4 Group address range, it is discouraged for applications to use IPv4 Multicast Group Addresses that may conflict with the reserved addresses at the Layer 2 level. This is because some devices may use just the Ethernet Destination MAC address to take actions on the packet.

Supported Layer 3 multicast routing protocols

Brocade Layer 3 switches support the multicast routing protocol DVMRP and PIM along with the Internet Group Membership Protocol (IGMP).

PIM and DVMRP are broadcast and pruning multicast protocol that deliver IP multicast datagrams. The protocol employs reverse path lookup check and pruning to allow source-specific multicast delivery trees to reach all group members. DVMRP and PIM build a different multicast tree for each source and destination host groups.

NOTE

DVMRP and PIM can concurrently operate on different ports of a Brocade Layer 3 switch.

Suppression of unregistered multicast packets

By default, unregistered multicast packets are always forwarded in hardware but not copied to the CPU. However on FSX platforms, if Layer 2 multicast (IGMP or MLD) or Layer 3 multicast (**router pim**) is enabled, then unregistered multicast packets are forwarded in hardware and also copied to the CPU.

Multicast terms

The following are commonly used terms in discussing multicast-capable routers. These terms are used throughout this chapter:

- **Node:** Refers to a router or Layer 3 switch.
- **Root Node:** The node that initiates the tree building process. It is also the router that sends the multicast packets down the multicast delivery tree.
- **Upstream:** Represents the direction from which a router receives multicast data packets. An upstream router is a node that sends multicast packets.
- **Downstream:** Represents the direction to which a router forwards multicast data packets. A downstream router is a node that receives multicast packets from upstream transmissions.
- **Group Presence:** Means that a multicast group has been learned from one of the directly connected interfaces. Members of the multicast group are present on the router.
- **Intermediate nodes:** Routers that are in the path between source routers and leaf routers.
- **Leaf nodes:** Routers that do not have any downstream routers.
- **Multicast Tree:** A unique tree is built for each source group (S,G) pair. A multicast tree is comprised of a root node and one or more nodes that are leaf or intermediate nodes.

Global IP multicast parameters

The following configurable parameters apply to PIM-DM, PIM-SM and DVMRP

- **Maximum number of PIM groups** – You can change the maximum number of groups of each type for which the software will allocate memory. By default, FastIron X Series Layer 3 switches support up to 1024 PIM groups. Brocade FCX Series Layer 3 switches support up to 4000 PIM groups.

- Maximum number of DVMRP groups – You can change the maximum number of groups for which the software will allocate memory. By default, FastIron X Series Layer 3 Switches support up to 1024 DVMRP groups.
- Internet Group Membership Protocol (IGMP) V1 and V2 parameters – You can change the query interval, group membership time, and maximum response time.
- Hardware forwarding of fragmented IP multicast packets – You can enable the Layer 3 switch to forward all fragments of fragmented IP multicast packets in hardware.

Changing dynamic memory allocation for IP multicast groups

Layer 3 switches support up to 1024 PIM groups and 1024 DVMRP groups by default. Memory for the groups is allocated dynamically as needed. For each protocol, previous releases support a maximum of 255 groups and 255 IGMP memberships.

NOTE

The number of interface groups you can configure for DVMRP and PIM is unlimited; therefore, the **system-max dvmrp-max-int-group** and the **system-max pim-max-int-group** commands that define their maximum table sizes have been removed.

The software allocates memory globally for each group, and also allocates memory separately for each interface IGMP membership in a multicast group. An interface becomes a member of a multicast group when the interface receives an IGMP group membership report. For example, if the Layer 3 switch learns about one multicast group, global memory for one group is used. In addition, if three interfaces on the device receive IGMP group membership reports for the group, interface memory for three IGMP memberships also is used.

Since the same group can use multiple allocations of memory (one for the group itself and one for each interface membership in the group), you can increase the maximum number of IGMP memberships, up to 8192.

NOTE

The total for IGMP memberships applies to the device, not to individual interfaces. You can have up to 8192 IGMP memberships on all the individual interfaces, not up to 8192 IGMP memberships on each interface.

Defining the maximum number of DVMRP cache entries

The DVMRP cache system parameter defines the maximum number of repeated DVMRP traffic being sent from the same source address and being received by the same destination address. To define this maximum, enter the **system-max dvmrp-mcache <num>** command.

```
Brocade(config)#system-max dvmrp-mcache 500
```

Syntax: **system-max dvmrp-mcache <num>**

The <num> parameter specifies the maximum number of multicast cache entries for DVMRP. Enter a number from 128 – 4096. The default is 512.

Defining the maximum number of PIM cache entries

The PIM cache system parameter defines the maximum number of repeated PIM traffic being sent from the same source address and being received by the same destination address. To define this maximum, enter the **system-max pim-mcache <num>** command.

```
Brocade(config)#system-max pim-mcache 999
```

Syntax: **system-max pim-mcache <num>**

The <num> parameter specifies the maximum number of multicast cache entries for PIM. Enter a number from 256 – 4096. The default is 1024.

Changing IGMP V1 and V2 parameters

IGMP allows Brocade routers to limit the multicast of IGMP packets to only those ports on the router that are identified as IP Multicast members. This section applies to Brocade devices that support IGMP versions 1 and 2.

The router actively sends out host queries to identify IP Multicast groups on the network, inserts the group information in an IGMP packet, and forwards the packet to IP Multicast neighbors.

The following IGMP V1 and V2 parameters apply to PIM and DVMRP:

- IGMP query interval – Specifies how often the Layer 3 switch queries an interface for group membership.
- IGMP group membership time – Specifies how many seconds an IP Multicast group can remain on a Layer 3 switch interface in the absence of a group report.
- IGMP maximum response time – Specifies how many seconds the Layer 3 switch will wait for an IGMP response from an interface before concluding that the group member on that interface is down and then removing the interface from the group.

To change these parameters, you must first enable IP multicast routing by entering the **ip multicast-routing** command at the global CLI level.

```
Brocade(config)#ip multicast-routing
```

Syntax: **[no] ip multicast-routing**

NOTE

You must enter the **ip multicast-routing** command before changing the global IP Multicast parameters. Otherwise, the changes do not take effect and the software uses the default values.

Modifying IGMP (V1 and V2) query interval period

The IGMP query interval period defines how often a router will query an interface for group membership.

To modify the default value for the IGMP (V1 and V2) query interval, enter the **ip igmp query-interval <num>** command.

```
Brocade(config)#ip igmp query-interval 120
```

Syntax: **ip igmp query-interval <num>**

The <num> variable specifies the IGMP query interval in number of seconds. Enter a value from 10 through 3600. The default value is 125.

Modifying IGMP (V1 and V2) membership time

The group membership time defines how long a group will remain active on an interface in the absence of a group report.

To define an IGMP (V1 and V2) membership time of 240 seconds, enter the **ip igmp group-membership-time <num>** command.

```
Brocade(config)#ip igmp group-membership-time 240
```

Syntax: **ip igmp group-membership-time <num>**

The <num> variable specifies the IGMP group membership time in number of seconds. Enter a value from 20 through 7200 seconds. The value you enter must be a little more than two times the query interval (2*query-interval +10). The default value is 260.

Modifying IGMP (V1 and V2) maximum response time

Maximum response time defines how long the Layer 3 switch will wait for an IGMP (V1 and V2) response from an interface before concluding that the group member on that interface is down, and then removing the interface from the group.

To change the IGMP (V1 and V2) maximum response time, enter the **ip igmp max-response-time <num>** command at the global CONFIG level of the CLI.

```
Brocade(config)#ip igmp max-response-time 8
```

Syntax: **[no] ip igmp max-response-time <num>**

The <num> parameter specifies the IGMP maximum response time in number of seconds. Enter a value from 1 through 10. The default is 10.

NOTE

Adding an interface to a multicast group

You can manually add an interface to a multicast group. This is useful in the following cases:

- Hosts attached to the interface are unable to add themselves as members of the group using IGMP.
- There are no members for the group attached to the interface.

When you manually add an interface to a multicast group, the Brocade device forwards multicast packets for the group but does not itself accept packets for the group.

You can manually add a multicast group to individual ports only. If the port is a member of a virtual routing interface, you must add the ports to the group individually.

To manually add a port to a multicast group, enter a command such as the following at the configuration level for the port.

```
Brocade(config-if-1/1)#ip igmp static-group 224.2.2.2
```

This command adds port 1/1 to multicast group 224.2.2.2.

To add a port that is a member of a virtual routing interface to a multicast group, enter a command such as the following at the configuration level for the virtual routing interface.

```
Brocade(config-vif-1)#ip igmp static-group 224.2.2.2 ethernet 5/2
```

This command adds port 5/2 in virtual routing interface 1 to multicast group 224.2.2.2.

Syntax: `[no] ip igmp static-group <ip-addr> [[count <number>] | drop | [ethernet <portnum>]]`

The `<ip-addr>` parameter specifies the group number.

The **count** `<number>` parameter specifies the number of contiguous groups per vlan. The `<number>` variable must be from 1 through 512.

The **drop** parameter specifies the number of dropped multicast packets. The **drop** option is available only for static groups on a vlan.

NOTE

On FSX devices, the **drop** option is not available for a multicast group.

The **ethernet** `<portnum>` parameter specifies the port number. Use this parameter if the port is a member of a virtual routing interface, and you are entering this command at the configuration level for the virtual routing interface.

Manually added groups are included in the group information displayed by the following commands:

- `show ip igmp group`
- `show ip pim group`

PIM Dense

NOTE

This section describes the “dense” mode of PIM, described in RFC 1075. Refer to [“PIM Sparse”](#) on page 1531 for information about PIM Sparse.

PIM was introduced to simplify some of the complexity of the routing protocol at the cost of additional overhead tied with a greater replication of forwarded multicast packets. PIM builds source-routed multicast delivery trees and employs reverse path check when forwarding multicast packets.

There are two modes in which PIM operates: Dense and Sparse. The Dense Mode is suitable for densely populated multicast groups, primarily in the LAN environment. The Sparse Mode is suitable for sparsely populated multicast groups with the focus on WAN.

PIM uses the IP routing table instead of maintaining its own, thereby being routing protocol independent.

Initiating PIM multicasts on a network

Once PIM is enabled on each router, a network user can begin a video conference multicast from the server on R1 as shown in [Figure 172](#). When a multicast packet is received on a PIM-capable router interface, the interface checks its IP routing table to determine whether the interface that received the message provides the shortest path back to the source. If the interface does provide the shortest path back to the source, the multicast packet is then forwarded to all neighboring PIM routers. Otherwise, the multicast packet is discarded and a prune message is sent back upstream.

In [Figure 172](#), the root node (R1) is forwarding multicast packets for group 229.225.0.1, which it receives from the server, to its downstream nodes, R2, R3, and R4. Router R4 is an intermediate router with R5 and R6 as its downstream routers. Because R5 and R6 have no downstream interfaces, they are leaf nodes. The receivers in this example are those workstations that are resident on routers R2, R3, and R6.

Pruning a multicast tree

As multicast packets reach these leaf routers, the routers check their IGMP databases for the group. If the group is not in a router IGMP database, the router discards the packet and sends a prune message to the upstream router. The router that discarded the packet also maintains the prune state for the source, group (S,G) pair. The branch is then pruned (removed) from the multicast tree. No further multicast packets for that specific (S,G) pair will be received from that upstream router until the prune state expires. You can configure the PIM Prune Timer (the length of time that a prune state is considered valid).

For example, in [Figure 172](#) the sender with address 207.95.5.1 is sending multicast packets to the group 229.225.0.1. If a PIM switch receives any groups other than that group, the switch discards the group and sends a prune message to the upstream PIM switch.

In [Figure 173](#), switch S5 is a leaf node with no group members in its IGMP database. Therefore, the switch must be pruned from the multicast tree. S5 sends a prune message upstream to its neighbor switch S4 to remove itself from the multicast delivery tree and install a prune state, as seen in [Figure 173](#). Switch S5 will not receive any further multicast traffic until the prune age interval expires.

When a node on the multicast delivery tree has all of its downstream branches (downstream interfaces) in the prune state, a prune message is sent upstream. In the case of S4, if both S5 and S6 are in a prune state at the same time, S4 becomes a leaf node with no downstream interfaces and sends a prune message to S1. With S4 in a prune state, the resulting multicast delivery tree would consist only of leaf nodes S2 and S3.

FIGURE 172 Transmission of multicast packets from the source to host group members

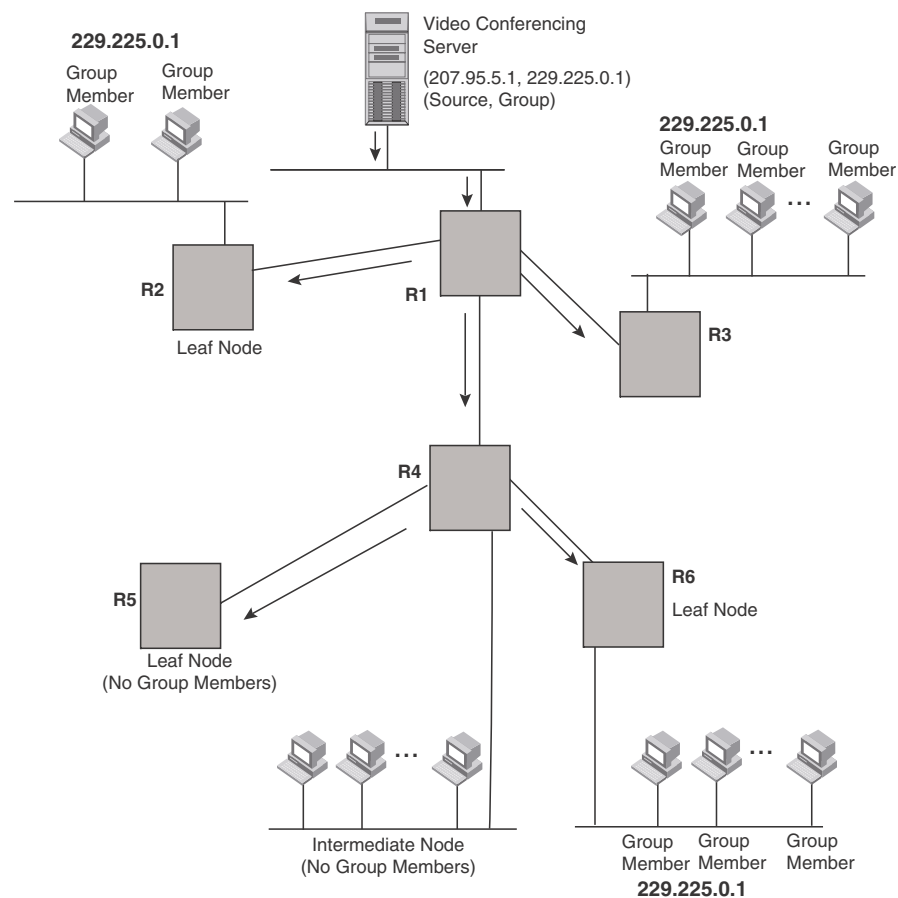
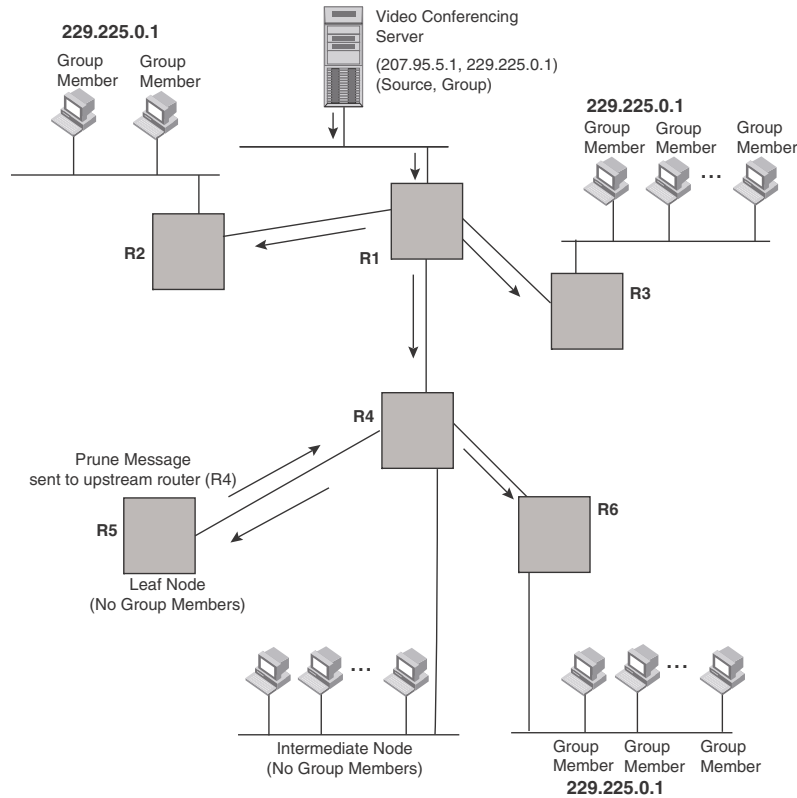


FIGURE 173 Pruning leaf nodes from a multicast tree



Grafts to a multicast Tree

A PIM switch restores pruned branches to a multicast tree by sending graft messages towards the upstream switch. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream switch.

In the example above, if a new 229.255.0.1 group member joins on switch S6, which was previously pruned, a graft is sent upstream to S4. Since the forwarding state for this entry is in a prune state, S4 sends a graft to S1. Once S4 has joined the tree, S4 and S6 once again receive multicast packets.

Prune and graft messages are continuously used to maintain the multicast delivery tree. No configuration is required on your part.

PIM DM versions

Brocade devices support PIM DM V1 and V2. The default is V2. You can specify the version on an individual interface basis.

The primary difference between PIM DM V1 and V2 is the methods the protocols use for messaging:

- PIM DM V1 – uses the Internet Group Management Protocol (IGMP) to send messages
- PIM DM V2 – sends messages to the multicast address 224.0.0.13 (ALL-PIM-ROUTERS) with protocol number 103

The CLI commands for configuring and managing PIM DM are the same for V1 and V2. The only difference is the command you use to enable the protocol on an interface.

NOTE

Version 2 is the default PIM DM version. The only difference between version 1 and version 2 is the way the protocol sends messages. The change is not apparent in most configurations. You can use version 2 instead of version 1 with no impact to your network. However, if you want to continue to use PIM DM V1 on an interface, you must change the version, then save the configuration.

NOTE

The note above does not mean you can run different PIM versions on devices that are connected to each other. The devices must run the same version of PIM. If you want to connect a Layer 3 switch running PIM to a device that is running PIM V1, you must change the version on the Layer 3 switch to V1 (or change the version on the device to V2, if supported).

PIM DM configuration

NOTE

This section describes how to configure the “dense” mode of PIM, described in RFC 1075. Refer to [“PIM Sparse configuration”](#) on page 1533 for information about configuring PIM Sparse.

Enabling PIM on the router and an interface

By default, PIM is disabled. To enable PIM, perform the following:

- Enable the feature globally.
- Configure the IP interfaces that will use PIM.
- Enable PIM locally on the ports that have the IP interfaces you configured for PIM.

Suppose you want to initiate the use of desktop video for fellow users on a sprawling campus network. All destination workstations have the appropriate hardware and software but the Brocade routers that connect the various buildings need to be configured to support PIM multicasts from the designated video conference server as shown in [Figure 172](#) on page 1523.

PIM is enabled on each of the Brocade routers shown in [Figure 172](#), on which multicasts are expected. You can enable PIM on each router independently or remotely from one of the routers with a Telnet connection. Follow the same steps for each router. A reset of the router is required when PIM is first enabled. Thereafter, all changes are dynamic.

Globally enabling and disabling PIM

To globally enable PIM, enter the following command.

```
Brocade(config)#router pim
```

Syntax: **[no] router pim**

The behavior of the **[no] router pim** command is as follows:

- Entering **router pim** command to enable PIM does not require a software reload.
- Entering a **no router pim** command removes all configuration for PIM multicast on a Layer 3 switch (**router pim** level) only.

Globally Enabling and Disabling PIM without Deleting Multicast Configuration

As stated above entering a **no router pim** command deletes the PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command.

```
Brocade(config)#router pim
Brocade(config-pim-router)#disable-pim
```

Syntax: [no] disable-pim

Use the [no] version of the command to re-enable PIM.

Enabling a PIM version

Using the CLI

To enable PIM on an interface, globally enable PIM, then enable PIM on interface 3, enter the following commands.

```
Brocade(config)#router pim
Brocade(config)#int e 3
Brocade(config-if-e1000-3)#ip address 207.95.5.1/24
Brocade(config-if-e1000-3)#ip pim
```

Syntax: [no] ip pim [version 1 | 2]

The **version 1 | 2** parameter specifies the PIM DM version. The default version is 2.

If you have enabled PIM version 1 but need to enable version 2 instead, enter either of the following commands at the configuration level for the interface.

```
Brocade(config-if-1/1)#ip pim version 2
Brocade(config-if-1/1)#no ip pim version 1
```

To disable PIM DM on the interface, enter the following command.

```
Brocade(config-if-1/1)#no ip pim
```

Modifying PIM global parameters

PIM global parameters come with preset values. The defaults work well in most networks, but you can modify the following parameters if you need to:

- Neighbor timeout
- Hello timer
- Prune timer
- Prune wait timer
- Graft retransmit timer
- Inactivity timer

Modifying neighbor timeout

Neighbor timeout is the interval after which a PIM router will consider a neighbor to be absent. Absence of PIM hello messages from a neighboring router indicates that a neighbor is not present.

The default value is 180 seconds.

To apply a PIM neighbor timeout value of 360 seconds to all ports on the router operating with PIM, enter the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#nbr-timeout 360
```

Syntax: **nbr-timeout** <60-8000>

The default is 180 seconds.

Modifying hello timer

This parameter defines the interval at which periodic hellos are sent out PIM interfaces. Routers use hello messages to inform neighboring routers of their presence. The default rate is 60 seconds.

To apply a PIM hello timer of 120 seconds to all ports on the router operating with PIM, enter the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#hello-timer 120
```

Syntax: **hello-timer** <10-3600>

The default is 60 seconds.

Modifying prune timer

This parameter defines how long a Brocade PIM router will maintain a prune state for a forwarding entry.

The first received multicast interface is forwarded to all other PIM interfaces on the router. If there is no presence of groups on that interface, the leaf node sends a prune message upstream and stores a prune state. This prune state travels up the tree and installs a prune state.

A prune state is maintained until the prune timer expires or a graft message is received for the forwarding entry. The default value is 180 seconds.

To set the PIM prune timer to 90, enter the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)##prune-timer 90
```

Syntax: **prune-timer** <10-3600>

The default is 180 seconds.

Modifying the prune wait timer

The CLI command **prune-wait** allows you to configure the amount of time a PIM router will wait before stopping traffic to neighbor routers that do not want the traffic. The value can be from zero to three seconds. The default is three seconds. A smaller prune wait value reduces flooding of unwanted traffic.

A prune wait value of zero causes the PIM router to stop traffic immediately upon receiving a prune message. If there are two or more neighbors on the physical port, then the **prune-wait** command should not be used because one neighbor may send a prune message while the other sends a join message at the during time or in less than three seconds.

To set the prune wait time to zero, enter the following commands.

```
Brocade(config)#router pim
Brocade(config-pim-router)#prune-wait 0
```

Syntax: `prune-wait <time>`

where <time> can be 0 - 3 seconds. A value of 0 causes the PIM router to stop traffic immediately upon receiving a prune message. The default is 3 seconds.

Viewing the prune wait time

To view the prune wait time, enter the **show ip pim dense** command at any level of the CLI. Refer to [“Displaying basic PIM Dense configuration information”](#) on page 1530.

Modifying graft retransmit timer

The Graft Retransmit Timer defines the interval between the transmission of graft messages.

A graft message is sent by a router to cancel a prune state. When a router receives a graft message, the router responds with a Graft Ack (acknowledge) message. If this Graft Ack message is lost, the router that sent the graft message will resend it.

To change the graft retransmit timer from the default of 180 to 90 seconds, enter the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#graft-retransmit-timer 10
```

Syntax: `graft-retransmit-timer <2 -10>`

The default is 3 seconds.

Modifying inactivity timer

The router deletes a forwarding entry if the entry is not used to send multicast packets. The PIM inactivity timer defines how long a forwarding entry can remain unused before the router deletes it.

To apply a PIM inactivity timer of 90 seconds to all PIM interfaces, enter the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#inactivity-timer 90
```

Syntax: `inactivity-timer <10-3600>`

The default is 180 seconds.

Selection of shortest path back to source

By default, when a multicast packet is received on a PIM-capable router interface in a multi-path topology, the interface checks its IP routing table to determine the shortest path back to the source. If the alternate paths have the same cost, the first alternate path in the table is picked as the path back to the source. For example, in the table below, the first four routes have the same cost back to the source. However, 137.80.127.3 will be chosen as the path to the source since it is the first one on the list. The router rejects traffic from any port other than Port V11 on which 137.80.127.3 resides.

Total number of IP routes: 19

	B:BGP	D:Connected	R:RIP	S:Static	O:OSPF	*:Candidate	default		
		Destination		NetMask		Gateway	Port	Cost	Type
9		172.17.41.4		255.255.255.252	*	137.80.127.3	v11	2	O
		172.17.41.4		255.255.255.252		137.80.126.3	v10	2	O
		172.17.41.4		255.255.255.252		137.80.129.1	v13	2	O
		172.17.41.4		255.255.255.252		137.80.128.3	v12	2	O
10		172.17.41.8		255.255.255.252	0.0.0.0		1/2	1	D

Due to the Highest IP RFP feature, the selection of the shortest path back to the source is based on which Reverse Path Forwarding (RPF) neighbor in the IP routing table has the highest IP address, if the cost of the routes are the same. For example, in the table above, Gateway 137.80.129.1 will be chosen as the shortest path to the source because it is the RPF neighbor with the highest IP address.

When choosing the RPF, the router first checks the Multicast Routing Table. If the table is not available, it chooses an RPF from the IP Routing Table. Multicast route is configured using the **ip mroute** command.

The Highest IP RFP feature is enabled by default.

Failover time in a multi-path topology

When a port in a multi-path topology fails, and the failed port is the input port of the downstream router, a new path is re-established within a few seconds, depending on the routing protocol being used.

No configuration is required for this feature.

Modifying the TTL

The time to live (TTL) defines the minimum value required in a packet for it to be forwarded out of the interface.

For example, if the TTL for an interface is set at 10, it means that only those packets with a TTL value of 10 or more will be forwarded. Likewise, if an interface is configured with a TTL Threshold value of 1, all packets received on that interface will be forwarded. Possible TTL values are 1 to 31. The default TTL value is 1.

Configuration notes for modifying the TTL

- If the TTL for an interface is greater than 1, PIM packets received on the interface are always forwarded in software because each packet TTL must be examined. Therefore, Brocade does not recommend modifying the TTL under normal operating conditions.
- Multicast packets with a TTL value of 1 are switched within the same VLAN. These packets cannot be routed between different VLANs.

Configuration syntax for modifying the TTL

To configure a TTL of 24, enter the following.

```
Brocade(config-if-3/24)#ip pim ttl-threshold 24
```

Syntax: `ip pim ttl-threshold <1-31>`

Dropping PIM traffic in hardware

Unwanted PIM Dense or PIM Sparse multicast traffic can be dropped in hardware on Layer 3 switches. This feature does not apply to DVMRP traffic. Refer to [“Passive multicast route insertion”](#) on page 1579.

Displaying basic PIM Dense configuration information

To display PIM Dense configuration information, enter the **show ip pim dense** command at any CLI level.

```
router(config)# show ip pim dense
Global PIM Dense Mode Settings
Hello interval: 60, Neighbor timeout: 180
Graft Retransmit interval: 3, Inactivity interval: 180
Route Expire interval: 0, Route Discard interval: 0
Prune age: 180, Prune wait: 3

Interface v5
PIM Dense: V2
TTL Threshold: 1, Disabled, DR: itself
Local Address: 209.157.23.1
```

Syntax: `show ip pim dense`

[Table 254](#) shows the information displayed by the **show ip pim dense** command.

TABLE 254 Output of the **show ip pim dense** command

Field	Description
Hello interval	How frequently the device sends hello messages out the PIM dense interfaces.
Neighbor timeout	The interval after which a PIM device will consider a neighbor to be absent.
Graft or Retransmit interval	How interval between the transmission of graft messages.
Inactivity interval	How long a forwarding entry can remain unused before the device deletes it.
Route Expiry Interval	How long a route is considered valid in the absence of the next route update
Route Discard Interval	Defines the period of time before a route is deleted.
Join or Prune interval	How long a PIM device will maintain a prune state for a forwarding entry.

TABLE 254 Output of the **show ip pim dense** command

Field	Description
Prune Age	The number of packets the device sends using the path through the RP before switching to using the SPT path.
Prune Wait Interval	The amount of time a PIM device waits before stopping traffic to neighbor devices that do not want the traffic. The value can be from zero to three seconds. The default is three seconds.
Interface	The type of interface and the interface number.
Local Address	Indicates the IP address configured on the port or virtual interface.
TTL Threshold	The minimum value required in a packet for it to be forwarded out of the interface.
DR	The priority of the designated device.

Displaying all multicast cache entries in a pruned state

You can use the **show ip pim prune** command to display all multicast cache entries that are currently in a pruned state and have not yet aged out.

```
router(config)# show ip pim prune
Port    SourceNet    Group          Nbr    PhyPort    Age
                                     sec
v800 172.10.1.10  239.255.255.250 0.0.0.0  1/1/3     140
```

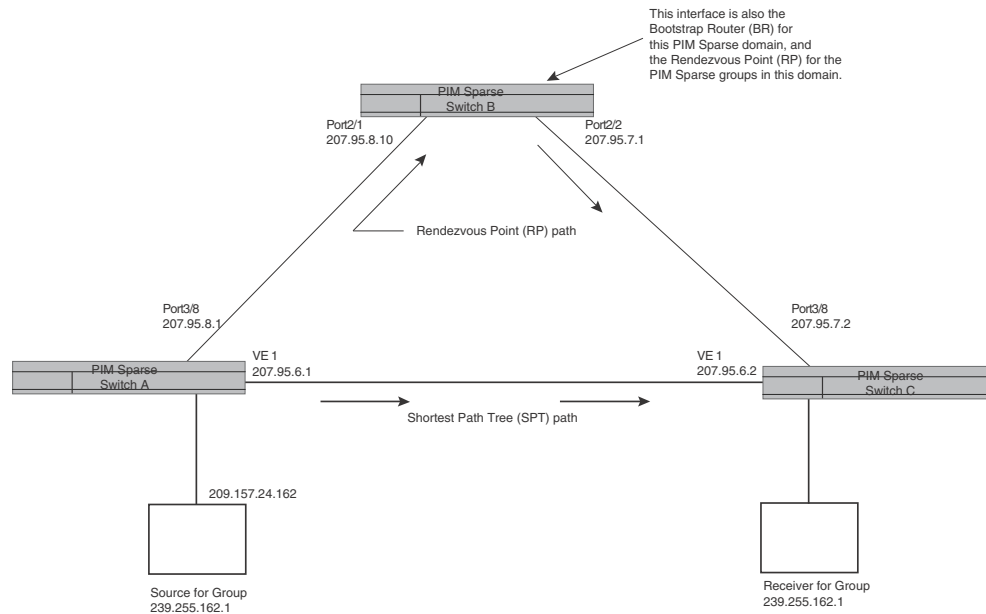
Syntax: **show ip pim prune**

PIM Sparse

Brocade devices support Protocol Independent Multicast (PIM) Sparse version 2. PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments. The Brocade implementation is based on RFC 2362.

In a PIM Sparse network, a PIM Sparse router that is connected to a host that wants to receive information for a multicast group must explicitly send a join request on behalf of the receiver (host).

PIM Sparse routers are organized into domains. A PIM Sparse domain is a contiguous set of routers that all implement PIM and are configured to operate within a common boundary. [Figure 174](#) shows a simple example of a PIM Sparse domain. This example shows three Layer 3 switches configured as PIM Sparse routers. The configuration is described in detail following the figure.

FIGURE 174 Example of a PIM Sparse domain

PIM Sparse switch types

Switches that are configured with PIM Sparse interfaces also can be configured to fill one or more of the following roles:

- **PMBR** – A PIM switch that has some interfaces within the PIM domain and other interface outside the PIM domain. PMBRs connect the PIM domain to the Internet.

NOTE

You cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse in the current software release.

- **BSR** – The Bootstrap Router (BSR) distributes RP information to the other PIM Sparse switches within the domain. Each PIM Sparse domain has one active BSR. For redundancy, you can configure ports on multiple switches as candidate BSRs. The PIM Sparse protocol uses an election process to select one of the candidate BSRs as the BSR for the domain. The BSR with the highest BSR priority (a user-configurable parameter) is elected. If the priorities result in a tie, then the candidate BSR interface with the highest IP address is elected. In the example in [Figure 174](#), PIM Sparse switch B is the BSR. Port 2/2 is configured as a candidate BSR.
- **RP** – The RP is the meeting point for PIM Sparse sources and receivers. A PIM Sparse domain can have multiple RPs, but each PIM Sparse multicast group address can have only one active RP. PIM Sparse switches learn the addresses of RPs and the groups for which they are responsible from messages that the BSR sends to each of the PIM Sparse switches. In the example in [Figure 174](#), PIM Sparse Switch B is the RP. Port 2/2 is configured as a candidate Rendezvous Point (RP).

To enhance overall network performance, Brocade Layer 3 switches use the RP to forward only the first packet from a group source to the group receivers. After the first packet, the Layer 3 switch calculates the shortest path between the receiver and source (the Shortest Path Tree, or SPT) and uses the SPT for subsequent packets from the source to the receiver. The Layer 3 switch calculates a separate SPT for each source-receiver pair.

NOTE

Brocade recommends that you configure the same ports as candidate BSRs and RPs.

RP paths and SPT paths

[Figure 174](#) shows two paths for packets from the source for group 239.255.162.1 and a receiver for the group. The source is attached to PIM Sparse Switch A and the recipient is attached to PIM Sparse Switch C. PIM Sparse Switch B is the RP for this multicast group. As a result, the default path for packets from the source to the receiver is through the RP. However, the path through the RP sometimes is not the shortest path. In this case, the shortest path between the source and the receiver is over the direct link between Switch A and Switch C, which bypasses the RP (Switch B).

To optimize PIM traffic, the protocol contains a mechanism for calculating the Shortest Path Tree (SPT) between a given source and receiver. PIM Sparse switches can use the SPT as an alternative to using the RP for forwarding traffic from a source to a receiver. By default, Brocade Layer 3 switches forward the first packet they receive from a given source to a given receiver using the RP path, but forward subsequent packets from that source to that receiver through the SPT. In [Figure 174](#), Switch A forwards the first packet from group 239.255.162.1 source to the destination by sending the packet to Switch B, which is the RP. Switch B then sends the packet to Switch C. For the second and all future packets that Switch A receives from the source for the receiver, Switch A forwards them directly to Switch C using the SPT path.

PIM Sparse configuration

To configure a Brocade Layer 3 switch for PIM Sparse, perform the following tasks:

- Configure the following global parameter:
 - Enable the PIM Sparse mode of multicast routing.
- Configure the following interface parameters:
 - Configure an IP address on the interface
 - Enable PIM Sparse.
 - Identify the interface as a PIM Sparse border, if applicable.

NOTE

You cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse in the current software release.

- Configure the following PIM Sparse global parameters:
 - Identify the Layer 3 switch as a candidate PIM Sparse Bootstrap Router (BSR), if applicable.
 - Identify the Layer 3 switch as a candidate PIM Sparse Rendezvous Point (RP), if applicable.
 - Specify the IP address of the RP (if you want to statically select the RP).

NOTE

Brocade recommends that you configure the same Layer 3 switch as both the BSR and the RP.

PIM Sparse limitations in this release

The implementation of PIM Sparse in the current software release has the following limitations:

- PIM Border Routers (PMBRs) are not supported. Thus, you cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse.
- PIM Sparse and regular PIM (dense mode) cannot be used on the same interface.
- You cannot configure or display PIM Sparse information using the Web Management Interface. (You can display some general PIM information, but not specific PIM Sparse information.)

Configuring Global PIM Sparse parameters

To configure the PIM Sparse global parameters, use either of the following methods.

To configure basic global PIM Sparse parameters, enter commands such as the following on each Layer 3 switch within the PIM Sparse domain.

```
Brocade(config)#router pim
```

Syntax: [no] router pim

NOTE

You do not need to globally enable IP multicast routing when configuring PIM Sparse.

The command in this example enables IP multicast routing, and enables the PIM Sparse mode of IP multicast routing. The command does not configure the Layer 3 switch as a candidate PIM Sparse Bootstrap Router (BSR) and candidate Rendezvous Point (RP). You can configure a Brocade Layer 3 switch as a PIM Sparse switch without configuring the it as a candidate BSR and RP. However, if you do configure the Layer 3 switch as one of these, Brocade recommends that you configure it as both. Refer to [“Configuring BSRs”](#) on page 1535.

The behavior of the [no] router pim command is as follows:

- Entering **no router pim** command to disable PIM or DVMRP does not require a software reload.
- Entering a **no router pim** command removes all configuration for PIM multicast on a Layer 3 switch (**router pim** level) only.

Globally enabling and disabling PIM without deleting the multicast configuration

As stated above entering a **no router pim** command deletes the PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command.

```
Brocade(config)#router pim
Brocade(config-pim-router)#disable-pim
```

Syntax: [no] disable-pim

Use the [no] version of the command to re-enable PIM.

Configuring PIM interface parameters

After you enable IP multicast routing and PIM Sparse at the global level, you must enable it on the individual interfaces connected to the PIM Sparse network. To do so, use the following CLI method.

To enable PIM Sparse mode on an interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 2/2
Brocade(config-if-2/2)#ip address 207.95.7.1 255.255.255.0
Brocade(config-if-2/2)#ip pim-sparse
```

Syntax: [no] ip pim-sparse

The commands in this example add an IP interface to port 2/2, then enable PIM Sparse on the interface.

If the interface is on the border of the PIM Sparse domain, you also must enter the following command.

```
Brocade(config-if-2/2)#ip pim border
```

Syntax: [no] ip pim border

NOTE

You cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse in the current software release.

Configuring BSRs

In addition to the global and interface parameters in the sections above, you need to identify an interface on at least one Layer 3 switch as a candidate PIM Sparse boot strap router (BSR) and candidate PIM Sparse Rendezvous Point (RP).

NOTE

It is possible to configure the Layer 3 switch as only a candidate BSR or RP, but Brocade recommends that you configure the same interface on the same Layer 3 switch as both a BSR and an RP.

This section presents how to configure BSRs. Refer to [“Configuring RPs”](#) on page 1536 for instructions on how to configure RPs.

To configure the Layer 3 switch as a candidate BSR and RP, enter commands such as the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#bsr-candidate ethernet 2/2 30 255
BSR address: 207.95.7.1, hash mask length: 30, priority: 255
```

This command configures the PIM Sparse interface on port 2/2 as a BSR candidate, with a hash mask length of 30 and a priority of 255. The information shown in italics above is displayed by the CLI after you enter the candidate BSR configuration command.

Syntax: [no] bsr-candidate ethernet [<slotnum>]/<portnum> | loopback <num> | ve <num> <hash-mask-length> [<priority>]

The <slotnum> parameter is required on chassis devices.

The <portnum> | loopback <num> | ve <num> parameter specifies the interface. The Layer 3 switch will advertise the specified interface IP address as a candidate BSR:

- Enter **ethernet** [*<slotnum>/*] *<portnum>* for a physical interface (port).
- Enter **ve** *<num>* for a virtual interface.
- Enter **loopback** *<num>* for a loopback interface.

The *<hash-mask-length>* parameter specifies the number of bits in a group address that are significant when calculating the group-to-RP mapping. You can specify a value from 1 – 32.

NOTE

Brocade recommends you specify 30 for IP version 4 (IPv4) networks.

The *<priority>* specifies the BSR priority. You can specify a value from 0 – 255. When the election process for BSR takes place, the candidate BSR with the highest priority becomes the BSR. The default is 0.

Enter the command `show ip pim bsr` to display BSR information. Refer to [“Displaying BSR information”](#) on page 1547.

Configuring RPs

Enter a command such as the following to configure the Layer 3 switch as a candidate rendezvous point (RP).

```
Brocade(config-pim-router)#rp-candidate ethernet 2/2
```

Syntax: `[no] rp-candidate ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>`

The *<slotnum>* parameter is required on chassis devices.

The *<portnum> | loopback <num> | ve <num>* parameter specifies the interface. The Layer 3 switch will advertise the specified interface IP address as a candidate RP:

- Enter **ethernet** [*<slotnum>/*] *<portnum>* for a physical interface (port).
- Enter **ve** *<num>* for a virtual interface.
- Enter **loopback** *<num>* for a loopback interface.

By default, this command configures the Layer 3 switch as a candidate RP for all group numbers beginning with 224. As a result, the Layer 3 switch is a candidate RP for all valid PIM Sparse group numbers. You can change this by adding or deleting specific address ranges. The following example narrows the group number range for which the Layer 3 switch is a candidate RP by explicitly adding a range.

```
Brocade(config-pim-router)#rp-candidate add 224.126.0.0 16
```

Syntax: `[no] rp-candidate add <group-addr> <mask-bits>`

The *<group-addr> <mask-bits>* specifies the group address and the number of significant bits in the subnet mask. In this example, the Layer 3 switch is a candidate RP for all groups that begin with 224.126. When you add a range, you override the default. The Layer 3 switch then becomes a candidate RP only for the group address ranges you add.

You also can change the group numbers for which the Layer 3 switch is a candidate RP by deleting address ranges. For example, to delete all addresses from 224.126.22.0 – 224.126.22.255, enter the **rp-candidate delete** command.

```
Brocade(config-pim-router)#rp-candidate delete 224.126.22.0 24
```

Syntax: `rp-candidate delete <group-addr> <mask-bits>`

The usage of the `<group-addr> <mask-bits>` parameter is the same as for the **rp-candidate add** command.

If you enter both commands shown in the example above, the net effect is that the Layer 3 switch becomes a candidate RP for groups 224.126.0.0 – 224.126.21.255 and groups 224.126.23.0 – 224.126.255.255.

Updating PIM-Sparse forwarding entries with new RP configuration

If you make changes to your static RP configuration, the entries in the PIM-Sparse multicast forwarding table continue to use the old RP configuration until they are aged out.

The **clear pim rp-map** command allows you to update the entries in the static multicast forwarding table immediately after making RP configuration changes. This command is meant to be used with **rp-address** command.

To update the entries in a PIM sparse static multicast forwarding table with new RP configuration, enter the following command at the privileged EXEC level of the CLI.

```
Brocade#clear pim rp-map
```

Syntax: **clear pim rp-map**

Statically specifying the RP

Brocade recommends that you use the PIM Sparse protocol RP election process so that a backup RP can automatically take over if the active RP router becomes unavailable. However, if you do not want the RP to be selected by the RP election process but instead you want to explicitly identify the RP by its IP address, you can do using the following CLI method.

Even if you explicitly specify the RP, by default the Layer 3 switch uses the set of candidate RPs supplied by the BSR and overrides the specified RP for all group-to-RP mappings.

NOTE

Specify the same IP address as the RP on all PIM Sparse routers within the PIM Sparse domain. Make sure the router is on the backbone or is otherwise well connected to the rest of the network.

To specify the IP address of the RP, enter commands such as the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#rp-address 207.95.7.1
```

Syntax: **[no] rp-address <ip-addr>**

The `<ip-addr>` parameter specifies the IP address of the RP.

The command in the example above identifies the router interface at IP address 207.95.7.1 as the RP for the PIM Sparse domain. The Layer 3 switch will use the specified RP and ignore group-to-RP mappings received from the BSR.

Changing the Shortest Path Tree (SPT) threshold

In a typical PIM Sparse domain, there may be two or more paths from a DR (designated router) for a multicast source to a PIM group receiver:

- **Path through the RP** – This is the path the Layer 3 switch uses the first time it receives traffic for a PIM group. However, the path through the RP may not be the shortest path from the Layer 3 switch to the receiver.

- **Shortest Path** – Each PIM Sparse router that is a DR for a multicast source calculates a shortest path tree (SPT) to all the PIM Sparse group receivers within the domain, with the Layer 3 switch itself as the root of the tree. The first time a Brocade Layer 3 switch configured as a PIM router receives a packet for a PIM receiver, the Layer 3 switch sends the packet to the RP for the group. The Layer 3 switch also calculates the SPT from itself to the receiver. The next time the Layer 3 switch receives a PIM Sparse packet for the receiver, the Layer 3 switch sends the packet toward the receiver using the shortest route, which may not pass through the RP.

By default, the device switches from the RP to the SPT after receiving the first packet for a given PIM Sparse group. The Layer 3 switch maintains a separate counter for each PIM Sparse source-group pair.

After the Layer 3 switch receives a packet for a given source-group pair, the Layer 3 switch starts a PIM data timer for that source-group pair. If the Layer 3 switch does not receive another packet for the source-group pair before the timer expires, it reverts to using the RP for the next packet received for the source-group pair. In accordance with the PIM Sparse RFC recommendation, the timer is 210 seconds and is not configurable. The counter is reset to zero each time the Layer 3 switch receives a packet for the source-group pair.

You can change the number of packets that the Layer 3 switch sends using the RP before switching to using the SPT. To do so, use the following CLI method.

```
Brocade(config)#router pim
Brocade(config-pim-router)#spt-threshold 1000
```

Syntax: [no] spt-threshold infinity | <num>

The **infinity** | <num> parameter specifies the number of packets. If you specify infinity, the Layer 3 switch sends packets using the RP indefinitely and does not switch over to the SPT. If you enter a specific number of packets, the Layer 3 switch does not switch over to using the SPT until it has sent the number of packets you specify using the RP.

NOTE

Since the hardware is not programmed for RPT, the rate that the client receives multicast traffic depends on the software forwarding capability of the CPU when the Layer 3 Switch sends packets using the RPT.

Changing the PIM join and prune message interval

By default, the Layer 3 switch sends PIM Sparse Join/Prune messages every 60 seconds. These messages inform other PIM Sparse routers about clients who want to become receivers (Join) or stop being receivers (Prune) for PIM Sparse groups.

You can change the Join/Prune message interval using the following CLI method.

NOTE

Use the same Join/Prune message interval on all the PIM Sparse routers in the PIM Sparse domain. If the routers do not all use the same timer interval, the performance of PIM Sparse can be adversely affected.

To change the Join/Prune interval, enter commands such as the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#message-interval 30
```

Syntax: [no] message-interval <num>

The `<num>` parameter specifies the number of seconds and can range from 1 – 65535. The default is 60.

Dropping PIM traffic in hardware

Unwanted PIM Dense or PIM Sparse multicast traffic can be dropped in hardware on Layer 3 switches. This feature does not apply to DVMRP. Refer to [“Passive multicast route insertion”](#) on page 1579.

ACL based RP assignment

The **rp-address** command allows multiple static RP configurations. For each static RP, an ACL can be given as an option to define the multicast address ranges that the static RP permit or deny to serve.

A static RP by default serves the range of 224.0.0.0/4 if the RP is configured without an ACL name. If an ACL name is given but the ACL is not defined, the static RP is set to inactive mode and it will not cover any multicast group ranges.

The optional static RP ACL can be configured as a standard ACL or as an extended ACL. For an extended ACL, the destination filter will be used to derive the multicast group range and all other filters are ignored. The content of the ACL needs to be defined in the order of prefix length; the longest prefix must be placed at the top of the ACL definition.

If there are overlapping group ranges among the static RPs, the static RP with the longest prefix match is selected. If more than one static RP covers the exact same group range, the highest IP static RP will be used.

Configuration considerations for assigning ACL-based RP

- The RP learned from the BSR has higher precedence over Static RP. For more information, refer to the definition of the `override` parameter in the section [“Using ACLs to limit static RP groups”](#) on page 1588
- There is a limit of 16 static RPs in the systems.

Configuring an ACL based RP assignment

To configure an ACL based RP assignment, enter commands such as the following.

```
Brocade(config)# router pim
Brocade(config-pim-router)# rp-address 130.1.1.1 20
```

Syntax: `[no] rp-address <ip_address> [<acl-num>]`

Use the **ip address** parameter to specify the IP address of the device you want to designate as an RP device.

Use the **acl-num** parameter to specify standard numbered ACL that specifies which multicast groups use this RP.

NOTE

Only standard numbered ACLs are supported when configuring an ACL based RP assignment.

Displaying the static RP

Use the **show ip pim rp-set** command to display static RP and the associated group ranges.

```
Brocade(config)# show ip pim rp-set
Static RP and associated group ranges
-----
Static RP count: 4
130.1.1.1
120.1.1.1
120.2.1.1
124.1.1.1
Number of group prefixes Learnt from BSR: 0
No RP-Set present.
```

Use the **show ip pim rp-map** command to display all current multicast group addresses to RP address mapping.

```
Brocade(config)# show ip pim rp-map
Number of group-to-RP mappings: 5
      Group address      RP address
-----
1  230.0.0.1             100.1.1.1
2  230.0.0.2             100.1.1.1
3  230.0.0.3             100.1.1.1
4  230.0.0.4             100.1.1.1
5  230.0.0.5             100.1.1.1
```

Anycast RP

Anycast RP is a method of providing intra-domain redundancy and load-balancing between multiple Rendezvous Points (RP) in a Protocol Independent Multicast Sparse mode (PIM-SM) network. It is accomplished by configuring all RPs within a domain with the same anycast RP address which is typically a loopback IP address. Multicast Source Discovery Protocol (MSDP) is used between all of the RPs in a mesh configuration to keep all RPs in sync regarding the active sources.

PIM-SM routers are configured to register (statically or dynamically) with the RP using the same anycast RP address. Since multiple RPs have the same anycast address, an Interior Gateway Protocol (IGP) such as OSPF routes the PIM-SM router to the RP with the best route. If the PIM-SM routers are distributed evenly throughout the domain, the loads on RPs within the domain will be distributed. If the RP with the best route goes out of service, the PIM-SM router IGP changes the route to the closest operating RP that has the same anycast address.

This configuration works because MSDP is configured between all of the RPs in the domain. Consequently, all of the RPs share information about active sources.

Configuring Anycast RP

To configure Anycast RP, you must do the following:

- Configure a loopback interface with the anycast RP address on each of the RPs within the domain and enable PIM-SM on these interfaces.
- Ensure that the anycast RP address is leaked into the IGP domain. This is typically done by enabling the IGP on the loopback interface (in passive mode) or redistributing the connected loopback IP address into the IGP.

NOTE

The anycast RP address must not be the IGP router-id.

- Enable PIM-SM on all interfaces on which multicast routing is desired.
- Enable an IGP on each of the loopback interfaces and physical interfaces configured for PIM-SM.
- Configure loopback interfaces with unique IP addresses on each of the RPs for MSDP peering. This loopback interface is also used as the MSDP originator-id.
- The non-RP PIM-SM routers may be configured to use the anycast RP address statically or dynamically (by the PIMv2 bootstrap mechanism).

Example of a simple Anycast-enabled network

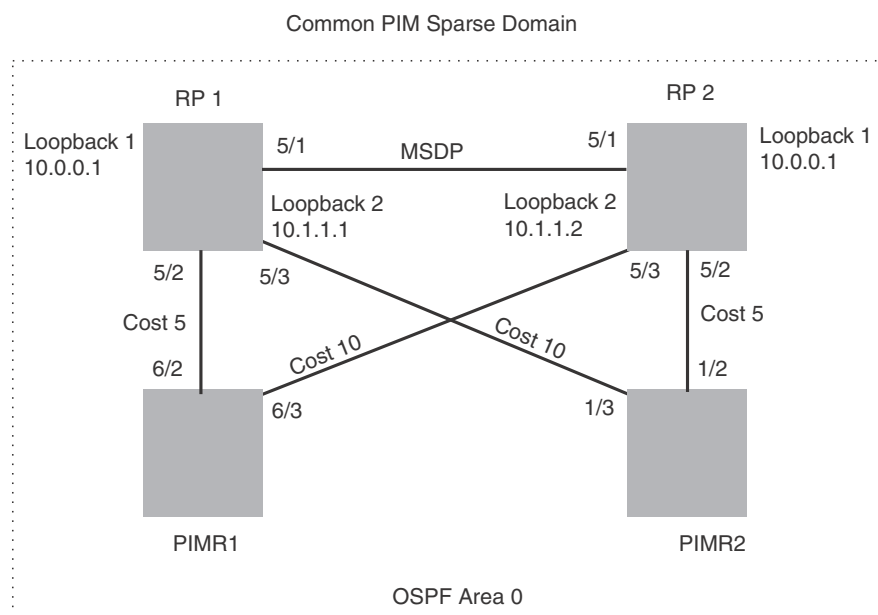
The example shown in Figure 175 is a simple Anycast-enabled network with two RPs and two PIM-SM routers. Loopback 1 in RP 1 and RP 2 have the same IP address. Loopback 2 in RP1 and Loopback 2 in RP2 have different IP addresses and are configured as MSDP peering IP addresses in a mesh configuration.

In the PIM configuration for PIM-SM routers PIMR1 and PIMR2 the RP address is configured to be the anycast RP address that was configured on the Loopback 1 interfaces on RP1 and RP2. OSPF is configured as the IGP for the network and all of the devices are in OSPF area 0.

Since PIMR1 has a lower cost path to RP1 and PIMR2 has a lower cost path to RP2 they will register with the respective RPs when both are up and running. This shares the load between the two RPs. If one of the RPs fails, the higher-cost path to the IP address of Loopback 1 on the RPs is used to route to the still-active RP.

The configuration examples demonstrate the commands required to enable this application.

FIGURE 175 Example of an Anycast RP network



RP 1 configuration

The following commands provide the configuration for the RP 1 router in [Figure 175](#).

```

Brocade RP1(config)#router ospf
Brocade RP1(config-ospf-router)# area 0
Brocade RP1(config-ospf-router)# exit
Brocade RP1(config)# interface loopback 1
Brocade RP1(config-lbif-1)# ip ospf area 0
Brocade RP1(config-lbif-1)# ip ospf passive
Brocade RP1(config-lbif-1)# ip address 10.0.0.1/32
Brocade RP1(config-lbif-1)# ip pim-sparse
Brocade RP1(config-lbif-1)# exit
Brocade RP1(config)# interface loopback 2
Brocade RP1(config-lbif-2)# ip ospf area 0
Brocade RP1(config-lbif-2)# ip ospf passive
Brocade RP1(config-lbif-2)# ip address 10.1.1.1/32
Brocade RP1(config-lbif-2)# exit
Brocade RP1(config)# interface ethernet 5/1
Brocade RP1(config-if-e1000-5/1)# ip ospf area 0
Brocade RP1(config-if-e1000-5/1)# ip address 192.1.1.1/24
Brocade RP1(config-if-e1000-5/1)# ip pim-sparse
Brocade RP1(config)# interface ethernet 5/2
Brocade RP1(config-if-e1000-5/2)# ip ospf area 0
Brocade RP1(config-if-e1000-5/2)# ip ospf cost 5
Brocade RP1(config-if-e1000-5/2)# ip address 192.2.1.1/24
Brocade RP1(config-if-e1000-5/2)# ip pim-sparse
Brocade RP1(config)# interface ethernet 5/3
Brocade RP1(config-if-e1000-5/3)# ip ospf area 0
Brocade RP1(config-if-e1000-5/3)# ip ospf cost 10
Brocade RP1(config-if-e1000-5/3)# ip address 192.3.1.1/24
Brocade RP1(config-if-e1000-5/3)# ip pim-sparse
Brocade RP1(config-if-e1000-5/3)# exit
Brocade RP1(config)# router pim
Brocade RP1(config-pim-router)# rp-candidate loopback 1
Brocade RP1(config-pim-router)# exit
Brocade RP1(config)# router msdp
Brocade RP1(config-msdp-router)# msdp-peer 10.1.1.2 connect-source loopback 2
Brocade RP1(config-msdp-router)# originator-id loopback 2

```

RP 2 configuration

The following commands provide the configuration for the RP 2 router in [Figure 175](#).

```

Brocade RP2(config)#router ospf
Brocade RP2(config-ospf-router)# area 0
Brocade RP2(config-ospf-router)# exit
Brocade RP2(config)# interface loopback 1
Brocade RP2(config-lbif-1)# ip ospf area 0
Brocade RP2(config-lbif-1)# ip ospf passive
Brocade RP2(config-lbif-1)# ip address 10.0.0.1/32
Brocade RP2(config-lbif-1)# ip pim-sparse
Brocade RP2(config-lbif-1)# exit
Brocade RP2(config)# interface loopback 2
Brocade RP2(config-lbif-2)# ip ospf area 0
Brocade RP2(config-lbif-2)# ip ospf passive
Brocade RP2(config-lbif-2)# ip address 10.1.1.2/32
Brocade RP2(config-lbif-2)# exit
Brocade RP2(config)# interface ethernet 5/1
Brocade RP2(config-if-e1000-5/1)# ip ospf area 0
Brocade RP2(config-if-e1000-5/1)# ip address 192.1.1.2/24

```

```

Brocade RP2(config-if-e1000-5/1)# ip pim-sparse
Brocade RP2(config)# interface ethernet 5/2
Brocade RP2(config-if-e1000-5/2)# ip ospf area 0
Brocade RP2(config-if-e1000-5/2)# ip ospf cost 5
Brocade RP2(config-if-e1000-5/2)# ip address 192.5.2.1/24
Brocade RP2(config-if-e1000-5/2)# ip pim-sparse
Brocade RP2(config)# interface ethernet 5/3
Brocade RP2(config-if-e1000-5/3)# ip ospf area 0
Brocade RP2(config-if-e1000-5/3)# ip ospf cost 10
Brocade RP2(config-if-e1000-5/3)# ip address 192.6.1.2/24
Brocade RP2(config-if-e1000-5/3)# ip pim-sparse
Brocade RP2(config-if-e1000-5/3)# exit
Brocade RP2(config)# router pim
Brocade RP2(config-pim-router)# rp-candidate loopback 1
Brocade RP2(config-pim-router)# exit
Brocade RP2(config)# router msdp
Brocade RP2(config-msdp-router)# msdp-peer 10.1.1.1 connect-source loopback 2
Brocade RP2(config-msdp-router)# originator-id loopback 2

```

PIMR1configuration

The following commands provide the configuration for the PIMR1 router in [Figure 175](#).

```

Brocade PIMR1(config)#router ospf
Brocade PIMR1(config-ospf-router)# area 0
Brocade PIMR1(config-ospf-router)# exit
Brocade PIMR1(config)# interface ethernet 6/2
Brocade PIMR1(config-if-e1000-6/2)# ip ospf area 0
Brocade PIMR1(config-if-e1000-6/2)# ip ospf cost 5
Brocade PIMR1(config-if-e1000-6/2)# ip address 192.2.1.2/24
Brocade PIMR1(config-if-e1000-6/2)# ip pim-sparse
Brocade PIMR1(config)# interface ethernet 6/3
Brocade PIMR1(config-if-e1000-6/3)# ip ospf area 0
Brocade PIMR1(config-if-e1000-6/3)# ip ospf cost 10
Brocade PIMR1(config-if-e1000-6/3)# ip address 192.6.1.1/24
Brocade PIMR1(config-if-e1000-6/3)# ip pim-sparse
Brocade PIMR1(config-if-e1000-6/3)# exit
Brocade PIMR1(config)# router pim
Brocade PIMR1(config-pim-router)# rp-address 10.0.0.1
Brocade PIMR1(config-pim-router)# exit

```

PIMR2 configuration

The following commands provide the configuration for the PIMR2 router in [Figure 175](#).

```

Brocade PIMR2(config)#router ospf
Brocade PIMR2(config-ospf-router)# area 0
Brocade PIMR2(config-ospf-router)# exit
Brocade PIMR2(config)# interface ethernet 1/2
Brocade PIMR2(config-if-e1000-1/2)# ip ospf area 0
Brocade PIMR2(config-if-e1000-1/2)# ip ospf cost 5
Brocade PIMR2(config-if-e1000-1/2)# ip address 192.5.2.2/24
Brocade PIMR2(config-if-e1000-1/2)# ip pim-sparse
Brocade PIMR2(config)# interface ethernet 1/3
Brocade PIMR2(config-if-e1000-1/3)# ip ospf area 0
Brocade PIMR2(config-if-e1000-1/3)# ip ospf cost 10
Brocade PIMR2(config-if-e1000-1/3)# ip address 192.3.1.2/24
Brocade PIMR2(config-if-e1000-1/3)# ip pim-sparse

```

PIM Sparse

```
Brocade PIMR2(config-if-e1000-1/3)# exit
Brocade PIMR2(config)# router pim
Brocade PIMR2(config-pim-router)# rp-address 10.0.0.1
Brocade PIMR2(config-pim-router)# exit
```

Displaying PIM Sparse configuration information and statistics

You can display the following PIM Sparse information:

- Basic PIM Sparse configuration information
- Group information
- BSR information
- Candidate RP information
- RP-to-group mappings
- RP information for a PIM Sparse group
- RP set list
- PIM Neighbor information
- The PIM flow cache
- The PIM multicast cache
- PIM traffic statistics

Displaying basic PIM Sparse configuration information

To display basic configuration information for PIM Sparse, enter the **show ip pim sparse** command at any CLI level.

```
Brocade#show ip pim sparse
```

```
Global PIM Sparse Mode Settings
  Hello interval: 60, Neighbor timeout: 180
  Bootstrap Msg interval: 130, Candidate-RP Advertisement interval: 60
  Join/Prune interval: 60, SPT Threshold: 1
```

```
Interface e11/1
PIM Sparse
TTL Threshold: 1, Enabled, DR: 50.1.1.2 on e11/1
Local Address: 50.1.1.1
Neighbor:
  50.1.1.2
```

```
Interface v40
PIM Sparse
TTL Threshold: 1, Enabled, DR: itself
Local Address: 40.1.1.2
Neighbor:
  40.1.1.1
```

Syntax: show ip pim sparse

Table 255 shows the information displayed by the **show ip pim sparse** command.

TABLE 255 Output of the **show ip pim sparse** command

Field	Description
Global PIM Sparse mode settings	
Hello interval	How frequently the Layer 3 switch sends PIM Sparse hello messages to its PIM Sparse neighbors. This field show the number of seconds between hello messages. PIM Sparse routers use hello messages to discover one another.
Neighbor timeout	How many seconds the Layer 3 switch will wait for a hello message from a neighbor before determining that the neighbor is no longer present and removing cached PIM Sparse forwarding entries for the neighbor.
Bootstrap Msg interval	How frequently the BSR configured on the Layer 3 switch sends the RP set to the RPs within the PIM Sparse domain. The RP set is a list of candidate RPs and their group prefixes. A candidate RP group prefix indicates the range of PIM Sparse group numbers for which it can be an RP. NOTE: This field contains a value only if an interface on the Layer 3 switch is elected to be the BSR. Otherwise, the field is blank.
Candidate-RP Advertisement interval	How frequently the candidate PR configured on the Layer 3 switch sends candidate RP advertisement messages to the BSR. NOTE: This field contains a value only if an interface on the Layer 3 switch is configured as a candidate RP. Otherwise, the field is blank.
Join/Prune interval	How frequently the Layer 3 switch sends PIM Sparse Join/Prune messages for the multicast groups it is forwarding. This field show the number of seconds between Join/Prune messages. The Layer 3 switch sends Join/Prune messages on behalf of multicast receivers who want to join or leave a PIM Sparse group. When forwarding packets from PIM Sparse sources, the Layer 3 switch sends the packets only on the interfaces on which it has received join requests in Join/Prune messages for the source group. You can change the Join/Prune interval if needed. Refer to “Changing the PIM join and prune message interval” on page 1538.
SPT Threshold	The number of packets the Layer 3 switch sends using the path through the RP before switching to using the SPT path.
PIM Sparse interface information	
NOTE: You also can display IP multicast interface information using the show ip pim interface command. However, this command lists all IP multicast interfaces, including regular PIM (dense mode) and DVMRP interfaces. The show ip pim sparse command lists only the PIM Sparse interfaces.	
Interface	The type of interface and the interface number. The interface type can be one of the following: <ul style="list-style-type: none"> • Ethernet • VE The number is either a port number (and slot number if applicable) or the virtual interface (VE) number.
TTL Threshold	Following the TTL threshold value, the interface state is listed. The interface state can be one of the following: <ul style="list-style-type: none"> • Disabled • Enabled
Local Address	Indicates the IP address configured on the port or virtual interface.

Displaying a list of multicast groups

To display a list of the IP multicast groups the Layer 3 switch is forwarding, enter the **show ip pim group** command at any CLI level.

```
Brocade#show ip pim group
```

```
Total number of Groups: 2
Index 1          Group 239.255.162.1      Ports e3/11
```

Syntax: **show ip pim group**

This display shows the following information.

TABLE 256 Output of show ip pim group

Field	Description
Total number of Groups	Lists the total number of IP multicast groups the Layer 3 switch is forwarding. NOTE: This list can include groups that are not PIM Sparse groups. If interfaces on the Layer 3 switch are configured for regular PIM (dense mode) or DVMRP, these groups are listed too.
Index	The index number of the table entry in the display.
Group	The multicast group address
Ports	The Layer 3 switch ports connected to the receivers of the groups.

Displaying BSR information

To display BSR information, enter the **show ip pim bsr** command at any CLI level.

```
Brocade#show ip pim bsr
```

```
PIMv2 Bootstrap information
```

```
This system is the elected Bootstrap Router (BSR)
  BSR address: 207.95.7.1
  Uptime: 00:33:52, BSR priority: 5, Hash mask length: 32
  Next bootstrap message in 00:00:20
```

```
Next Candidate-RP-advertisement in 00:00:10
  RP: 207.95.7.1
    group prefixes:
      224.0.0.0 / 4
Candidate-RP-advertisement period: 60
```

This example shows information displayed on a Layer 3 switch that has been elected as the BSR. The following example shows information displayed on a Layer 3 switch that is not the BSR. Notice that some fields shown in the example above do not appear in the example below.

```
Brocade#show ip pim bsr
```

```
PIMv2 Bootstrap information, state=ELECTED-BSR
  local BSR address = 99.99.99.5, age=0
  local BSR priority = 255
```

Syntax: **show ip pim bsr**

Table 257 shows the information displayed for the **show ip pim bsr** command.

TABLE 257 Output of **show ip pim bsr**

Field	Description
BSR address or local BSR address	The IP address of the interface configured as the PIM Sparse Bootstrap Router (BSR). NOTE: If the word “local” does not appear in the field, this Layer 3 switch is the BSR. If the word “local” does appear, this Layer 3 switch is not the BSR.
Uptime	The amount of time the BSR has been running. NOTE: This field appears only if this Layer 3 switch is the BSR.
BSR priority or local BSR priority	The priority assigned to the interface for use during the BSR election process. During BSR election, the priorities of the candidate BSRs are compared and the interface with the highest BSR priority becomes the BSR. NOTE: If the word “local” does not appear in the field, this Layer 3 switch is the BSR. If the word “local” does appear, this Layer 3 switch is not the BSR.
Hash mask length	The number of significant bits in the IP multicast group comparison mask. This mask determines the IP multicast group numbers for which the Layer 3 switch can be a BSR. The default is 32 bits, which allows the Layer 3 switch to be a BSR for any valid IP multicast group number. NOTE: This field appears only if this Layer 3 switch is the BSR.
Next bootstrap message in	Indicates how many seconds will pass before the BSR sends its next Bootstrap message. NOTE: This field appears only if this Layer 3 switch is the BSR.
Next Candidate-PR-advertisement message in	Indicates how many seconds will pass before the BSR sends its next candidate PR advertisement message. NOTE: This field appears only if this Layer 3 switch is the BSR.
RP	Indicates the IP address of the Rendezvous Point (RP). NOTE: This field appears only if this Layer 3 switch is the BSR.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. NOTE: This field appears only if this Layer 3 switch is the BSR.
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. NOTE: This field appears only if this Layer 3 switch is the BSR.

Displaying PIM resources

To display the hardware resource information such as hardware allocation, availability, and limit for software data structure, enter the **show ip pim resource** command.

```
Brocade#show ip pim resource
```

	alloc	in-use	avail	get-fail	limit	get-mem	size	init
NBR list	256	250	6	0	512	24950	20	64
timer	256	0	256	0	8192	36	27	256
pimsm J/P elem	0	0	0	0	48960	0	9	1530
pimsm J/P Hash	50	2	48	0	11600	1384	12	50
pimsm group2rp	64	1	63	0	16384	14	10	64
pimsm L2 reg xmt	512	0	512	0	14848	100648	20	64
mcache	256	2	254	0	4096	35	137	256
mcache hash link	997	2	995	0	231304	35	8	997
mcache 2nd hash	9	0	9	0	997	0	124	9
graft if no mcache	197	0	197	0	45704	0	10	197
pim/dvm global group	256	0	256	0	59392	0	8	256
pim/dvmrp prune	128	0	128	0	40960	0	24	128
Output intf?vlan	2000	2	1998	0	464000	208	14	2000
group hash link	97	0	97	0	22504	0	8	97
2D vlan for nbr, glb	2000	2	1998	0	464000	208	14	2000
Output intf.	1024	2	1022	0	237568	208	20	1024
2D for glb grp	1024	0	1024	0	237568	0	12	1024
pim/dvm config. intf	512	508	4	0	29696	508	8	128
Prune rate limit	256	0	256	0	59392	0	24	256
Distributed add cpu	128	0	128	0	29696	951	12	128
L2 VIDX	256	1	255	0	4096	129	266	256
L2 VIDX hash	997	1	996	0	231304	129	8	997
igmp group	256	1	255	0	4096	112	16	256
igmp phy port	1024	1	1023	0	200000	112	22	1024
igmp exist phy port	13312	6019	7293	0	32768	58601	29	256
igmp G/GS query	0	0	0	0	29696	0	12	128
igmp v3 source	0	0	0	0	500000	0	10	2000
igmp v3 tracking	0	0	0	0	118784	0	10	512
igmp glb sorted list	2000	1	1999	0	500000	112	8	2000
total pool memory	680830 bytes							

```
# of PIM ports: physical 1, VEs 253 (max: 512), loopback 1, tunnels 0
```

```
Available VIDX: 4059, Blackhole VIDX: 0
```

```
Total Mlls in pool: 4095(Hw),8190(Sw) Allocated MLL: 1(Hw),2(Sw) Available MLL: 4094(Hw),8188(Sw)
```

```
SW processed pkts 0
```

Syntax: show ip pim resource

Table 258 shows the information displayed for each software data structure listed in the output of the **show ip pim resource** command.

TABLE 258 Output of **show ip pim resource**

Field	Description
alloc	Number of nodes of that data that are currently allocated in memory.
in-use	Number of allocated nodes in use
avail	Number of allocated nodes are not in use
get-fail	Number of allocated notes that failed
limit	Maximum number of nodes that can be allocated for a data structure. This may or may not be configurable, depending on the data structure
get-mem	Number of attempts made to use allocated nodes
size	Memory size (in bytes) of the given resource.
init	Initial allocated pool size of the given resource.
#of PIM ports	Total number of PIM ports, by port type, on the device
Total, allocated, and available Mlls	In Layer 3 multicast, this refers to the Multicast Linked List that contains information on where (S,G) gets forwarded. Each (S,G) entry requires a single MLL entry to forward traffic to all physical, untagged ports. Also, one MLL entry is required per VLAN that has tagged outbound ports. There can be up to 1024 MLL entries.

NOTE

When the product of the number of active PIM interfaces multiplied by the number of multicast streams exceeds the total number of MLL, the CLI displays the message, “MLL pool out of memory”.

NOTE

The total number of MLL available changes according to the hardware configuration.

Displaying candidate RP information

To display candidate RP information, enter the **show ip pim rp-candidate** command at any CLI level.

```
Brocade#show ip pim rp-candidate

Next Candidate-RP-advertisement in 00:00:10
  RP: 198.168.1.160
    group prefixes:
      224.0.0.0 / 4

Candidate-RP-advertisement period: 60
```

This example show information displayed on a Layer 3 switch that is a candidate RP. The following example shows the message displayed on a Layer 3 switch that is not a candidate RP.

```
Brocade#show ip pim rp-candidate

This system is not a Candidate-RP.
```

Syntax: **show ip pim rp-candidate**

Table 259 shows the information displayed by the **show ip pim rp-candidate** command.

TABLE 259 Output of `show ip pim rp-candidate`

Field	Description
Candidate-RP-advertisement in	Indicates how many seconds will pass before the BSR sends its next RP message. NOTE: This field appears only if this Layer 3 switch is a candidate RP.
RP	Indicates the IP address of the Rendezvous Point (RP). NOTE: This field appears only if this Layer 3 switch is a candidate RP.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. NOTE: This field appears only if this Layer 3 switch is a candidate RP.
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. NOTE: This field appears only if this Layer 3 switch is a candidate RP.

Displaying RP-to-group mappings

To display RP-to-group-mappings, enter the **show ip pim rp-map** command at any CLI level.

```
Brocade#show ip pim rp-map
Number of group-to-RP mappings: 1

Group address      RP address      age
-----
1 239.200.1.1      192.168.98.212  0
```

Syntax: `show ip pim rp-map`

[Table 260](#) shows the information displayed by the **show ip pim rp-map** command.

TABLE 260 Output of `show ip pim rp-map`

Field	Description
Group address	Indicates the PIM Sparse multicast group address using the listed RP.
RP address	Indicates the IP address of the Rendezvous Point (RP) for the listed PIM Sparse group.

Displaying RP information for a PIM Sparse group

To display RP information for a PIM Sparse group, enter the **show ip pim rp-hash** command at any CLI level.

```
Brocade#show ip pim rp-hash 225.0.0.15

RP: 192.168.1.160, version 2
Info source: 192.168.1.160, via bootstrap
```

Syntax: `show ip pim rp-hash <group-addr>`

The `<group-addr>` parameter is the address of a PIM Sparse IP multicast group.

Table 261 shows the information displayed by the **show ip pim rp-hash** command.

TABLE 261 Output of **show ip pim rp-hash**

Field	Description
RP	Indicates the IP address of the Rendezvous Point (RP) for the specified PIM Sparse group. Following the IP address is the method through which this Layer 3 Switch learned the identity of the RP.
Info source	Indicates the IP address on which the RP information was received. Following the IP address is the method through which this Layer 3 switch learned the identity of the RP.

Displaying the RP set list

To display the RP set list, enter the **show ip pim rp-set** command at any CLI level.

```
router(config)#show ip pim rp-set
No static-RP configured

BSR state=ELECTED-BSR, # of group prefixes Learntn from BSR: 1

Group prefix=224.0.0.0/4      #RPs:1
    RP1:192.168.1.160      priority=0, age=30, holdtime=150
```

Syntax: **show ip pim rp-set**

Table 262 shows the information displayed by the **show ip pim rp-set** command.

TABLE 262 Output of **show ip pim rp-set**

Field	Description
Number of group prefixes	The number of PIM Sparse group prefixes for which the RP is responsible.
Group prefix	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP.
RP <num>	Indicates the RP number. If there are multiple RPs in the PIM Sparse domain, a line of information for each of them is listed, and they are numbered in ascending numerical order.
priority	The RP priority of the candidate RP. During the election process, the candidate RP with the highest priority is elected as the RP.
age	The age (in seconds) of this RP-set. NOTE: If this Layer 3 switch is not a BSR, this field contains zero. Only the BSR ages the RP-set.
holdtime	Holdtime indicates the time allowed by the Bootstrap Router (BSR) before timing out RPs. If the BSR does not receive a candidate RP advertisement from an RP within the holdtime, the BSR removes that router from its list of candidate RPs.

Displaying multicast neighbor information

To display information about the Layer 3 switch PIM neighbors, enter the **show ip pim neighbor** command at any CLI level.

```
Brocade#show ip pim neighbor
Total number of neighbors: 250 on 1 ports
Port   Phy_p   Neighbor Holdtime   Age   UpTime  GenID
v101   3/1/12  173.1.1.2 180        60    17144   0x2CB53BBB
Port   Phy_p   Neighbor Holdtime   Age   UpTime  GenID
v102   3/1/11  173.1.2.2 180        60    17144   0x26976038
Port   Phy_p   Neighbor Holdtime   Age   UpTime  GenID
v103   3/1/13  173.1.3.2 180        60    17084   0x7F6E7643
Port   Phy_p   Neighbor Holdtime   Age   UpTime  GenID
v104   2/1/11  173.1.4.2 180        60    17144   0x6A014EB4
Port   Phy_p   Neighbor Holdtime   Age   UpTime  GenID
v105   2/1/12  173.1.5.2 180        60    17144   0x14FF45A1
```

Syntax: **show ip pim neighbor**

[Table 263](#) shows the information displayed by the **show ip pim neighbor** command.

TABLE 263 Output of **show ip pim neighbor**

Field	Description
Port	The interface through which the Layer 3 switch is connected to the neighbor.
Neighbor	The IP interface of the PIM neighbor interface.
Holdtime	Indicates how many seconds the neighbor wants this Layer 3 switch to hold the entry for this neighbor in memory. The neighbor sends the Hold Time in its Hello packets: <ul style="list-style-type: none"> If the Layer 3 switch receives a new Hello packet before the Hold Time received in the previous packet expires, the Layer 3 switch updates its table entry for the neighbor. If the Layer 3 switch does not receive a new Hello packet from the neighbor before the Hold time expires, the Layer 3 switch assumes the neighbor is no longer available and removes the entry for the neighbor.
Age	The number of seconds since the Layer 3 switch received the last hello message from the neighbor.
UpTime	The number of seconds the PIM neighbor has been up. This timer starts when the Layer 3 switch receives the first Hello messages from the neighbor.
GenID	The Generation ID of the neighbor's Hello messages.

Displaying information about an upstream neighbor device

You can view information about the upstream neighbor device for a given source IP address for IP PIM and DVMRP. The software uses the IP route table or multicast route table to lookup the upstream neighbor device. For DVMRP, the software uses the DVMRP route table to locate the upstream neighbor device.

Enter the **show ip pim rpf** command at the Privileged EXEC level of the CLI.

```
Brocade#show ip pim rpf 172.10.1.5 225.0.0.19
PIM sparse mcache of (172.10.1.5, 225.0.0.19) is used in RPF lookup
Upstream neighbor=40.1.1.1 on v40 using ip route
```

Syntax: `show ip pim | dvmrp rpf <IP address>`

where <IP address> is a valid source IP address.

NOTE

If there are multiple equal cost paths to the source, the `show ip pim rpf` command output may not be accurate. If your system has multiple equal cost paths, use the command `show ip pim mcache` to view information about the upstream neighbor.

Displaying the PIM flow cache

To display the PIM flow cache, enter the following command at any CLI level.

```
Brocade #show ip pim flowcache 228.2.2.2
```

```
Outbound Ports for Multicast flow (12.12.12.1 228.2.2.2):  
L2 : None
```

```
L3 :  
    Tagged Port in Vlan 1 : ethe 3/14
```

```
Outbound Ports for Multicast flow (0.0.0.0 228.2.2.2):  
L2 : 8192  
Hardware MC Entry not found
```

```
2 flow printed
```

Syntax: `show ip pim flowcache`

In the above output of the `show ip pim flowcache` command, ethe 3/14 is the outbound interface on which the multicast traffic is being sent out.

Displaying the PIM multicast cache

To display the PIM multicast cache, enter the **show ip pim mcache** command at any CLI level.

```
Brocade#show ip pim mcache
Total 2 entries
Example: (S G) in v40 (e2/3) cnt= : e2/3 is phy. of input v40, cnt: SW hit incl.
drop
HW: CAM switched, SW: cpu switched, OAR: SW one?arm?routing, VL: vlan
trunking: TR(e3/3,e3/4): e3/3 is primary trunk port, e3/4 is real out p.
FLAGS: fast/slow: could be HW switched or not
      pruned from upstream
      frag: packet fragmented, SW forwarded unless multicast?perf configured
      tag/tnnl: OIF has tagged/tunnel ports
      swL2/hwL2: SW or HW L2 forwarding
      age: send 1 pkt to cpu from HW switch to reset age
      0L2C: no L2 CAM hash
      ClSr: has client on input port, never send prune even OIF empty
      drop: use cam to drop pkts if no OIFs.

1      (173.21.1.13 239.200.1.1) in 1/2/4 (1/2/4), cnt=109
      Source is directly connected
      Sparse Mode, RPT=0 SPT=1 REG=0 MSDP Adv=1 MSDP Create=0
      L3 (HW) 1: tag TR(1/1/11,2/1/12)(VL101)
      fast=1 slow=0 pruned=0 graft age
      age=60s up-time=7m HW=1 L2?vidx=4131 has mll

2      (* 239.200.1.1) RP 192.168.98.212, in NIL (NIL), cnt=0
      No upstream neighbor because RP 192.168.98.212 is itself
      Sparse Mode, RPT=1 SPT=0 REG=0 MSDP Adv=0 MSDP Create=0
      L3 (SW) 1: tag TR(1/1/11,1/1/11)(VL101)
      fast=1 slow=0 pruned=1 graft
      age=0s up-time=1379m HW=0
```

Syntax: **show ip pim mcache**

Table 264 shows the information displayed by the **show ip pim mcache** command.

TABLE 264 Output of **show ip pim mcache**

Field	Description
(<source>, <group>)	The comma-separated values in parentheses is a source-group pair. The <source> is the PIM source for the multicast <group>. For example, the following entry means source 209.157.24.162 for group 239.255.162.1: (209.157.24.162,239.255.162.1) If the <source> value is * (asterisk), this cache entry uses the RP path. The * value means "all sources". If the <source> is a specific source address, this cache entry uses the SPT path.
RP<ip-addr>	Indicates the RP for the group for this cache entry. NOTE: The RP address appears only if the RPT flag is set to 1 and the SPT flag is set to 0 (see below).
forward port	The port through which the Layer 3 switch reaches the source.
Count	The number of packets forwarded using this cache entry.

TABLE 264 Output of **show ip pim mcache** (Continued)

Field	Description
Sparse Mode	Indicates whether the cache entry is for regular PIM (dense mode) or PIM Sparse. This flag can have one of the following values: <ul style="list-style-type: none"> 0 – The entry is not for PIM Sparse (and is therefore for the dense mode of PIM). 1 – The entry is for PIM Sparse.
RPT	Indicates whether the cache entry uses the RP path or the SPT path. The RPT flag can have one of the following values: <ul style="list-style-type: none"> 0 – The SPT path is used instead of the RP path. 1 – The RP path is used instead of the SPT path. NOTE: The values of the RP and SPT flags are always opposite (one is set to 0 and the other is set to 1).
SPT	Indicates whether the cache entry uses the RP path or the SPT path. The SP flag can have one of the following values: <ul style="list-style-type: none"> 0 – The RP path is used instead of the SPT path. 1 – The SPT path is used instead of the RP path. NOTE: The values of the RP and SPT flags are always opposite (one is set to 0 and the other is set to 1).
Register Suppress	Indicates whether the Register Suppress timer is running. This field can have one of the following values: <ul style="list-style-type: none"> 0 – The timer is not running. 1 – The timer is running.
member ports	Indicates the Layer 3 switch physical ports to which the receivers for the source and group are attached. The receivers can be directly attached or indirectly attached through other PIM Sparse routers.
virtual ports	Indicates the virtual interfaces to which the receivers for the source and group are attached. The receivers can be directly attached or indirectly attached through other PIM Sparse routers.
prune ports	Indicates the physical ports on which the Layer 3 switch has received a prune notification (in a Join/Prune message) to remove the receiver from the list of recipients for the group.
virtual prune ports	Indicates the virtual interfaces ports on which the Layer 3 switch has received a prune notification (in a Join/Prune message) to remove the receiver from the list of recipients for the group.

Displaying PIM traffic statistics

To display PIM traffic statistics, enter the **show ip pim traffic** command.

```
Brocade#show ip pim traffic
```

Port	Hello		J/P		Register		RegStop		Assert	
	[Rx]	[Tx]	[Rx]	[Tx]	[Rx]	[Tx]	[Rx]	[Tx]	[Rx]	[Tx]
1/2/4	0	8691	0	0	0	0	0	0	0	0
v24	0	1383	0	0	0	0	0	0	0	0
v25	0	1383	0	0	0	0	0	0	0	0
v101	44745	10091	21851	0	0	0	0	0	0	0
v102	41556	10024	0	0	0	0	0	0	0	0
v103	28035	10005	0	0	0	0	0	0	0	0

IGMP Statistics:

Total Recv/Xmit 2488418/11787722

Syntax: **show ip pim traffic**

NOTE

If you have configured interfaces for standard PIM (dense mode) on the Layer 3 switch, statistics for these interfaces are listed first by the display.

This display shows the following information.

TABLE 265 Output of show ip pim traffic

Field	Description
Port	The port or virtual interface on which the PIM interface is configured.
Hello	The number of PIM Hello messages sent or received on the interface.
J/P	The number of Join/Prune messages sent or received on the interface. NOTE: Unlike PIM dense, PIM Sparse uses the same messages for Joins and Prunes.
Register	The number of Register messages sent or received on the interface.
RegStop	The number of Register Stop messages sent or received on the interface.
Assert	The number of Assert messages sent or received on the interface.
Total Recv/Xmit	The total number of IGMP messages sent and received by the Layer 3 switch.
Total Discard/chksum	The total number of IGMP messages discarded, including a separate counter for those that failed the checksum comparison.

Clearing the PIM message counters

You can clear the PIM message counters using the **clear pim traffic** command.

```
Brocade# clear pim traffic
```

Syntax: clear pim traffic

Displaying and clearing PIM errors

If you want to determine how many PIM errors there are on the device, enter the **show ip pim error** command.

```
Brocade# show ip pim error
**** Warning counter pim route change = 1
HW tagged replication enabled, SW processed pkts 0
```

Syntax: show ip pim error

This command displays the number of warnings and non-zero PIM errors on the device. This count can increase during transition periods such as reboots and topology changes; however, if the device is stable, the number of errors should not increase. If warnings keep increasing in a stable topology, then there may be a configuration error or problems on the device.

To clear the counter for PIM errors, enter the **clear pim counters** command.

```
Brocade#clear pim counters
```

Syntax: clear pim counters

PIM Passive

PIM Passive is used to reduce and minimize unnecessary PIM Hello and other PIM control messages.

PIM Passive allows you to specify that the interface is “passive” in regards to PIM. No PIM control packets are sent or processed (if received), but hosts can still send and receive multicast traffic and IGMP control traffic on that interface. Also, PIM Passive prevents any malicious router from taking over as the designated router (DR), which can prevent all hosts on the LAN from joining multicast traffic outside the LAN.

The following guidelines apply to PIM Passive:

1. This is a Layer 3 interface [Ethernet/Ve] level feature.
2. Since the loopback interfaces are never used to form PIM neighbors, this feature is not supported on loopback interface.
3. Both PIM SM and PIM DM modes support this feature.
4. Applying the PIM Passive on an interface requires PIM to be enabled on that interface.
5. The sent and received statistics of a PIM Hello message are not changed for an interface, while it is configured as PIM passive.

To enable PIM Passive on an interface, enter the following commands:

```
Brocade# config term
Brocade(config)#router pim
Brocade(config-pim-router)#exit
Brocade(config)#interface ethernet 2
Brocade(config-if-e1000-2)#ip pim
Brocade(config-if-e1000-2)#ip pim passive
Brocade(config-if-e1000-2)#exit
Brocade(config)#interface ve 2
Brocade(config-vif-2)#ip pim-sparse
Brocade(config-vif-2)#ip pim passive
Brocade(config-vif-2)#exit
```

Syntax: [no] ip pim passive3

Multicast Source Discovery Protocol (MSDP)

The Multicast Source Discovery Protocol (MSDP) is used by Protocol Independent Multicast (PIM) Sparse routers to exchange routing information for PIM Sparse multicast groups across PIM Sparse domains. Routers running MSDP can discover PIM Sparse sources that are in other PIM Sparse domains.

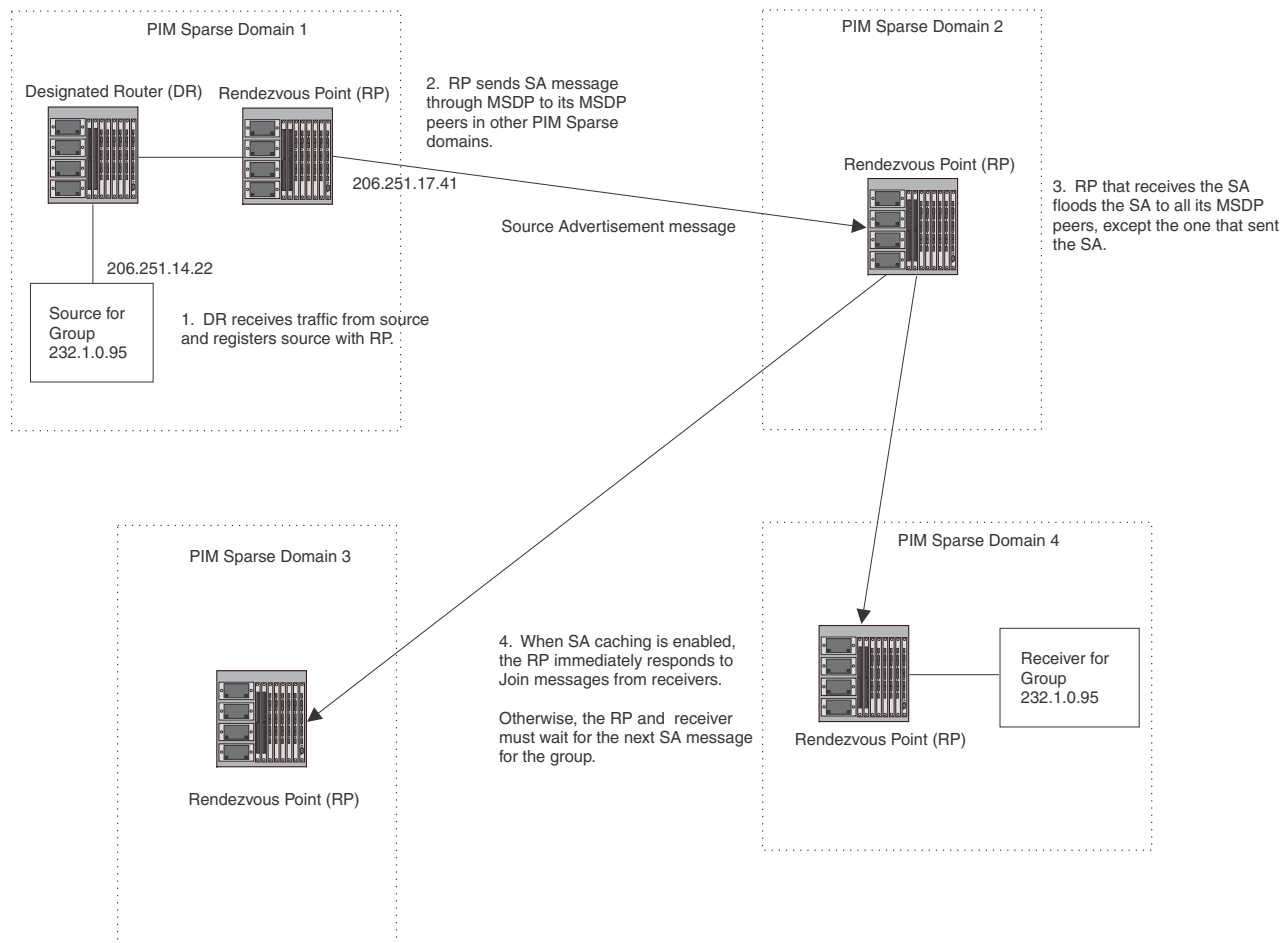
NOTE

MSDP is not supported on FCX and ICX platforms.

PIM Sparse routers use MSDP to register PIM Sparse multicast sources in a domain with the Rendezvous Point (RP) for that domain.

Figure 176 shows an example of some PIM Sparse domains. For simplicity, this example shows only one Designated Router (DR), one group source, and one receiver for the group. Only one PIM Sparse router within each domain needs to run MSDP.

FIGURE 176 PIM Sparse domains joined by MSDP routers



In this example, the source for PIM Sparse multicast group 232.0.1.95 is in PIM Sparse domain 1. The source sends a packet for the group to its directly attached DR. The DR sends a Group Advertisement message for the group to the domain RP. The RP is configured for MSDP, which enables the RP to exchange source information with other PIM Sparse domains by communicating with RPs in other domains that are running MSDP.

The RP sends the source information to each of its peers by sending a Source Active message. The message contains the IP address of the source, the group address to which the source is sending, and the IP address of the RP interface with its peer. By default, the IP address included in the RP address field of the SA message is the IP address of the originating RP. However, if MSDP is instructed to use a specific address as the IP address of the RP in a Source Address message (CLI command **originator-id** <type> <number>), the Source Active message can be the IP address of any interface on the originating RP. The interface is usually a loopback interface.

In this example, the Source Active message contains the following information:

- Source address: 206.251.14.22
- Group address: 232.1.0.95
- RP address: 206.251.17.41

Figure 176 shows only one peer for the MSDP router (which is also the RP here) in domain 1, so the Source Active message goes to only that peer. When an MSDP router has multiple peers, it sends a Source Active message to each of those peers. Each peer sends the Source Advertisement to its other MSDP peers. The RP that receives the Source Active message also sends a Join message for the group if the RP that received the message has receivers for the group.

Peer Reverse Path Forwarding (RPF) flooding

When the Multicast Source Discovery Protocol (MSDP) router (also the RP) in domain 2 receives the Source Active message from its peer in domain 1, the MSDP router in domain 2 forwards the message to all its other peers. The propagation process is sometimes called “peer Reverse Path Forwarding (RPF) flooding”. This term refers to the fact that the MSDP router uses its PIM Sparse RPF tree to send the message to its peers within the tree. In Figure 176, the MSDP router floods the Source Active message it receives from its peer in domain 1 to its other peers, in domains 3 and 4.

Note that the MSDP router in domain 2 does not forward the Source Active back to its peer in domain 1, because that is the peer from which the router received the message. An MSDP router never sends a Source Active message back to the peer that sent it. The peer that sent the message is sometimes called the “RPF peer”. The MSDP router uses the unicast routing table for its Exterior Gateway Protocol (EGP) to identify the RPF peer by looking for the route entry that is the next hop toward the source. Often, the EGP protocol is Border Gateway Protocol (BGP) version 4.

NOTE

MSDP depends on BGP for interdomain operations.

NOTE

MSDP is supported only on FSX devices.

The MSDP routers in domains 3 and 4 also forward the Source Active message to all their peers except the ones that sent them the message. Figure 176 does not show additional peers.

Source active caching

When the Multicast Source Discovery Protocol (MSDP) router that is also an RP receives a Source Active message, the RP checks its PIM Sparse multicast group table for receivers for the group. If the DR has a receiver for the group being advertised in the Source Active message, the DR sends a Join message for that receiver back to the DR in the domain from which the Source Active message came. Usually, the DR is also the MSDP router that sent the Source Active message.

In Figure 176, if the MSDP router and RP in domain 4 has a table entry for the receiver, the RP sends a Join message on behalf of the receiver back through the RPF tree to the RP for the source, in this case the RP in domain 1.

Some MSDP routers that are also RPs can cache Source Active messages. If the RP is not caching Source Active messages, the RP does not send a Join message unless it already has a receiver that wants to join the group. Otherwise, the RP does not send a Join message and does not remember the information in the Source Active message after forwarding it. If the RP receives a request from a receiver for the group, the RP and receiver must wait for the next Source Active message for that group before the RP can send a Join message for the receiver.

However, if Source Active caching is enabled on the MSDP and RP router, the RP caches the Source Active messages it receives. In this case, even if the RP does not have a receiver for a group when the RP receives the Source Active message for the group, the RP can immediately send a Join for a new receiver that wants to join the group, without waiting for the next Source Active message from the RP in the source domain.

The maximum size of the cache used to store MSDP Source Active messages is 8K and the default size is 4K.

MSDP configuration

To configure Multicast Source Discovery Protocol (MSDP) on a Layer 3 switch, perform the following tasks:

- Enable MSDP
- Configure the MSDP peers

NOTE

The PIM Sparse Rendezvous Point (RP) is also an MSDP peer.

NOTE

Routers that run MSDP must also run BGP. Also, the source address used by the MSDP router must be the same source address used by BGP.

Enabling MSDP

To enable MSDP, enter the **router msdp** command.

```
Brocade(config)# router msdp
```

NOTE

When enabling and disabling MSDP, you do not need to save the configuration and reload the software. The configuration change takes effect immediately, as soon as you enter the CLI command.

Syntax: [no] router msdp

Configuring MSDP peers

To configure an MSDP peer, enter the **msdp-peer** command at the MSDP configuration level.

```
Brocade(config-msdp-router)# msdp-peer 205.216.162.1
```

Syntax: [no] msdp-peer <ip-addr> [connect-source loopback <num>]

The <ip-addr> parameter specifies the IP address of the neighbor.

The **connect-source loopback <num>** parameter specifies the loopback interface you want to use as the source for sessions with the neighbor.

NOTE

It is strongly recommended that you use the **connect-source loopback <num>** parameter when issuing the **msdp-peer** command. If you do not use this parameter, the Layer 3 switch uses the subnet interface configured on the port.

Also, make sure the IP address of the connect-source loopback is the same as the source IP address used by the MSDP router, the PIM-RP, and the BGP router.

The commands in the following example add an MSDP neighbor and specify a loopback interface as the source interface for sessions with the neighbor. By default, the Layer 3 switch uses the subnet address configured on the physical interface where you configure the neighbor as the source address for sessions with the neighbor.

```
Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 9.9.9.9/32
Brocade(config-lbif-1)#exit
Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 2.2.2.99 connect-source loopback 1
```

Designating an interface IP address as the RP IP address

When an RP receives a Source Active message, it checks its PIM Sparse multicast group table for receivers for the group. If it finds a receiver, the RP sends a Join message for that receiver back to the RP that originated the Source Active message. The originator RP is identified by its RP address.

By default, the IP address included in the RP address field of the SA message is the IP address of the originating RP. Beginning with this release, an SA message can use the IP address of any interface on the originating RP. (The interface is usually a loopback interface.)

To designate an interface IP address to be the IP address of the RP, enter commands such as the following:

```
Brocade(config)# interface loopback 2
Brocade(config-lbif-2)# ip address 2.2.1.99/32
Brocade(config)# router msdp
Brocade(config-msdp-router)# originator-id loopback 2
Brocade(config-msdp-router)# exit
```

Syntax: [no] **originator-id** <type> <number>

The **originator-id** parameter instructs MSDP to use the specified address as the IP address of the RP in an SA message. This address must be the address of the interface used to connect the RP to the source. There are no default originator-ids.

The <type> parameter indicates the type of interface used by the RP. Ethernet, loopback and virtual routing interfaces (ve) can be used.

The <number> parameter specifies the interface number (for example: loopback number, port number or virtual routing interface number.)

Filtering MSDP source-group pairs

The following commands allow you to filter individual source-group pairs in MSDP Source-Active messages:

- **sa-filter in** – Filters source-group pairs received in Source-Active messages from an MSDP neighbor
- **sa-filter originate** – Filters source-group pairs in Source-Active messages in advertisements to an MSDP neighbor

Filtering incoming source-active messages

The following example configures filters for incoming Source-Active messages from three MSDP neighbors:

- For peer 2.2.2.99, all source-group pairs in Source-Active messages from the neighbor are filtered out (dropped).
- For peer 2.2.2.97, all source-group pairs except those with 10.x.x.x as the source are permitted.
- For peer 2.2.2.96, all source-group pairs except those associated with RP 2.2.42.3 are permitted.

Example of configuring MSDP peers

The following commands configure an IP address on port 3/1. This is the port on which the MSDP neighbors will be configured.

```
Brocade(config)# interface ethernet 3/1
Brocade(config-if-3/1)# ip address 2.2.2.98/24
Brocade(config-if-3/1)# exit
```

The following commands configure a loopback interface. The Layer 3 switch will use this interface as the source address for communicating with the MSDP neighbors.

```
Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 9.9.9.8/32
Brocade(config-lbif-1)# exit
```

The following commands configure extended ACLs. The ACLs will be used in route maps, which will be used by the Source-Active filters.

```
Brocade(config)# access-list 123 permit 10.0.0.0 0.255.255.255 any
Brocade(config)# access-list 124 permit 2.2.42.3 0.0.0.0 any
Brocade(config)# access-list 125 permit any any
```

The following commands configure the route maps.

```
Brocade(config)# route-map msdp_map deny 1
Brocade(config-routemap msdp_map)# match ip address 123
Brocade(config-routemap msdp_map)# route-map msdp_map permit 10
Brocade(config-routemap msdp_map)# exit
Brocade(config)# route-map msdp2_map permit 1
Brocade(config-routemap msdp2_map)# match ip address 125
Brocade(config-routemap msdp2_map)# exit
Brocade(config)# route-map msdp2_rp_map deny 1
Brocade(config-routemap msdp2_rp_map)# match ip route-source 124
Brocade(config-routemap msdp2_rp_map)# route-map msdp2_rp_map permit 10
Brocade(config-routemap msdp2_rp_map)# exit
```

NOTE

The default action rule for **route-map** is to deny all routes that are not explicitly permitted. If you configure a “deny” route map but want to permit other routes that do not match the rule, configure an “empty” permit route map, as shown in the following example.

```
Brocade(config)#route-map abc deny 10
Brocade(config-route-map abc)#match metric 20
Brocade(config-route-map abc)#route-map abc permit 20
```

Without the last line in the above example, all routes would be denied.

The following commands enable MSDP and configure the MSDP neighbors on port 3/1.

```
Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 2.2.2.99 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 2.2.2.97 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 2.2.2.96 connect-source loopback 1
Brocade(config-msdp-router)# exit
```

The following commands configure the Source-Active filters.

```
Brocade(config)# router msdp
Brocade(config-msdp-router)# sa-filter in 2.2.2.99
Brocade(config-msdp-router)# sa-filter in 2.2.2.97 route-map msdp_map
Brocade(config-msdp-router)# sa-filter in 2.2.2.96 route-map msdp2_map
rp-route-map msdp2_rp_map
```

The **sa-filter** commands configure the following filters:

- **sa-filter in 2.2.2.99** – This command drops all source-group pairs received from neighbor 2.2.2.99.

NOTE

The default action is to deny all source-group pairs from the specified neighbor. If you want to permit some pairs, use route maps.

- **sa-filter in 2.2.2.97 route-map msdp_map** – This command drops source-group pairs received from neighbor 2.2.2.97 if the pairs have source address 10.x.x.x and any group address.
- **sa-filter in 2.2.2.96 route-map msdp2_map rp-route-map msdp2_rp_map** – This command accepts all source-group pairs except those associated with RP 2.2.42.3.

Syntax: [no] **sa-filter** in <ip-addr> [route-map <map-tag>] [rp-route-map <rp-map-tag>]

The <ip-addr> parameter specifies the IP address of the MSDP neighbor. The filter applies to Active-Source messages received from this neighbor.

The **route-map <map-tag>** parameter specifies a route map. The Layer 3 switch applies the filter to source-group pairs that match the route map. Use the **match ip address <acl-id>** command in the route map to specify an extended ACL that contains the source and group addresses.

The **rp-route-map <rp-map-tag>** parameter specifies a route map to use for filtering based on Rendezvous Point (RP) address. Use this parameter if you want to filter Source-Active messages based on their origin. If you use the **route-map** parameter instead, messages are filtered based on source-group pairs but not based on origin. Use the **match ip route-source <acl-id>** command in the route map to specify the RP address.

NOTE

The default filter action is deny. If you want to permit some source-group pairs, use a route map. A permit action in the route map allows the Layer 3 switch to receive the matching source-group pairs. A deny action in the route map drops the matching source-group pairs.

Filtering advertised source-active messages

The following example configures the Layer 3 switch to advertise all source-group pairs except the ones that have source address 10.x.x.x.

Example of configuring source Active (SA) filters

The following commands configure an IP address on port 3/1. This is the port on which the MSDP neighbors will be configured.

```
Brocade(config)# interface ethernet 3/1
Brocade(config-if-3/1)# ip address 2.2.2.98/24
Brocade(config-if-3/1)# exit
```

The following commands configure a loopback interface. The Layer 3 switch will use this interface as the source address for communicating with the MSDP neighbors.

```
Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 9.9.9.8/32
Brocade(config-lbif-1)# exit
```

The following command configures an extended ACL to specify the source and group addresses you want to filter.

```
Brocade(config)# access-list 123 permit 10.0.0.0 0.255.255.255 any
```

The following commands configure a route map. The map matches on source address 10.x.x.x and any group address. Since the action is deny, the Source-Active filter that uses this route map will remove the source-group pairs that match this route map from the Source-Active messages to the neighbor.

```
Brocade(config)# route-map msdp_map deny 1
Brocade(config-routemap msdp_map)# match ip address 123
Brocade(config-routemap msdp_map)# exit
```

The following commands enable MSDP and configure MSDP neighbors.

```
Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 2.2.2.99 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 2.2.2.97 connect-source loopback 1
Brocade(config-msdp-router)# exit
```

The following commands configure the Source-Active filter.

```
Brocade(config)# router msdp
Brocade(config-msdp-router)# sa-filter originate route-map msdp_map
```

This filter removes source-group pairs that match route map msdp_map from Source-Active messages before sending them to MSDP neighbors.

Syntax: [no] sa-filter originate [route-map <map-tag>]

The **route-map** *<map-tag>* parameter specifies a route map. The Layer 3 switch applies the filter to source-group pairs that match the route map. Use the **match ip address** *<acl-id>* command in the route map to specify an extended ACL that contains the source and group addresses.

NOTE

The default filter action is deny. If you want to permit some source-group pairs, use a route map. A permit action in the route map allows the Layer 3 switch to receive the matching source-group pairs. A deny action in the route map drops the matching source-group pairs.

MSDP mesh groups

A PIM Sparse domain can have several RPs that are connected to each other to form an MSDP mesh group. To qualify as a mesh group, the RPs have to be fully meshed; that is, each RP must be connected to all peer RPs in a domain. (See Figure 177.)

A mesh group reduces the forwarding of SA messages within a domain. Instead of having every RP in a domain forward SA messages to all the RPs within that domain, only one RP forwards the SA message. Since an MSDP mesh group is fully meshed, peers do not forward SA messages received in a domain from one member to every member of the group. The RP that originated the SA or the first RP in a domain that receives the SA message is the only one that can forward the message to the members of a mesh group. An RP can forward an SA message to any MSRP router as long as that peer is farther away from the originating RP than the current MSRP router.

Figure 177 shows an example of an MSDP mesh group. In a PIM-SM mesh group the RPs are configured to be peers of each other. They can also be peers of RPs in other domains.

Configuring an MSDP mesh group

To configure an MSDP mesh group, enter commands such as the following on each device that will be included in the mesh group:

```
Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 17.17.17.7
Brocade(config-msdp-router)# mesh-group 1234 1.1.4.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.3.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.2.1
Brocade(config-msdp-router)# exit
```

Syntax: [no] **mesh-group** *<group-name>* *<peer-address>*

The example configuration above reflects the configuration in Figure 177. On RP 1.1.1.1, you specify its peers within the same domain (1.1.3.1, 1.1.4.1, and 1.1.2.1).

You first configure the MSDP peers using the **msdp-peer** command to assign their IP addresses and the loopback interfaces. This information will be used as the source for sessions with the neighbor.

Next, place the MSDP peers within a domain into a mesh group. Use the **mesh-group** command. There are no default mesh groups.

The **group-name** parameter identifies the group. Enter up to 31 characters for group-name. You can have up to 4 mesh groups within a multicast network. Each mesh group can include up to 32 peers.

The **peer-address** parameter specifies the IP address of the MSDP peer that is being placed in the group.

NOTE

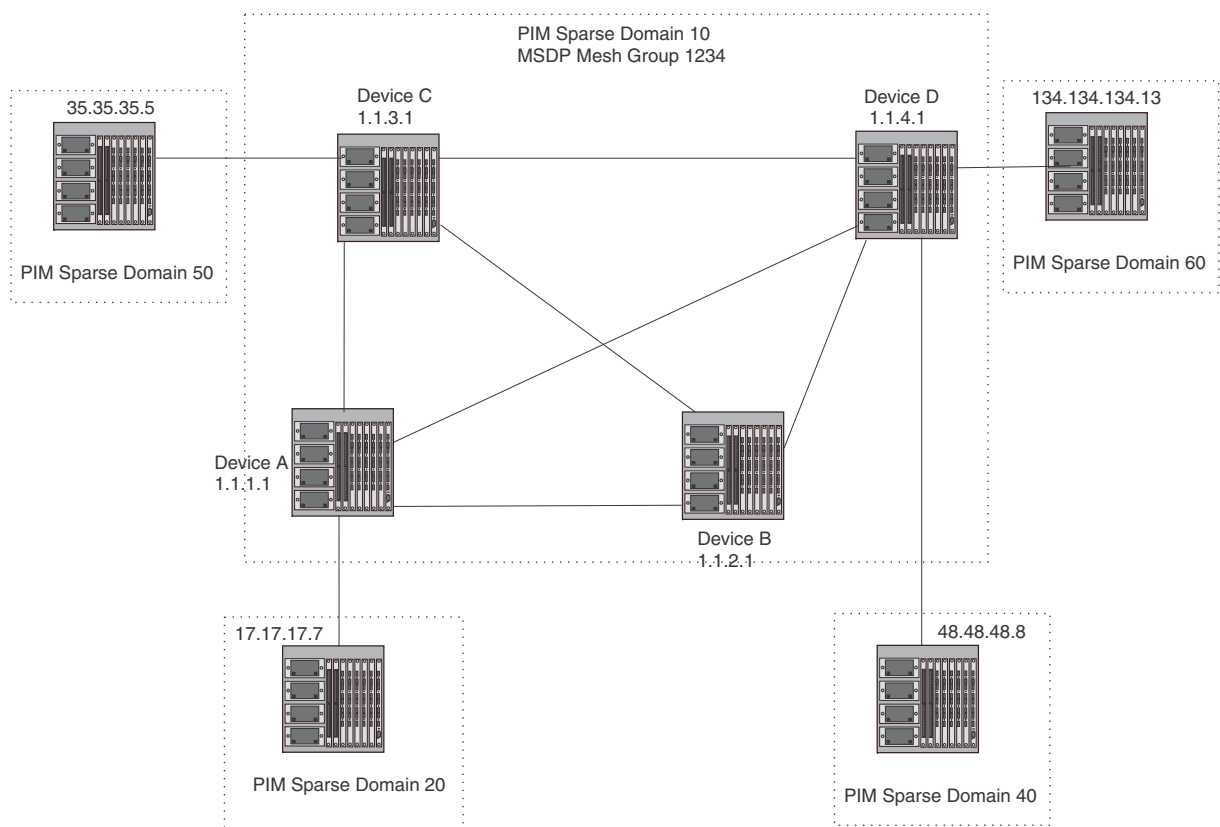
On each of the device that will be part of the mesh-group, there must be a mesh-group definition for all the peers in the mesh-group.

Up to 32 MSDP peers can be configured per mesh group.

Example configuration of an MSDP mesh group

In Figure 177, devices A, B, C, and D are in Mesh Group 1234. The example configuration following the figure shows how the devices are configured to be part of the MSDP mesh group. The example also shows the features that need to be enabled for the MSDP mesh group to work.

FIGURE 177 MSDP mesh group 1234



MSDP mesh group configuration for Device A

The following set of commands configure the MSDP peers of Device A (1.1.1.1) that are inside and outside MSDP mesh group 1234. Device A peers inside the mesh group 1234 are 1.1.2.1, 1.1.3.1, and 1.1.4.1. Device 17.17.17.7 is a peer of Device A, but is outside mesh group 1234. Multicast is enabled on Device A interfaces. PIM and BGP are also enabled.

```
Brocade(config)# router pim
```

```

Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 17.17.17.7
Brocade(config-msdp-router)# mesh-group 1234 1.1.4.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.3.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.2.1
Brocade(config-msdp-router)# exit

Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 1.1.1.1 255.255.255.0
Brocade(config-lbif-1)# ip pim-sparse
Brocade(config-lbif-1)# exit

Brocade(config)# interface ethernet 1/1
Brocade(config-if-1/1)# ip address 14.14.14.1 255.255.255.0
Brocade(config-if-1/1)# ip pim-sparse
Brocade(config-if-1/1)# exit

Brocade(config)# interface ethernet 2/1
Brocade(config-if-2/1)# ip address 12.12.12.1 255.255.255.0
Brocade(config-if-2/1)# ip pim-sparse
Brocade(config-if-2/1)# exit

Brocade(config)# interface ethernet 2/20
Brocade(config-if-2/20)# ip address 159.159.159.1 255.255.255.0
Brocade(config-if-2/20)# ip pim-sparse
Brocade(config-if-2/20)# exit

Brocade(config)# interface ethernet 4/1
Brocade(config-if-4/1)# ip address 31.31.31.1 255.255.255.0
Brocade(config-if-4/1)# ip pim-sparse
Brocade(config-if-4/1)# exit

Brocade(config)# interface ethernet 4/8
Brocade(config-if-4/8)# ip address 17.17.17.1 255.255.255.0
Brocade(config-if-4/8)# ip pim-sparse
Brocade(config-if-4/8)# ip pim border
Brocade(config-if-4/8)# exit

Brocade(config)# router pim
Brocade(config-router-pim)# bsr-candidate loopback 1 1 31
Brocade(config-router-pim)# rp-candidate loopback 1
Brocade(config-router-pim)# exit

Brocade(config)# router bgp
Brocade(config-bgp-router)# local-as 111
Brocade(config-bgp-router)# neighbor 31.31.31.3 remote-as 333
Brocade(config-bgp-router)# neighbor 31.31.31.3 next-hop-self
Brocade(config-bgp-router)# neighbor 12.12.12.2 remote-as 222
Brocade(config-bgp-router)# neighbor 12.12.12.2 next-hop-self
Brocade(config-bgp-router)# neighbor 14.14.14.4 remote-as 444
Brocade(config-bgp-router)# neighbor 14.14.14.4 next-hop-self
Brocade(config-bgp-router)# neighbor 17.17.17.7 remote-as 777
Brocade(config-bgp-router)# neighbor 17.17.17.7 next-hop-self
Brocade(config-bgp-router)# redistribute connected
Brocade(config-bgp-router)# write memory

```

MSDP mesh group configuration for Device B

The following set of commands configure the MSDP peers of Device B. All Device B peers (1.1.1.1, 1.1.3.1, and 1.1.4.1) are in the MSDP mesh group 1234. Multicast is enabled on Device B interfaces. PIM and BGP are also enabled.

```

Brocade(config)# router pim

Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
Brocade(config-msdp-router)# mesh-group 1234 1.1.1.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.3.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.4.1
Brocade(config-msdp-router)# exit

Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 1.1.2.1 255.255.255.0
Brocade(config-lbif-1)# ip pim-sparse
Brocade(config-lbif-1)# exit

Brocade(config)# interface ethernet 1/1
Brocade(config-if-1/1)# ip address 12.12.12.2 255.255.255.0
Brocade(config-if-1/1)# ip pim-sparse
Brocade(config-if-1/1)# exit

Brocade(config)# interface ethernet 1/12
Brocade(config-if-1/12)# ip address 165.165.165.1 255.255.255.0
Brocade(config-if-1/12)# ip pim-sparse
Brocade(config-if-1/12)# exit

Brocade(config)# interface ethernet 1/24
Brocade(config-if-1/24)# ip address 168.72.2.2 255.255.255.0
Brocade(config-if-1/24)# exit

Brocade(config)# interface ethernet 1/25
Brocade(config-if-1/25)# ip address 24.24.24.2 255.255.255.0
Brocade(config-if-1/25)# ip pim-sparse
Brocade(config-if-1/24)# exit

Brocade(config)# interface ethernet 8/1
Brocade(config-if-8/1)# ip address 32.32.32.2 255.255.255.0
Brocade(config-if-8/1)# ip pim-sparse
Brocade(config-if-1/24)# exit

Brocade(config)# router pim
Brocade(config-router-pim)# bsr-candidate loopback 1 2 32
Brocade(config-router-pim)# rp-candidate loopback 1
Brocade(config-router-pim)# exit

Brocade(config)# router bgp
Brocade(config-router-bgp)# local-as 222
Brocade(config-router-bgp)# neighbor 32.32.32.3 remote-as 333
Brocade(config-router-bgp)# neighbor 32.32.32.3 next-hop-self
Brocade(config-router-bgp)# neighbor 24.24.24.4 remote-as 444
Brocade(config-router-bgp)# neighbor 24.24.24.4 next-hop-self
Brocade(config-router-bgp)# neighbor 12.12.12.1 remote-as 111

```



```

Brocade(config-router-bgp)# neighbor 12.12.12.1 next-hop-self
Brocade(config-router-bgp)# redistribute connected
Brocade(config-router-bgp)# write memory

```

MSDP mesh group configuration for Device C

The following set of commands configure the MSDP peers of Device C (1.1.3.1) that are inside and outside MSDP mesh group 1234. Device C peers inside the mesh group 1234 are 1.1.1.1, 1.1.2.1, and 1.1.4.1. Device 35.35.35.5 is a peer of Device C, but is outside mesh group 1234. Multicast is enabled on Device C interfaces. PIM and BGP are also enabled.

```

Brocade(config)# router pim

Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 35.35.35.5
Brocade(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
Brocade(config-msdp-router)# mesh-group 1234 1.1.2.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.1.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.4.1
Brocade(config-msdp-router)# exit

Brocade(config)# interface loopback 1
Brocade(config-lbif-1)# ip address 1.1.3.1 255.255.255.0
Brocade(config-lbif-1)# ip pim-sparse
Brocade(config-lbif-1)# exit

Brocade(config)# interface ethernet 3/1
Brocade(config-if-3/1)# ip address 32.32.32.3 255.255.255.0
Brocade(config-if-3/1)# ip pim-sparse
Brocade(config-if-3/1)# exit

Brocade(config)# interface ethernet 10/1
Brocade(config-if-10/1)# ip address 31.31.31.3 255.255.255.0
Brocade(config-if-10/1)# ip pim-sparse
Brocade(config-if-10/1)# exit

Brocade(config)# interface ethernet 10/8
Brocade(config-if-10/8)# ip address 35.35.35.3 255.255.255.0
Brocade(config-if-10/8)# ip pim-sparse
Brocade(config-if-10/8)# ip pim border
Brocade(config-if-10/8)# exit

Brocade(config)# interface ethernet 12/2
Brocade(config-if-12/1)# ip address 34.34.34.3 255.255.255.0
Brocade(config-if-12/1)# ip pim-sparse
Brocade(config-if-12/1)# exit

Brocade(config)# interface ethernet 14/4
Brocade(config-if-14/4)# ip address 154.154.154.1 255.255.255.0
Brocade(config-if-12/1)# ip pim-sparse
Brocade(config-if-12/1)# exit

Brocade(config)# router pim
Brocade(config-router-pim)# bsr-candidate loopback 1 1 3
Brocade(config-router-pim)# rp-candidate loopback 1
Brocade(config-router-pim)# exit

```

```

Brocade(config)# router bgp
Brocade(config-router-bsr)# local-as 333
Brocade(config-router-bsr)# neighbor 35.35.35.5 remote-as 555
Brocade(config-router-bsr)# neighbor 35.35.35.5 next-hop-self
Brocade(config-router-bsr)# neighbor 32.32.32.2 remote-as 222
Brocade(config-router-bsr)# neighbor 32.32.32.2 next-hop-self
Brocade(config-router-bsr)# neighbor 34.34.34.4 remote-as 444
Brocade(config-router-bsr)# neighbor 34.34.34.4 next-hop-self
Brocade(config-router-bsr)# neighbor 31.31.31.1 remote-as 111
Brocade(config-router-bsr)# neighbor 31.31.31.1 next-hop-self
Brocade(config-router-bsr)# redistribute connected
Brocade(config-router-bsr)# write memory

```

MSDP mesh group configuration for Device D

The following set of commands configure the MSDP peers of Device D (1.1.4.1) that are inside and outside MSDP mesh group 1234. Device D peers inside the mesh group 1234 are 1.1.1.1, 1.1.2.1, and 1.1.3.1. Device 48.48.48.8 and 134.134.134.13 are also peers of Device D, but are outside mesh group 1234. Multicast is enabled on Device D interfaces. PIM and BGP are also enabled.

```

Brocade(config)# router pim

Brocade(config)# router msdp
Brocade(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
Brocade(config-msdp-router)# msdp-peer 48.48.48.8
Brocade(config-msdp-router)# msdp-peer 134.134.134.13
Brocade(config-msdp-router)# mesh-group 1234 1.1.1.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.3.1
Brocade(config-msdp-router)# mesh-group 1234 1.1.2.1
Brocade(config-msdp-router)# exit
Brocade(config)# interface loopback 1
Brocade(config-lbif-)# ip address 1.1.4.1 255.255.255.0
Brocade(config-lbif-)# ip pim-sparse
Brocade(config-lbif-)# exit
Brocade(config)# interface ethernet 1/1
Brocade(config-if-)# ip address 24.24.24.4 255.255.255.0
Brocade(config-if-)# ip pim-sparse
Brocade(config-if-)# exit
Brocade(config)# interface ethernet 2/6
Brocade(config-if-)# ip address 156.156.156.1 255.255.255.0
Brocade(config-if-)# ip pim-sparse
Brocade(config-if-)# exit
Brocade(config)# interface ethernet 5/1
Brocade(config-if-)# ip address 34.34.34.4 255.255.255.0
Brocade(config-if-)# ip pim-sparse
Brocade(config-if-)# exit
Brocade(config)# interface ethernet 7/1
Brocade(config-if-)# ip address 14.14.14.4 255.255.255.0
Brocade(config-if-)# ip pim-sparse
Brocade(config-if-)# exit
Brocade(config)# interface ethernet 7/7
Brocade(config-if-)# ip address 48.48.48.4 255.255.255.0
Brocade(config-if-)# ip pim-sparse
Brocade(config-if-)# ip pim border
Brocade(config-if-)# exit

```

```

Brocade(config)# interface ethernet 7/8
Brocade(config-if-)# ip address 134.134.134.4 255.255.255.0
Brocade(config-if-)# ip pim-sparse
Brocade(config-if-)# ip pim border
Brocade(config-if-)# exit
Brocade(config)# router pim
Brocade(config-router-pim)# bsr-candidate loopback 1 14 34
Brocade(config-router-pim)# rp-candidate loopback 1
Brocade(config-router-pim)# exit
Brocade(config)# router bgp
Brocade(config-router-bsr)# local-as 444
Brocade(config-router-bsr)# neighbor 34.34.34.3 remote-as 333
Brocade(config-router-bsr)# neighbor 34.34.34.3 next-hop-self
Brocade(config-router-bsr)# neighbor 14.14.14.1 remote-as 111
Brocade(config-router-bsr)# neighbor 14.14.14.1 next-hop-self
Brocade(config-router-bsr)# neighbor 24.24.24.2 remote-as 222
Brocade(config-router-bsr)# neighbor 24.24.24.2 next-hop-self
Brocade(config-router-bsr)# neighbor 48.48.48.8 remote-as 888
Brocade(config-router-bsr)# neighbor 48.48.48.8 next-hop-self
Brocade(config-router-bsr)# neighbor 134.134.134.13 remote-as 1313
Brocade(config-router-bsr)# neighbor 134.134.134.13 next-hop-self
Brocade(config-router-bsr)# redistribute connected
Brocade(config-router-bsr)# write memory

```

Displaying MSDP information

You can display the following MSDP information:

- Summary information – the IP addresses of the peers, the state of the Layer 3 switch MSDP session with each peer, and statistics for Keepalive, Source Active, and Notification messages sent to and received from each of the peers
- Peer information – the IP address of the peer, along with detailed MSDP and TCP statistics
- Source Active cache entries – the Source Active messages cached by the Layer 3 switch

Displaying summary information

To display summary MSDP information, enter the **show ip msdp summary** command at any level of the CLI:

```
Brocade(config-msdp-router)# show ip msdp summary
```

MSDP Peer Status Summary

KA: Keepalive SA:Source-Active NOT: Notification

Peer Address	State	KA		SA		NOT	
		In	Out	In	Out	In	Out
206.251.17.30	ESTABLISH	3	3	0	640	0	0
206.251.17.41	ESTABLISH	0	3	651	0	0	0

Syntax: show ip msdp summary

This display shows the following information.

TABLE 266 MSDP summary information

Field	Description
Peer Address	The IP address of the peer interface with the Layer 3 switch
State	The state of the MSDP router connection with the peer. The state can be one of the following: <ul style="list-style-type: none">• CONNECTING – The session is in the active open state.• ESTABLISHED – The MSDP session is fully up.• INACTIVE – The session is idle.• LISTENING – The session is in the passive open state.
KA In	The number of MSDP Keepalive messages the MSDP router has received from the peer
KA Out	The number of MSDP Keepalive messages the MSDP router has sent to the peer
SA In	The number of Source Active messages the MSDP router has received from the peer
SA Out	The number of Source Active messages the MSDP router has sent to the peer
NOT In	The number of Notification messages the MSDP router has received from the peer
NOT Out	The number of Notification messages the MSDP router has sent to the peer

Displaying peer information

To display summary MSDP peer information, enter the **show ip msdp peer** command.

```
Brocade(config-msdp-router)# show ip msdp peer
```

```

Total number of MSDP Peers: 2

1      IP Address      State
      206.251.17.30    ESTABLISHED
      Keep Alive Time  Hold Time
      60               90

      Message Sent      Message Received
      Keep Alive        2               3
      Notifications     0               0
      Source-Active     0               640
      Last Connection Reset Reason:Reason Unknown
      Notification Message Error Code Received:Unspecified
      Notification Message Error SubCode Received:Not Applicable
      Notification Message Error Code Transmitted:Unspecified
      Notification Message Error SubCode Transmitted:Not Applicable
      TCP Connection state: ESTABLISHED
      Local host: 206.251.17.29, Local Port: 8270
      Remote host: 206.251.17.30, Remote Port: 639
      ISentSeq: 16927 SendNext: 685654 TotUnAck: 0
      SendWnd: 16384 TotSent: 668727 ReTrans: 1
      IRcvSeq: 45252428 RcvNext: 45252438 RcvWnd: 16384
      TotalRcv: 10 RcvQue: 0 SendQue: 0

```

Syntax: show ip msdp peer

This display shows the following information.

TABLE 267 MSDP peer information

Field	Description
Total number of MSDP peers	The number of MSDP peers configured on the Layer 3 switch
IP Address	The IP address of the peer interface with the Layer 3 switch
State	The state of the MSDP router connection with the peer. The state can be one of the following: <ul style="list-style-type: none"> CONNECTING – The session is in the active open state. ESTABLISHED – The MSDP session is fully up. INACTIVE – The session is idle. LISTENING – The session is in the passive open state.
Keep Alive Time	The keep alive time, which specifies how often this MSDP router sends keep alive messages to the neighbor. The keep alive time is 60 seconds and is not configurable.
Hold Time	The hold time, which specifies how many seconds the MSDP router will wait for a KEEPALIVE or UPDATE message from an MSDP neighbor before deciding that the neighbor is dead. The hold time is 90 seconds and is not configurable.
Keep Alive Message Sent	The number of Keep Alive messages the MSDP router has sent to the peer.
Keep Alive Message Received	The number of Keep Alive messages the MSDP router has received from the peer.
Notifications Sent	The number of Notification messages the MSDP router has sent to the peer.

TABLE 267 MSDP peer information (Continued)

Field	Description
Notifications Received	The number of Notification messages the MSDP router has received from the peer.
Source-Active Sent	The number of Source Active messages the MSDP router has sent to the peer.
Source-Active Received	The number of Source Active messages the MSDP router has received from the peer.
Last Connection Reset Reason	The reason the previous session with this neighbor ended.
Notification Message Error Code Received	<p>If the MSDP router receives a NOTIFICATION messages from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages:</p> <ul style="list-style-type: none"> • 1 – Message Header Error • 2 – SA-Request Error • 3 – SA-Message/SA-Response Error • 4 – Hold Timer Expired • 5 – Finite State Machine Error • 6 – Notification • 7 – Cease <p>For information about these error codes, see section 17 in the Internet draft describing MSDP, “draft-ietf-msdp-spec”.</p>
Notification Message Error SubCode Received	See above.
Notification Message Error Code Transmitted	The error message corresponding to the error code in the NOTIFICATION message this MSDP router sent to the neighbor. See the description for the Notification Message Error Code Received field for a list of possible codes.
Notification Message Error SubCode Transmitted	See above.

TCP Statistics

TABLE 267 MSDP peer information (Continued)

Field	Description
TCP connection state	<p>The state of the connection with the neighbor. The connection can have one of the following states:</p> <ul style="list-style-type: none"> • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Local host	The IP address of the MSDP router interface with the peer.
Local port	The TCP port the MSDP router is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port number of the peer end of the connection.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the MSDP router that have not been acknowledged by the neighbor.
SendWnd	The size of the send window.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the MSDP router retransmitted because they were not acknowledged.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
RcvWnd	The size of the receive window.
TotalRcv	The number of sequence numbers received from the neighbor.
RcvQue	The number of sequence numbers in the receive queue.
SendQue	The number of sequence numbers in the send queue.

Displaying source active cache information

To display the Source Actives in the MSDP cache, enter the **show ip msdp sa-cache** command.

```
Brocade(config-msdp-router)# show ip msdp sa-cache
```

```
Total Entry 4096, Used 1800 Free 2296
Index   SourceAddr  GroupAddr          Age
1      (100.100.1.254, 232.1.0.95), RP:206.251.17.41, Age:0
2      (100.100.1.254, 237.1.0.98), RP:206.251.17.41, Age:30
3      (100.100.1.254, 234.1.0.48), RP:206.251.17.41, Age:30
4      (100.100.1.254, 239.1.0.51), RP:206.251.17.41, Age:30
5      (100.100.1.254, 234.1.0.154), RP:206.251.17.41, Age:30
6      (100.100.1.254, 236.1.0.1), RP:206.251.17.41, Age:30
7      (100.100.1.254, 231.1.0.104), RP:206.251.17.41, Age:90
8      (100.100.1.254, 239.1.0.157), RP:206.251.17.41, Age:30
9      (100.100.1.254, 236.1.0.107), RP:206.251.17.41, Age:30
10     (100.100.1.254, 233.1.0.57), RP:206.251.17.41, Age:90
```

Syntax: **show ip msdp sa-cache**

This display shows the following information.

TABLE 268 MSDP source active cache

Field	Description
Total Entry	The total number of entries the cache can hold.
Used	The number of entries the cache currently contains.
Free	The number of additional entries for which the cache has room.
Index	The cache entry number.
SourceAddr	The IP address of the multicast source.
GroupAddr	The IP multicast group to which the source is sending information.
RP	The RP through which receivers can access the group traffic from the source
Age	The number of seconds the entry has been in the cache

Clearing MSDP information

You can clear the following MSDP information:

- Peer information
- Source Active cache
- MSDP statistics

Clearing peer information

To clear MSDP peer information, enter the **clear ip msdp peer** command at the Privileged EXEC level of the CLI:

```
Brocade# clear ip msdp peer 205.216.162.1
Remote connection closed
```

Syntax: **clear ip msdp peer** <ip-addr>

The command in this example clears the MSDP peer connection with MSDP router 205.216.162.1. The CLI displays a message to indicate when the connection has been successfully closed.

Clearing the source active cache

To clear the entries from the Source Active cache, enter the **clear ip msdp sa-cache** command at the Privileged EXEC level of the CLI:

```
Brocade# clear ip msdp sa-cache
```

Syntax: **clear ip msdp sa-cache** [*<source-addr>* | *<group-addr>*]

The command in this example clears all the cache entries. Use the *<source-addr>* parameter to clear only the entries for a specified source. Use the *<group-addr>* parameter to clear only the entries for a specific group.

Clearing MSDP statistics

To clear MSDP statistics, enter the **clear ip msdp statistics** command at the Privileged EXEC level of the CLI:

```
Brocade# clear ip msdp statistics
```

Syntax: **clear ip msdp statistics** [*<ip-addr>*]

The command in this example clears statistics for all the peers. To clear statistics for a specific peer, enter the peer IP address.

Passive multicast route insertion

Passive Multicast Route Insertion (PMRI) enables a Layer 3 switch running PIM Sparse to create an entry for a multicast route (e.g., (S,G)), with no directly attached clients or when connected to another PIM router (transit network).

PMRI is critical for Service Providers wanting to deliver IP-TV services or multicast-based video services. Service Providers, who have transit networks, distribute multicast-based video services to other Service Providers, regardless of whether a client subscribes to a video service.

PMRI is enabled by default. To disable it, enter the following commands at the **router pim** level of the CLI.

```
Brocade(config)#router pim
Brocade#(config-pim-router)#no hardware-drop
```

Syntax: **[no] hardware-drop**

When PMRI is enabled, the **show ip pim mcache** command output displays the multicast cache entry along with a drop flag, indicating that the device is dropping packets in hardware. If the HW flag is set to 1 (HW=1), it implies that the packets are being dropped in hardware. If the HW flag is set to 0, (HW=0), it indicates that the packets are being processed in software. The following shows an example display output.

```

Brocade#show ip pim mcache
1 (10.10.10.18 226.0.1.56) in v10 (e1), cnt=2
Source is directly connected
Sparse Mode, RPT=0 SPT=1 REG=1 MSDP Adv=0 MSDP Create=0
fast=0 slow=0 pru=1 graft age drop
age=0s up-time=2m HW=1 L2-vidx=8191

```

DVMRP overview

Brocade routers provide multicast routing with the Distance Vector Multicast Routing Protocol (DVMRP) routing protocol. DVMRP uses Internet Group Membership Protocol (IGMP) to manage the IP multicast groups.

DVMRP is a broadcast and pruning multicast protocol that delivers IP multicast datagrams to its intended receivers. The receiver registers the interested groups using IGMP. DVMRP builds a multicast delivery tree with the sender forming the root. Initially, multicast datagrams are delivered to all nodes on the tree. Those leaves that do not have any group members send **prune messages** to the upstream router, noting the absence of a group. The upstream router maintains a prune state for this group for the given sender. A prune state is aged out after a given configurable interval, allowing multicasts to resume.

DVMRP employs **reverse path forwarding** and **pruning** to keep source specific multicast delivery trees with the minimum number of branches required to reach all group members. DVMRP builds a multicast tree for each source and destination host group.

NOTE

DVMRP is supported only on FastIron SX devices.

Initiating DVMRP multicasts on a network

Once DVMRP is enabled on each router, a network user can begin a video conference multicast from the server on R1. **Multicast Delivery Trees** are initially formed by source-originated multicast packets that are propagated to downstream interfaces as seen in [Figure 178](#). When a multicast packet is received on a DVMRP-capable router interface, the interface checks its DVMRP routing table to determine whether the interface that received the message provides the shortest path back to the source. If the interface does provide the shortest path, the interface forwards the multicast packet to adjacent peer DVMRP routers, except for the router interface that originated the packet. Otherwise, the interface discards the multicast packet and sends a prune message back upstream. This process is known as reverse path forwarding.

In [Figure 178](#), the root node (R1) is forwarding multicast packets for group 229.225.0.2 that it receives from the server to its downstream nodes, R2, R3, and R4. Router R4 is an intermediate router with R5 and R6 as its downstream routers. Because R5 and R6 have no downstream interfaces, they are leaf nodes.

The receivers in this example are those workstations that are resident on routers R2, R3, and R6.

Pruning a multicast tree

After the multicast tree is constructed, **pruning** of the tree will occur after IP multicast packets begin to traverse the tree.

As multicast packets reach leaf networks (subnets with no downstream interfaces), the local IGMP database checks for the recently arrived IP multicast packet address. If the local database does not contain the address (the address has not been learned), the router prunes (removes) the address from the multicast tree and no longer receives multicasts until the prune age expires.

In [Figure 178](#), Router 5 is a leaf node with no group members in its local database. Consequently, Router 5 sends a prune message to its upstream router. This router will not receive any further multicast traffic until the prune age interval expires.

FIGURE 178 Downstream broadcast of IP multicast packets from source host

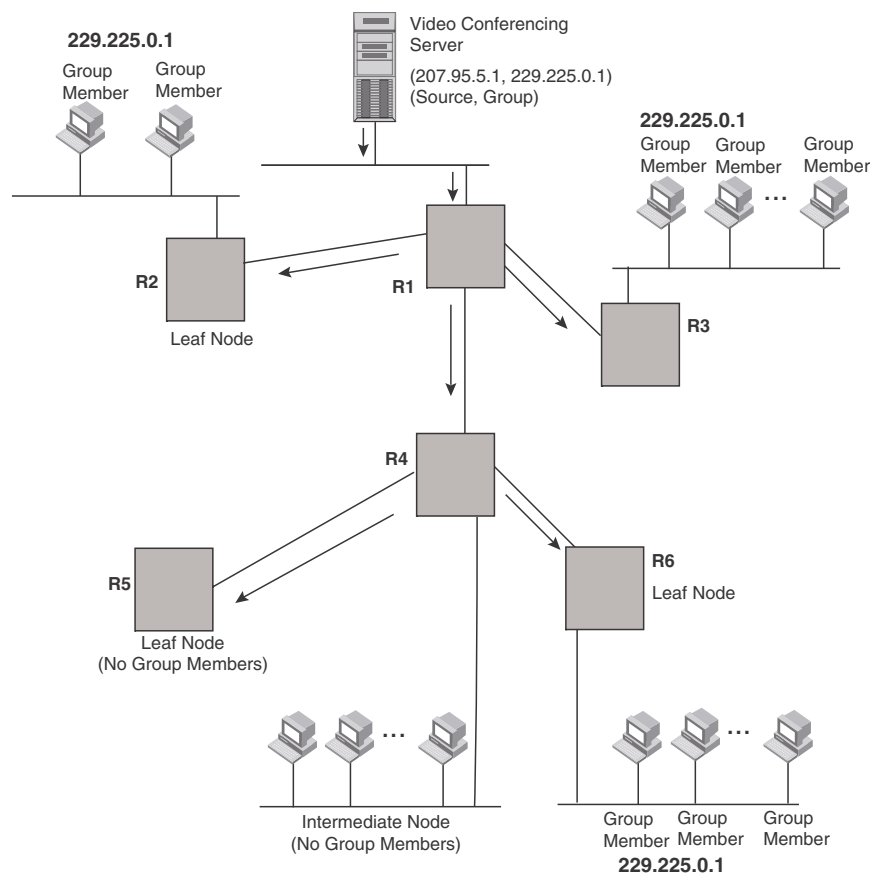
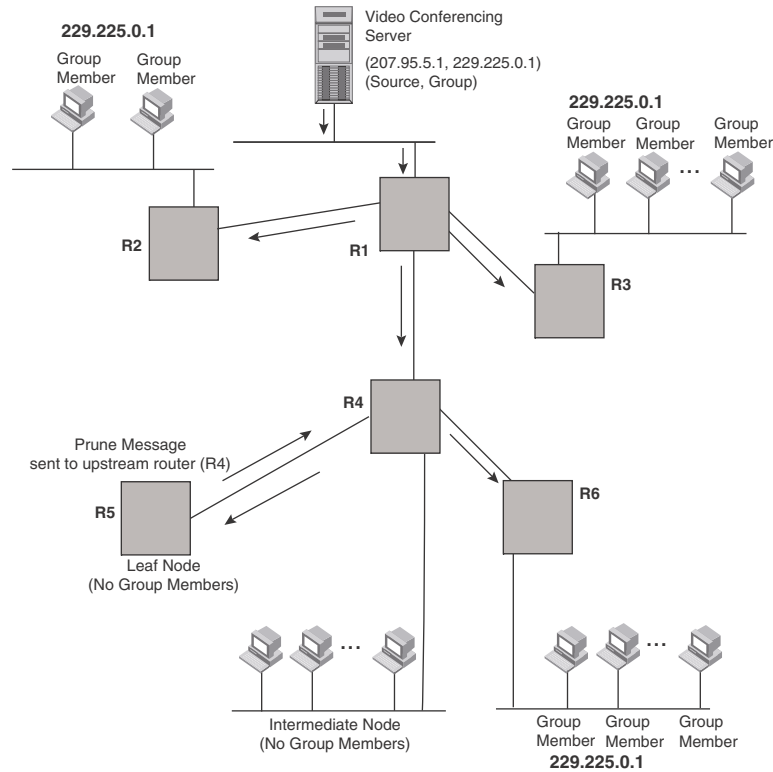


FIGURE 179 Pruning leaf nodes from a multicast tree



Grafts to a multicast tree

A DVMRP router restores pruned branches to a multicast tree by sending graft messages towards the upstream switch. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream switch.

In the example above, if a new 229.255.0.1 group member joins on switch S6, which had been pruned previously, a graft will be sent upstream to S4. Since the forwarding state for this entry is in a prune state, S4 sends a graft to S1. Once S4 has joined the tree, it and S6 will once again receive multicast packets.

You do not need to perform any configuration to maintain the multicast delivery tree. The prune and graft messages automatically maintain the tree.

DVMRP configuration on the Layer 3 switch and interface

Suppose you want to initiate the use of desktop video for fellow users on a sprawling campus network. All destination workstations have the appropriate hardware and software but the Layer 3 switches that connect the various buildings need to be configured to support Distance Vector Multicast Protocol (DVMRP) multicasts from the designated video conference server as seen in [Figure 178](#).

DVMRP is enabled on each of the Brocade Layer 3 switches shown in [Figure 178](#), on which multicasts are expected. You can enable DVMRP on each Layer 3 switch independently or remotely from one Layer 3 switch by a Telnet connection. Follow the same steps for each Layer 3 switch.

Globally enabling and disabling DVMRP

To globally enable DVMRP, enter the **router dvmrp** command.

```
Router1(config)#router dvmrp
```

Syntax: [no] router dvmrp

The behavior of the [no] **router dvmrp** command is as follows:

- Entering a **router dvmrp** command to enable DVMRP does not require a software reload.
- Entering a **no router dvmrp** command removes all configuration for PIM multicast on a Layer 3 switch (**router pim** level) only.

Globally enabling or disabling DVMRP without deleting multicast configuration

As stated above enter **no router dvmrp** removed PIM configuration. If you want to disable or enable DVMRP without removing PIM configuration, enter the following commands.

```
Brocade(config)#router dvmrp
Brocade(config-pim-router)#disable-dvmrp
```

Syntax: [no] disable-dvmrp

Use the [no] version of the command to re-enable DVMRP.

Enabling DVMRP on an interface

After globally enabling DVMRP on a Layer 3 switch, enable it on each interface that will support the protocol.

To enable DVMRP on S1 and interface 3, enter the following commands.

```
Router1(config)#router dvmrp
Router1(config-dvmrp-router)#int e 3
Router1(config-if-3)#ip dvmrp
```

Modifying DVMRP global parameters

DVMRP global parameters come with preset values. The defaults work well in most networks, but you can modify the following global parameters if you need to:

- Neighbor timeout
- Route expire time
- Route discard time
- Prune age
- Graft retransmit time
- Probe interval
- Report interval
- Trigger interval

- Default route

Modifying neighbor timeout

The neighbor timeout specifies the period of time that a router will wait before it defines an attached DVMRP neighbor router as down. Possible values are 40 – 8000 seconds. The default value is 180 seconds.

To modify the neighbor timeout value to 100, enter the **nbr <num>** command.

```
Brocade(config-dvmrp-router)#nbr 100
```

Syntax: **nbr-timeout <40-8000>**

The default is 180 seconds.

Modifying Route Expires Time

The Route Expire Time defines how long a route is considered valid in the absence of the next route update. Possible values are from 20 – 4000 seconds. The default value is 200 seconds.

To modify the route expire setting to 50, enter the **route-expire-timeout <num>** command.

```
Brocade(config-dvmrp-router)#route-expire-timeout 50
```

Syntax: **route-expire-timeout <20-4000>**

Modifying Route Discard Time

The Route Discard Time defines the period of time before a route is deleted. Possible values are from 40 – 8000 seconds. The default value is 340 seconds.

To modify the route discard setting to 150, enter the **route-discard-timeout <num>** command.

```
Brocade(config-dvmrp-router)#route-discard-timeout 150
```

Syntax: **route-discard-timeout <40-8000>**

Modifying Prune Age

The Prune Age defines how long a prune state will remain in effect for a source-routed multicast tree. After the prune age period expires, flooding will resume. Possible values are from 20 – 3600 seconds. The default value is 180 seconds.

To modify the prune age setting to 150, enter the **prune <num>** command.

```
Brocade(config-dvmrp-router)#prune 25
```

Syntax: **prune-age <20-3600>**

Modifying Graft Retransmit Time

The Graft Retransmit Time defines the initial period of time that a router sending a graft message will wait for a graft acknowledgement from an upstream router before re-transmitting that message.

Subsequent retransmissions are sent at an interval twice that of the preceding interval. Possible values are from

5 – 3600 seconds. The default value is 10 seconds.

To modify the setting for graft retransmit time to 120, enter the **graft <num>** command.

```
Brocade(config-dvmrp-router)#graft 120
```

Syntax: **graft-retransmit-time** <5-3600>

Modifying Probe Interval

The Probe Interval defines how often neighbor probe messages are sent to the ALL-DVMRP-ROUTERS IP multicast group address. A router probe message lists those neighbor DVMRP routers from which it has received probes. Possible values are from 5 – 30 seconds. The default value is 10 seconds.

To modify the probe interval setting to 10, enter the **probe <num>** command.

```
Brocade(config-dvmrp-router)#probe 10
```

Syntax: **probe-interval** <5-30>

Modifying Report Interval

The Report Interval defines how often routers propagate their complete routing tables to other neighbor DVMRP routers. Possible values are from 10 – 2000 seconds. The default value is 60 seconds.

To support propagation of DVMRP routing information to the network every 90 seconds, enter the **report <num>** command.

```
Brocade(config-dvmrp-router)#report 90
```

Syntax: **report-interval** <10-2000>

Modifying Trigger Interval

The Trigger Interval defines how often trigger updates, which reflect changes in the network topology, are sent. Example changes in a network topology include router up or down or changes in the metric. Possible values are from 5 – 30 seconds. The default value is 5 seconds.

To support the sending of trigger updates every 20 seconds, enter the **trigger-interval <num>** command.

```
Brocade(config-dvmrp-router)#trigger-interval 20
```

Syntax: **trigger-interval** <5-30>

Modifying default route

To define the default gateway for DVMRP, enter the following command.

```
Brocade(config-dvmrp-router)#default-gateway 192.35.4.1
```

Syntax: **default-gateway** <ip-addr>

Modifying DVMRP interface parameters

DVMRP global parameters come with preset values. The defaults work well in most networks, but you can modify the following interface parameters if you need to:

- TTL
- Metric
- Advertising

Modifying the TTL

The time to live (TTL) value defines the minimum value required in a packet in order for the packet to be forwarded out the interface. For example, if the TTL for an interface is set at 10 it means that only those packets with a TTL value of 10 or more are forwarded. Likewise, if an interface is configured with a TTL Threshold value of 1, all packets received on that interface are forwarded. Possible values are from 1 – 64. The default value is 1.

To set a TTL of 64, enter the following commands.

```
Brocade(config)#int e 1/4  
Brocade(config-if-1/4)#ip dvmrp ttl 60
```

Syntax: ttl-threshold <1-64>

Modifying the metric

The router uses the metric when establishing reverse paths to some networks on directly attached interfaces. Possible values are from 1 – 31 hops. The default is 1.

NOTE

This command is not supported on Brocade Layer 2 Switches.

To set a metric of 15 for a DVMRP interface, enter the following commands.

```
Brocade(config)#interface 3/5  
Brocade(config-if-3/5)#ip dvmrp metric 15
```

Syntax: ip dvmrp metric <1-31>

Enabling advertising

You can turn the advertisement of a local route on (enable) or off (disable) on the interface. By default, advertising is enabled.

To enable advertising on an interface, enter the following commands.

```
Brocade(config-if-1/4)#ip dvmrp advertise-local on
```

Syntax: advertise-local on | off

Displaying information about an upstream neighbor device

You can view information about the upstream neighbor device for a given source IP address for IP PIM packets. The software uses the IP route table or multicast route table to lookup the upstream neighbor device.

The following shows example messages that the Brocade device can display with this command.

```
Brocade#show ip dvmrp rpf 1.1.20.2
directly connected or through an L2 neighbor
Brocade#show ip dvmrp rpf 1.2.3.4
no route
Brocade#show ip dvmrp rpf 1.10.10.24
upstream neighbor=1.1.20.1 on v21 using ip route
```

Syntax: `show ip dvmrp rpf <IP address>`

where `<IP address>` is a valid source IP address

NOTE

If there are multiple equal cost paths to the source, the `show ip dvmrp rpf` command output may not be accurate. If your system has multiple equal cost paths, use the command `show ip dvmrp mcache` to view information about the upstream neighbor.

```
Brocade(config)#router dvmrp
Brocade(config-pim-router)#disable-dvmrp
```

Syntax: `[no] disable-dvmrp`

Use the `[no]` version of the command to re-enable DVMRP.

Enabling DVMRP on an interface

After globally enabling DVMRP on a Layer 3 switch, enable it on each interface that will support the protocol.

To enable DVMRP on S1 and interface 3, enter the following.

```
Router1(config)#router dvmrp
Router1(config-dvmrp-router)#int e 3
Router1(config-if-3)#ip dvmrp
```

IP tunnel configuration

IP tunnels are used to send traffic through routers that do not support IP multicasting. IP Multicast datagrams are encapsulated within an IP packet and then sent to the remote address. Routers that are not configured for IP Multicast route the packet as a normal IP packet. When the IP Multicast router at the remote end of the tunnel receives the packet, the router strips off the IP encapsulation and forwards the packet as an IP Multicast packet.

NOTE

An IP tunnel must have a remote IP interface at each end. Also, for IP tunneling to work, the remote routers must be reachable by an IP routing protocol.

NOTE

Multiple tunnels configured on a router cannot share the same remote address.

Example

To configure an IP tunnel as seen in [Figure 178](#), enter the IP tunnel destination address on an interface of the router.

To configure an IP address on Router A, enter the following commands.

```
FastIron(config)#int e1
FastIron(config-if-1)#ip tunnel 192.3.45.6
```

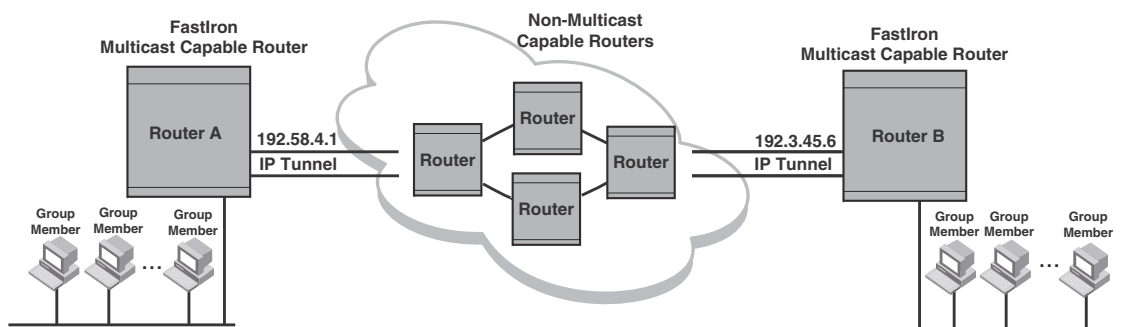
NOTE

The IP tunnel address represents the configured IP tunnel address of the destination router. In the case of Router A, its destination router is Router B. Router A is the destination router of Router B.

For router B, enter the following.

```
FastIron(config-if-1)#ip tunnel 192.58.4.1
```

FIGURE 180 IP in IP tunneling on multicast packets in a unicast network



Using ACLs to control multicast features

You can use ACLs to control the following multicast features:

- Limit the number of multicast groups that are covered by a static rendezvous point (RP)
- Control which multicast groups for which candidate RPs sends advertisement messages to bootstrap routers
- Identify which multicast group packets will be forwarded or blocked on an interface

Using ACLs to limit static RP groups

You can limit the number of multicast groups covered by a static RP using standard ACLs. In the ACL, you specify the group to which the RP address applies. The following examples set the RP address to be applied to multicast groups with some minor variations.

To configure an RP that covers multicast groups in 239.255.162.x, enter commands such as the following.

```
Brocade(config)#access-list 2 permit 239.255.162.0 0.0.0.255
Brocade(config)#router pim
Brocade(config-pim-router)#rp-address 43.43.43.1 2
```

To configure an RP that covers multicast groups in the 239.255.162.x range, except the 239.255.162.2 group, enter commands such as the following.

```
Brocade(config)#access-list 5 deny host 239.255.162.2
Brocade(config)#access-list 5 permit 239.255.0.0 0.0.255.255
Brocade(config)#router pim
Brocade(config-pim-router)#bsr-candidate ve 43 32 100
Brocade(config-pim-router)#rp-candidate ve 43
Brocade(config-pim-router)#rp-address 99.99.99.5 5
```

To configure an RP for multicast groups using the override switch, enter commands such as the following.

```
Brocade(config)#access-list 44 permit 239.255.162.0 0.0.0.255
Brocade(config)#router pim
Brocade(config-pim-router)#rp-address 43.43.43.1
Brocade(config-pim-router)#rp-address 99.99.99.5 44 override
```

Syntax: [no] **rp-address** <ip-address> [<access-list-num>] [**override**]

The access-list-num parameter is the number of the standard ACL that will filter the multicast group.

NOTE

Extended ACLs cannot be used to limit static RP groups.

The **override** parameter directs the Layer 3 switch to ignore the information learned by a BSR if there is a conflict between the RP configured in this command and the information that is learned by the BSR. An RP address learned dynamically from PIM Bootstrap protocol takes precedence over static RP configuration unless the override parameter is used.

You can use the **show ip pim rp-set** command to display the ACLs used to filter the static RP groups.

Example

```
Brocade#show ip pim rp-set
Group address      Static-RP-address  Override
-----
Access-List 44    99.99.99.5         On
Number of group prefixes Learnt from BSR: 1
Group prefix = 224.0.0.0/4 #RPs: 1
RP 1: 43.43.43.1 priority=0 age=0
```

In the example above, the display shows the following information:

- The Group Address table shows the static RP address that is covered by the access list, and whether or not the override parameter has been enabled.
- The Group prefix line shows the multicast group prefix for the static RP.
- The RP #line shows the configured IP address of the RP candidate.

Enter the **show ip pim rp-map** command to display the group-to-RP mapping.

```
router(config)# show ip pim rp-map
Number of group-to-RP mappings: 5
Group address RP address
-----
1  230.0.0.1 100.1.1.1
2  230.0.0.2 100.1.1.1
3  230.0.0.3 100.1.1.1
4  230.0.0.4 100.1.1.1
5  230.0.0.5 100.1.1.1
```

The display shows the multicast group addresses covered by the RP candidate and the IP address of the RP for the listed multicast group. In the example above, you see the following:

- The first three lines show the multicast group addresses that are covered by the RP candidate.
- The last three lines show the multicast group addresses covered by the static RP.

Using ACLs to limit PIM RP candidate advertisement

You can use standard ACLs to control the groups for which the candidate RP will send advertisement messages to the bootstrap router. For example, ACL 5 can be configured to be applied to the multicast groups within the IP address 239.x.x.x range. You can configure the Layer 3 switch to advertise itself as a candidate RP to the bootstrap router only for groups in the range of 239.x.x.x. Enter commands such as the following.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#ip address 99.99.99.5 255.255.255.0
Brocade(config-if-1/1)#ip pim-sparse
Brocade(config-if-1/1)#exit
Brocade(config)#access-list 5 deny host 239.255.162.2
Brocade(config)#access-list 5 permit 239.0.0.0 0.0.255.255
Brocade(config)#router pim
Brocade(config-pim-router)#bsr-candidate ethernet 1/1 32 100
Brocade(config-pim-router)#rp-candidate ethernet 1/1 group-list 5
```

The example above shows a configuration for an Ethernet interface. To configure ACLs that are applied to a virtual routing interface, enter commands such as the following.

```
Brocade(config)#interface ve 16
Brocade(config-vif-16)#ip address 16.16.16.1 255.255.255.0
Brocade(config-vif-16)#ip pim-sparse
Brocade(config-vif-16)#exit
Brocade(config)#access-list 5 deny host 239.255.162.2
Brocade(config)#access-list 5 permit 239.255.0.0 0.0.255.255
Brocade(config)#router pim
Brocade(config-pim-router)#bsr-candidate ve 16 32 100
Brocade(config-pim-router)#rp-candidate ve 16 group-list 5
```

To configure ACLs that are applied to a loopback interface, enter commands such as the following.

```
Brocade(config)#interface loopback 1
Brocade(config-lbif-1)#ip address 88.88.88.8 255.255.255.0
Brocade(config-lbif-1)#ip pim-sparse
Brocade(config-lbif-1)#exit
```

```

Brocade(config)#access-list 5 deny host 239.255.162.2
Brocade(config)#access-list 5 permit 239.255.0.0 0.0.255.255
Brocade(config)#router pim
Brocade(config-pim-router)#bsr-candidate loopback 1 32 100
Brocade(config-pim-router)#rp-candidate loopback 1 group-list 5

```

Syntax: [no] **rp-candidate ethernet** [<slotnum> / <portnum> | **loopback** <num> | **ve** <num>]
[group-list <access-list-num>]

The <slotnum> parameter is required on chassis devices.

The <portnum> | **loopback** <num> | **ve** <num> parameter specifies the interface. The Layer 3 switch will advertise the specified interface IP address as a candidate RP:

- Enter **ethernet** [<slotnum> / <portnum>] for a physical interface (port).
- Enter **ve** <num> for a virtual interface.
- Enter **loopback** <num> for a loopback interface.

The **group-list** <access-list-num> indicates that a standard ACL is used to filter for which multicast group the advertisement will be made.

NOTE

Extended ACLs cannot be used for group-list.

Disabling CPU processing for select multicast groups

In IPv4 multicast, Brocade Layer 3 switches do not forward multicast packets with destination addresses in the range between 224.0.0.0 and 224.0.0.255. These group addresses are reserved for various routing protocols. By default, packets destined to these groups are processed by the CPU. However, when a large number of packets for these groups are received by the Brocade device all at once, CPU resources may be overloaded. To alleviate the load on the CPU, you could disable CPU processing of packets for these groups. When applied, this feature protects the CPU from traffic sent to IPV4 multicast addresses in the range 224.0.0.1 - 224.0.0.254, and instead floods these packets in hardware within the incoming VLAN.

This feature can be applied on a VLAN or a VLAN-group. If applied on a VLAN, traffic received on a port of the VLAN will be flooded to all other ports of the VLAN. If applied on a VLAN-group, traffic will be flooded only at the individual VLAN level. Once this feature is applied on a VLAN or VLAN-group, ports that are statically or dynamically added to the VLAN or VLAN-group will inherit the configuration. Likewise, ports that are statically or dynamically removed from the VLAN or VLAN-group will drop the configuration.

This feature can be enabled for packets destined to a multicast group or set of groups in the range 224.0.0.1 – 224.0.0.254, except for the reserved multicast addresses listed in the following table.

TABLE 269 Reserved multicast addresses

Multicast address	Reserved for...
224.0.0.1	all nodes
224.0.0.2	PIM
224.0.0.3	DVMRP
224.0.0.4	DVMRP
224.0.0.5	OSPF

TABLE 269 Reserved multicast addresses (Continued)

Multicast address	Reserved for...
224.0.0.6	OSPF
224.0.0.9	RIP V2
224.0.0.13	PIM V2
224.0.0.18	VRRP
224.0.0.22	IGMP V3 reports

CLI command syntax to disable CPU processing

To disable CPU processing for selective multicast groups, enter commands such as the following.

```
Brocade# config t
Brocade(config)# vlan 5
Brocade(config-vlan-5)# disable multicast-to-cpu 224.0.0.5
Brocade(config-vlan-5)# disable multicast-to-cpu 224.0.0.14 224.0.0.230
Brocade(config-vlan-5)# vlan 10
Brocade(config-vlan-10)# disable multicast-to-cpu 224.0.0.23
Brocade(config-vlan-10)# vlan 20
Brocade(config-vlan-20)# disable multicast-to-cpu 224.0.0.50 224.0.0.140
```

Syntax: [no] **disable multicast-to-cpu** <multicast group address> [<multicast group range end address>]

The <multicast group address> must be in the range 224.0.0.1 - 224.0.0.254, but cannot be one of the reserved multicast addresses listed in [Table 269](#) on page 1591.

Viewing disabled multicast addresses

To display disabled multicast addresses for all configured VLANs, enter the **show disabled-multicast-to-cpu** command. The following shows an example display.

```
Brocade# show disabled-multicast-to-cpu

Disabled multicast addresses to cpu for PORT-VLAN 5 :
  224.0.0.5
  224.0.0.14 to 224.0.0.230

Disabled multicast addresses to cpu for PORT-VLAN 10 :
  224.0.0.23

Disabled multicast addresses to cpu for PORT-VLAN 20 :
  224.0.0.50 to 224.0.0.140
```

To display disabled multicast addresses for a particular VLAN, include the VLAN ID with the **show disabled-multicast-to-cpu** command. The following shows an example display.

```
Brocade# show disabled-multicast-to-cpu 5
Disabled multicast addresses to cpu for PORT-VLAN 5 :
  224.0.0.5
  224.0.0.14 to 224.0.0.230
```

Syntax: `show disabled-multicast-to-cpu [<vlan-id>]`

For <vlan-id>, enter a valid VLAN ID. Note that each VLAN must have at least one port added to it.

Configuring a static multicast route

Static multicast routes allow you to control the network path used by multicast traffic. Static multicast routes are especially useful when the unicast and multicast topologies of a network are different. You can avoid the need to make the topologies similar by instead configuring static multicast routes.

NOTE

This feature is not supported for DVMRP.

You can configure more than one static multicast route. The Layer 3 switch always uses the most specific route that matches a multicast source address. Thus, if you want to configure a multicast static route for a specific multicast source and also configure another multicast static route for all other sources, you can configure two static routes as shown in the examples below.

To add static routes to multicast router A (refer to [Figure 181](#)), enter commands such as the following.

```
PIMRouterA(config)#ip mroute 1 207.95.10.0 255.255.255.0 interface ethernet 1/2
distance 1
PIMRouterA(config)#ip mroute 2 0.0.0.0 0.0.0.0 interface ethernet 2/3 distance 1
PIMRouterA(config)#write memory
```

Syntax: `mroute <route-num> <ip-addr> interface ethernet [<slotnum>/]<portnum> | ve <num> [distance <num>]`

or

Syntax: `mroute <route-num> <ip-addr> rpf_address <rpf-num>`

The <route-num> parameter specifies the route number.

The <ip-addr> command specifies the PIM source for the route.

NOTE

In IP multicasting, a route is handled in terms of its source, rather than its destination.

You can use the **ethernet** [<slotnum>/]<portnum> parameter to specify a physical port or the **ve** <num> parameter to specify a virtual interface.

NOTE

The **ethernet** [<slotnum>/]<portnum> parameter does not apply to PIM SM.

The **distance** <num> parameter sets the administrative distance for the route. When comparing multiple paths for a route, the Layer 3 switch prefers the path with the lower administrative distance.

NOTE

Regardless of the administrative distances, the Layer 3 switch always prefers directly connected routes over other routes.

The **rpf_address** <rpf-num> parameter specifies an RPF number.

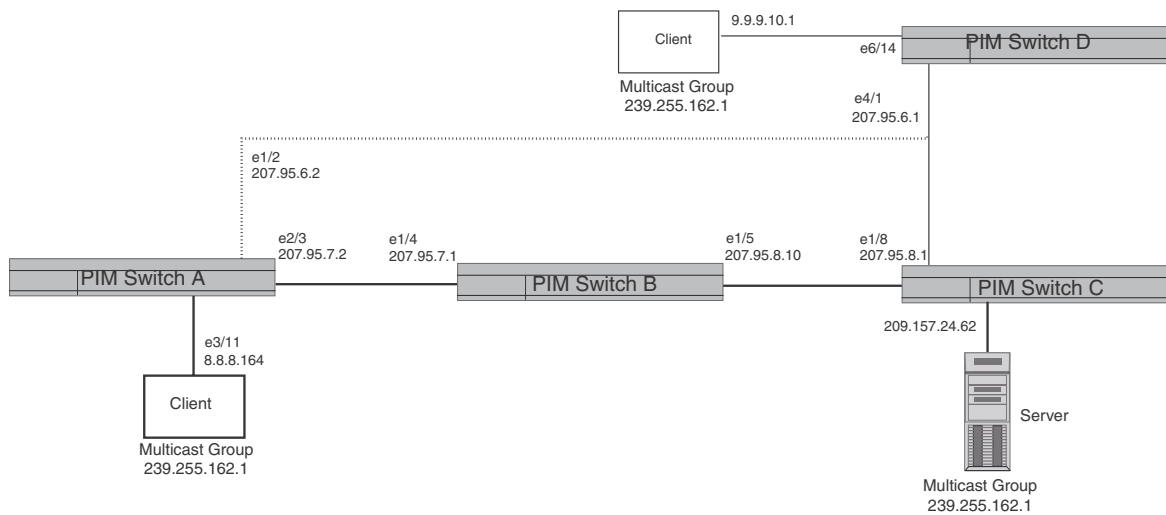
Configuring a static multicast route

The example above configures two static multicast routes. The first route is for a specific source network, 207.95.10.0/24. If the Layer 3 switch receives multicast traffic for network 207.95.10.0/24, the traffic must arrive on port 1/2. The second route is for all other multicast traffic. Traffic from multicast sources other than 207.95.10.0/24 must arrive on port 2/3.

Figure 181 shows an example of an IP Multicast network. The two static routes configured in the example above apply to this network. The commands in the example above configure PIM router A to accept PIM packets from 207.95.10.0/24 when they use the path that arrives at port 1/2, and accept all other PIM packets only when they use the path that arrives at port 2/3.

The distance parameter sets the administrative distance. This parameter is used by the software to determine the best path for the route. Thus, to ensure that the Layer 3 switch uses the default static route, assign a low administrative distance value. When comparing multiple paths for a route, the Layer 3 switch prefers the path with the lower administrative distance.

FIGURE 181 Example of multicast static routes



To add a static route to a virtual interface, enter commands such as the following.

```
Brocade(config)# ip mroute 3 0.0.0.0 0.0.0.0 int ve 1 distance 1
Brocade(config)# write memory
```


Displaying the multicast configuration for another multicast router

The Brocade implementation of Mrinfo is based on the DVMRP Internet draft by T. Pusateri, but applies to PIM and not to DVMRP. To display the PIM configuration of another PIM router, use the following commands.

NOTE

This feature is not supported for DVMRP.

To display another PIM router PIM configuration, enter a command such as the following.

```
Brocade#mrinfo 207.95.8.1
207.95.8.1 -> 207.95.8.10 [PIM/0 /1]
207.95.10.2 -> 0.0.0.0 [PIM/0 /1 /leaf]
209.157.25.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
209.157.24.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
207.95.6.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
128.2.0.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
```

Syntax: `mrinfo <ip-addr>`

The `<ip-addr>` parameter specifies the IP address of the PIM router.

The output in this example is based on the PIM group shown in [Figure 174](#) on page 1532. The output shows the PIM interfaces configured on PIM router C (207.95.8.1). In this example, the PIM router has six PIM interfaces. One of the interfaces goes to PIM router B. The other interfaces go to leaf nodes, which are multicast end nodes attached to the router PIM interfaces. (For simplicity, the figure shows only one leaf node.)

When the arrow following an interface in the display points to a router address, this is the address of the next hop PIM router on that interface. In this example, PIM interface 207.95.8.1 on PIM router 207.95.8.1 is connected to PIM router 207.95.8.10. The connection can be a direct one or can take place through non-PIM routers. In this example, the PIM routers are directly connected.

When the arrow following an interface address points to zeros (0.0.0.0), the interface is not connected to a PIM router. The interface is instead connected to a leaf node.

NOTE

This display shows the PIM interface configuration information, but does not show the link states for the interfaces.

The information in brackets indicates the following:

- The multicast interface type (always PIM; this display is not supported in DVMRP)
- The Time-to-Live (TTL) for the interface.
- The metric for the interface
- Whether the interface is connected to a leaf node ("leaf" indicates a leaf node and blank indicates another PIM router)

For example, the information for the first interface listed in the display is "PIM/0 /1". This information indicates that the interface is a PIM interface, has a TTL of 0, and a metric of 1. The interface is not a leaf node interface and thus is an interface to another PIM router.

The information for the second interface in the display is “PIM/0 /1/leaf”. This information indicates that the interface is a PIM interface, has a TTL of 0 and a metric of 1, and is connected to a leaf node.

IGMP V3

The Internet Group Management Protocol (IGMP) allows an IPV4 interface to communicate IP Multicast group membership information to its neighboring routers. The routers in turn limit the multicast of IP packets with multicast destination addresses to only those interfaces on the router that are identified as IP Multicast group members. This release introduces the support of IGMP version 3 (IGMP V3) on Layer 3 switches.

In IGMP V2, when a router sent a query to the interfaces, the clients on the interfaces respond with a membership report of multicast groups to the router. The router can then send traffic to these groups, regardless of the traffic source. When an interface no longer needs to receive traffic from a group, it sends a leave message to the router which in turn sends a group-specific query to that interface to see if any other clients on the same interface is still active.

In contrast, IGMP V3 provides selective filtering of traffic based on traffic source. A router running IGMP V3 sends queries to every multicast enabled interface at the specified interval. These queries determine if any interface wants to receive traffic from the router. The queries include the IP address of the traffic source (S) or the ID of the multicast group (G, or both).

The interfaces respond to these queries by sending a membership report that contains one or more of the following records that are associated with a specific group:

- Current-State Record that indicates from which sources the interface wants to receive and not receive traffic. The record contains source address of interfaces and whether or not traffic will be received or included (IS_IN) or not received or excluded (IS_EX) from that source.
- Filter-mode-change record. If the interface changes its current state from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if an interface current state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.

IGMP V2 Leave report is equivalent to a TO_IN(empty) record in IGMP V3. This record means that no traffic from this group will be received regardless of the source.

An IGMP V2 group report is equivalent to an IS_EX(empty) record in IGMP V3. This record means that all traffic from this group will be received regardless of source.

- Source-List-Change Record. If the interface wants to add or remove traffic sources from its membership report, the membership report can have an ALLOW record, which contains a list of new sources from which the interface wishes to receive traffic. It can also contain a BLOCK record, which lists current traffic sources from which the interfaces want to stop receiving traffic.

In response to membership reports from the interfaces, the router sends a Group-Specific or a Group-and-Source Specific query to the multicast interfaces. Each query is sent three times with a one-second interval in between each transmission to ensure the interfaces receive the query. For example, a router receives a membership report with a Source-List-Change record to block old sources from an interface. The router sends Group-and-Source Specific Queries to the source and group (S,G) identified in the record. If none of the interfaces is interested in the (S,G), it is removed from (S,G) list for that interface on the router.

Each IGMP V3-enabled router maintains a record of the state of each group and each physical port within a virtual routing interface. This record contains the group, group-timer, filter mode, and source records information for the group or interface. Source records contain information on the source address of the packet and source timer. If the source timer expires when the state of the group or interface is in Include mode, the record is removed.

Default IGMP version

IGMP V3 is available on Brocade devices; however, the devices are shipped with IGMP V2 enabled. You must enable IGMP V3 globally or per interface.

Also, you must specify what version of IGMP you want to run on a device globally, on each interface (physical port or virtual routing interface), and on each physical port within a virtual routing interface. If you do not specify an IGMP version, IGMP V2 will be used.

Compatibility with IGMP V1 and V2

Different multicast groups, interfaces, and routers can run their own version of IGMP. Their version of IGMP is reflected in the membership reports that the interfaces send to the router. Routers and interfaces must be configured to recognize the version of IGMP you want them to process.

An interface or router sends the queries and reports that include its IGMP version specified on it. It may recognize a query or report that has a different version, but it may not process them. For example, an interface running IGMP V2 can recognize IGMP V3 packets, but cannot process them. Also, a router running IGMP V3 can recognize and process IGMP V2 packet, but when that router sends queries to an IGMP V2 interface, the host on that interface may not recognize the IGMP V3 queries. The interface or router does not automatically downgrade the IGMP version running on them to avoid version deadlock.

If an interface continuously receives queries from routers that are running versions of IGMP that are different from what is on the interface, the interface logs warning messages in the syslog every five minutes. Reports sent by interfaces to routers that contain different versions of IGMP do not trigger warning messages; however, you can see the versions of the packets using the **show ip igmp traffic** command.

The version of IGMP can be specified globally, per interface (physical port or virtual routing interface), and per physical port within a virtual routing interface. The IGMP version set on a physical port within a virtual routing interface supersedes the version set on a physical or virtual routing interface. Likewise, the version on a physical or virtual routing interface supersedes the version set globally on the device. The sections below present how to set the version of IGMP.

Globally enabling the IGMP version

To globally identify the IGMP version on a Brocade device, enter the **ip igmp <version-number>** command.

```
Brocade(config)#ip igmp version 3
```

Syntax: **ip igmp version** <version-number>

Enter 1, 2, or 3 for <version-number>. Version 2 is the default version.

Enabling the IGMP version per interface setting

To specify the IGMP version for a physical port, enter a command such as the following.

```
Brocade(config)#interface eth 1/5
Brocade(config-if-1/5)#ip igmp version 3
```

To specify the IGMP version for a virtual routing interface on a physical port, enter a command such as the following.

```
Brocade(config)#interface ve 3
Brocade(config-vif-1) ip igmp version 3
```

Syntax: [no] ip igmp version <version-number>

Enter 1, 2, or 3 for <version-number>. Version 2 is the default version.

Enabling the IGMP version on a physical port within a virtual routing interface

To specify the IGMP version recognized by a physical port that is a member of a virtual routing interface, enter a command such as the following.

```
Brocade(config)#interface ve 3
Brocade(config-vif-3)#ip igmp version 2
Brocade(config-vif-3)#ip igmp port-version 3 e1/3-e1/7 e2/9
```

In this example, the second line sets IGMP V2 on virtual routing interface 3. However, the third line set IGMP V3 on ports 1/3 through 1/7 and port e2/9. All other ports in this virtual routing interface are configured with IGMP V2.

Syntax: ip igmp port-version <version-number> ethernet [<slotnum>/]<port-number>

Enter 1, 2, or 3 for <version-number>. IGMP V2 is the default version.

The **ethernet** <port-number> parameter specifies which physical port within a virtual routing interface is being configured. If you are entering this command on a chassis device, specify the slot number as well as the port number.

Enabling membership tracking and fast leave

IGMP V3 provides membership tracking and fast leave to clients. In IGMP V2, only one client on an interface needs to respond to a router queries; therefore, some of the clients may be invisible to the router, making it impossible for the router to track the membership of all clients in a group. Also, when a client leaves the group, the router sends group specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the router waits three seconds before it stops the traffic.

IGMP V3 contains the tracking and fast leave feature that you enable on virtual routing interfaces. Once enabled, all physical ports on that virtual routing interface will have the feature enabled. IGMP V3 requires all clients to respond to general and group specific queries so that all clients on an interface can be *tracked*. *Fast leave* allows clients to leave the group without the three second waiting period, if the following conditions are met:

- If the interface, to which the client belongs, has IGMP V3 clients only. Therefore, all physical ports on a virtual routing interface must have IGMP V3 enabled and no IGMP V1 or V2 clients can be on the interface. (Although IGMP V3 can handle V1 and V2 clients, these two clients cannot be on the interface in order for fast leave to take effect.)
- No other client on the interface is receiving traffic from the group to which the client belongs.

Every group on the physical interface of a virtual routing interface keeps its own tracking record. However, it can track group membership only; it cannot track by (source, group).

For example, two clients (Client A and Client B) belong to group1 but each is receiving traffic streams from different sources. Client A receives a stream from (source_1, group1) and Client B receives it from (source_2, group1). The router still waits for three seconds before it stops the traffic because the two clients are in the same group. If the clients are in different groups, then the three second waiting period is not applied and traffic is stopped immediately. The **show ip igmp group tracking** command displays that clients in a group that are being tracked.

If a client sends a leave message, the client is immediately removed from the group. If a client does not send a report during the the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

To enable the tracking and fast leave feature, enter commands such as the following.

```
Brocade(config)#interface ve 13
Brocade(config-vif-13)#ip igmp tracking
```

Syntax: **ip igmp tracking**

Setting the query interval

The IGMP query interval period defines how often a router will query an interface for group membership.

To modify the default value for the IGMP query interval, enter the **ip igmp query-interval <num>** command.

```
Brocade(config)#ip igmp query-interval 120
```

Syntax: **ip igmp query-interval <num>**

The <num> variable specifies the IGMP query interval in number of seconds. Enter a value from 10 through 3600. The default value is 125.

Setting the group membership time

The group membership time defines how long a group will remain active on an interface in the absence of a group report.

To define an IGMP membership time of 240 seconds, enter the **ip igmp group-membership-time <num>** command.

```
Brocade(config)#ip igmp group-membership-time 240
```

Syntax: **ip igmp group-membership-time <num>**

The <num> variable specifies the IGMP group membership time in number of seconds. Enter a value from 20 through 7200 seconds. The value you enter must be a little more than two times the query interval ($2 * \text{query-interval} + 10$). The default value is 260.

Setting the maximum response time

Maximum response time defines how long the Layer 3 switch will wait for an IGMP (V1 and V2) response from an interface before concluding that the group member on that interface is down, and then removing the interface from the group.

To change the IGMP maximum response time, enter the **ip igmp max-response-time <num>** command at the global CONFIG level of the CLI.

```
Brocade(config)#ip igmp max-response-time 8
```

Syntax: [no] ip igmp max-response-time <num>

The <num> parameter specifies the IGMP maximum response time in number of seconds. Enter a value from 1 through 10. The default is 10.

IGMP V3 and source-specific multicast protocols

Enabling IGMP V3 enables source specific multicast (SSM) filtering for DVMRP and PIM Dense (PIM-DM) for multicast group addresses in the 224.0.1.0 through 239.255.255.255 address range. However, if PIM Sparse is used as the multicast protocol, the SSM protocol should be enabled if you want to filter unwanted traffic before the Shortest Path Tree protocol switchover occurs for groups in the 232/8 range. Not configuring the SSM protocol in PIM Sparse may cause the switch or router to leak unwanted packets with the same group, but containing undesired sources, to clients. After SPT switch over, the leak stops and source specific multicast works correctly even without configuring the SSM protocol.

If the SSM protocol is not enabled and before the SPT switchover, the multicast router creates one (*, G) entry for the entire multicast group, which can have many sources. If the SSM protocol is enabled, one (S,G) entry is created for every member of the multicast group, even for members with non-existent traffic. For example, if there are 1,000 members in the group, 1,000 (S,G) entries will be created. Therefore, enabling the SSM protocol for PIM-SM requires more resources than leaving the protocol disabled.

Enabling SSM

To enable the SSM protocol on a Brocade device running PIM-SM, enter a command such as the following.

```
Brocade(config)#router pim
Brocade(config-pim-router)#ssm-enable
```

Syntax: [no] ssm-enable

Enter the ssm-enable command under the router pim level to globally enable the SSM protocol on a Layer 3 switch.

Displaying IGMP V3 information on Layer 3 switches

The sections below present the show commands available for IGMP V3 on Layer 3 switches. For show commands on Layer 2 Switches, use the **show ip multicast** commands which are discussed in the section [“IGMP snooping show commands”](#) on page 1477.

Displaying IGMP group status

NOTE

This report is available on Layer 3 switches.

To display the status of all IGMP multicast groups on a device, enter the **ip igmp group** command.

```
Brocade#show ip igmp group
p-:physical, ST:static, QR:querier, EX:exclude, IN: include, Y:yes, N:no
v101 : 1 groups, 1 group-port
      group      p-port      ST      QR      life mode      source
1       239.200.1.1  1/1/11    no      no      260    EX      0
```

To display the status of one IGMP multicast group, enter a command such as the following.

```
Brocade#show ip igmp group 239.0.0.1 detail
Display group 239.0.0.1 in all interfaces.
Interface v18 : 1 groups
      group      phy-port static querier life mode      #_src
1       239.0.0.1  e4/20    no      yes      include 19
group: 239.0.0.1, include, permit 19 (source, life):
(3.3.3.1 40) (3.3.3.2 40) (3.3.3.3 40) (3.3.3.4 40) (3.3.3.5 40)
(3.3.3.6 40) (3.3.3.7 40) (3.3.3.8 40) (3.3.3.9 40) (3.3.3.10 40)
(3.3.3.11 40) (3.3.3.12 40) (3.3.3.13 40) (3.3.3.14 40) (3.3.3.15 40)
(3.3.3.16 40) (3.3.3.17 40) (3.3.3.18 40) (3.3.3.19 40)
Interface v110 : 1 groups
      group      phy-port static querier life mode      #_src
2       239.0.0.1  e4/5     no      yes      include 10
group: 239.0.0.1, include, permit 10 (source, life):
(2.2.3.0 80) (2.2.3.1 80) (2.2.3.2 80) (2.2.3.3 80) (2.2.3.4 80)
(2.2.3.5 80) (2.2.3.6 80) (2.2.3.7 80) (2.2.3.8 80) (2.2.3.9 80)
```

If the tracking and fast leave feature is enabled, you can display the list of clients that belong to a particular group by entering commands such as the following.

```
Brocade#show ip igmp group 224.1.10.1 tracking
Display group 224.1.10.1 in all interfaces with tracking enabled.
Interface v13 : 1 groups, tracking_enabled
      group      phy-port static querier life mode      #_src
1       224.1.10.1  e4/15    no      yes      include 3
receive reports from 3 clients:
110.110.110.7 110.110.110.8 110.110.110.9
```

Syntax: **show ip igmp group** [<group-address>] [**detail** | **tracking**]

If you want a report for a specific multicast group, enter that group address for <group-address>. Omit the <group-address> if you want a report for all multicast groups.

Enter **detail** if you want to display the source list of the multicast group.

Enter **tracking** if you want information on interfaces that have tracking enabled.

The following table defines the statistics for the **show ip igmp group** command output.

TABLE 270 Output of show ip igmp group

Field	Description
Group	The address of the multicast group
Phy-port	The physical port on which the multicast group was received.
Static	A “yes” entry in this column indicates that the multicast group was configured as a static group; “No” means it was not. Static multicast groups can be configured in IGMP V2 using the ip igmp static command. In IGMP V3, static sources cannot be configured in static groups.
Querier	“Yes” means that the port is a querier port; “No” means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the port.
Life	Shows the number of seconds the interface can remain in exclude mode. An exclude mode changes to include mode if it does not receive an "IS_EX" or "TO_EX" message during a certain period of time. The default is 140 seconds. There is no "life" displayed in include mode.
Mode	Indicates current mode of the interface: Include or Exclude. If the interface is in Include mode, it admits traffic only from the source list. If an interface is in Exclude mode, it denies traffic from the source list and accepts the rest.
#_src	Identifies the source list that will be included or excluded on the interface. If IGMP V2 group is in Exclude mode with a #_src of 0, the group excludes traffic from 0 (zero) source list, which means that all traffic sources are included.
Group:	<p>If you requested a <i>detailed</i> report, the following information is displayed:</p> <ul style="list-style-type: none"> • The multicast group address • The mode of the group • A list of sources from which traffic will be admitted (include) or denied (exclude) on the interface is listed. • The life of each source list. <p>If you requested a <i>tracking</i> report, the clients from which reports were received are identified.</p>

Displaying the IGMP status of an interface

You can display the status of a multicast enabled port by entering the **ip igmp interface** command.

NOTE

This report is available on Layer 3 switches.


```

Brocade#show ip igmp interface
query interval = 60, max response time= 3, group membership time=140
v5: default V2,          PIM dense, addr=1.1.1.2
   e4/12  has    0 groups, non-Querier (age=40), default V2
v18: default V2,         DVMRP, addr=2.2.2.1
   e4/20  has    0 groups, Querier, default V2
v20: configured V3,      PIM dense (port down), addr=1.1.20.1
v110: configured V3,     PIM dense, addr=110.110.110.1
   e4/6   has    2 groups, Querier, default V3
       group: 239.0.0.1, exclude, life=100, deny 13
       group: 224.1.10.1, include, permit 2
   e4/5   has    3 groups, Querier, default V3
       group: 224.2.2.2, include, permit 100
       group: 239.0.0.1, include, permit 10
       group: 224.1.10.1, include, permit 1

```

Syntax: `show ip igmp interface [ve | ethernet <number> <group-address>]`

Enter **ve** and its <number> or **ethernet** and its <number> to display information for a specific virtual routing interface or ethernet interface.

Entering an address for <group-address> displays information for a specified group on the specified interface.

The report shows the following information.

TABLE 271 Output of show ip igmp interface

Field	Description
Query interval	Displays how often a querier sends a general query on the interface.
Max response	The maximum number of seconds a client can wait before it replies to the query.
Group membership time	The number of seconds multicast groups can be members of this group before aging out.
(details)	<p>The following is displayed for each interface:</p> <ul style="list-style-type: none"> • The ID of the interface • The IGMP version that it is running (default IGMP V2 or configured IGMP V3) • The multicast protocol it is running: DVMRP, PIM-DM, PIM-SM • Address of the multicast group on the interface • If the interface is a virtual routing interface, the physical port to which that interface belongs, the number of groups on that physical port, whether or not the port is a querier or a non-querier port, the age of the port, and other multicast information for the port are displayed.

Displaying IGMP traffic status

To display the traffic status on each virtual routing interface, enter the **show ip igmp traffic** command.

NOTE

This report is available on Layer 3 switches.

```
Brocade#show ip igmp traffic
Recv  QryV2  QryV3  G-Qry  GSQry  MbrV2  MbrV3  Leave  IsIN  IsEX  ToIN  ToEX  ALLOW  BLK
v5      29      0      0      0      0      0      0      0      0      0      0      0      0
v18     15      0      0      0      0     30      0     60      0      0      0      0      0
v110      0      0      0      0      0     97      0    142     37      2      2      3      2
Send  QryV1  QryV2  QryV3  G-Qry  GSQry
v5      0      2      0      0      0
v18      0      0     30     30      0
v110      0      0     30     44     11
```

Syntax: show ip igmp traffic

The report shows the following information.

TABLE 272 Output of show ip igmp traffic

Field	Description
QryV2	Number of general IGMP V2 query received or sent by the virtual routing interface.
QryV3	Number of general IGMP V3 query received or sent by the virtual routing interface.
G-Qry	Number of group specific query received or sent by the virtual routing interface.
GSQry	Number of source specific query received or sent by the virtual routing interface.
MbrV2	The IGMP V2 membership report.
MbrV3	The IGMP V3 membership report.
Leave	Number of IGMP V2 "leave" messages on the interface. (See ToEx for IGMP V3.)
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from exclude to include.
ToEX	Number of times the interface mode changed from include to exclude.
ALLOW	Number of times that additional source addresses were allowed or denied on the interface.
BLK	Number of times that sources were removed from an interface.

Clearing IGMP statistics

To clear statistics for IGMP traffic, enter the **clear igmp traffic** command.

```
Brocade#clear igmp traffic
```

Syntax: clear igmp traffic

This command clears all the multicast traffic information on all interfaces on the device.

IGMP Proxy

IGMP Proxy provides a means for the FastIron X Series routers to receive any or all multicast traffic from an upstream device if the router is not able to run PIM.

IGMP Proxy enables the router to issue IGMP host messages on behalf of hosts that the router discovered through standard PIM interfaces. The router acts as a proxy for its hosts and performs the host portion of the IGMP task on the upstream interface as follows:

- When queried, the router sends group membership reports for the groups learned
- When one of its hosts joins a multicast address group to which none of its other hosts belong, the router sends unsolicited membership reports to that group.
- When the last of its hosts in a particular multicast group leaves the group, the FastIron X Series router sends an unsolicited leave group membership report to group for all routers (multicast IP address 224.0.0.2)

IGMP proxy configuration notes

When using IGMP Proxy, you must do the following.

1. Configure PIM on all multicast client ports to build the group membership table. The group membership table will be reported by the proxy interface. Refer to [“Globally enabling and disabling PIM”](#) on page 1525.
2. Enable IP multicast on an interface to an upstream FastIron X Series router that will be the IGMP proxy interface and configure IGMP Proxy on that interface

IGMP proxy limitations

- IGMP Proxy cannot be enabled on the same interface on which PIM SM, PIM DM, or DVMRP is enabled.
- IGMP Proxy is only supported in a PIM Dense environment where there are IGMP clients connected to the Brocade device. The Brocade device will not send IGMP reports on an IGMP proxy interface for remote clients connected to a PIM neighbor, as it will not be aware of groups that the remote clients are interested in.

Configuring IGMP Proxy

Perform the following steps to configure IGMP Proxy.

1. Configure router PIM globally.

```
Brocade(config)#router pim
```
2. Configure an IP address on the interface (physical or virtual routing interface) that will serve as the IGMP proxy for an upstream device by entering commands such as the following.

```
Brocade(config)#int e 1/3
Brocade(config-if-e1000-1/3)#ip address 207.95.5.1/24
```
3. Enable IGMP Proxy on the interface.

```
Brocade(config-if-e1000-1/3)#ip igmp proxy
```

Syntax: [no] ip igmp proxy

Once IGMP Proxy is configured and the FastIron X Series router receives a query on an IGMP Proxy interface, the router sends a report in response to the query before the IGMP maximum response time expires.

Displaying IGMP Proxy traffic

Use the **show ip igmp traffic** command to see traffic for IGMP Proxy.

```
Brocade#show ip igmp traffic
```

Recv	QryV2	QryV3	G-Qry	GSQry	MbrV2	MbrV3	Leave	IsIN	IsEX	ToIN	ToEX	ALLO	BLK
e1/14	0	0	0	0	27251	0	12	0	27251	12	0	0	0
v10	250	0	0	0	244	0	0	0	244	0	0	0	0
Send	QryV1	QryV2	QryV3	G-Qry	GSQry	MbrV1	Mbrv2	Leave					
e1/14	0	1365	0	48	0	0	0	0					
v10	0	1	0	0	0	0	25602	1					

Syntax: show ip igmp traffic

Refer to [“Displaying IGMP traffic status”](#) on page 1604 to interpret the information in the output. The fields in bold show information for IGMP Proxy.

IP multicast protocols and IGMP snooping on the same device

The Brocade device supports global Layer 2 IP multicast traffic reduction (IGMP snooping) and Layer 3 multicast routing (DVMRP or PIM-Sparse or PIM-Dense) together on the same device in the full Layer 3 software image, as long as the Layer 2 feature configuration is at the VLAN level.

For Layer 2 multicast traffic reduction, IGMP snooping is performed independently within all VLANs that have the feature configured. Layer 3 multicast routing is performed between the IP interfaces that are configured for DVMRP, PIM-Sparse, or PIM-Dense. A Layer 3 interface could be a physical, loopback, or VE port configured with an IP address.

If there are two sources for a single group, where one source sends traffic into a VLAN with IGMP snooping enabled, while the other source sends traffic to a PIM enabled Layer 3 interface, a client for the group in the same VLAN as the first source will only receive traffic from that source. It will not receive traffic from the second source connected to the Layer 3 interface. Similarly, if there is another IP interface with a Layer 3 client or PIM or DVMRP neighbor that requests traffic for the same group, it will only receive traffic from the second source and not the first.

IP multicast protocols and IGMP snooping configuration example

Figure 182 and Figure 183 show an example IGMP snooping and PIM forwarding configuration.

FIGURE 182 Example 1: IGMP Snooping and PIM forwarding

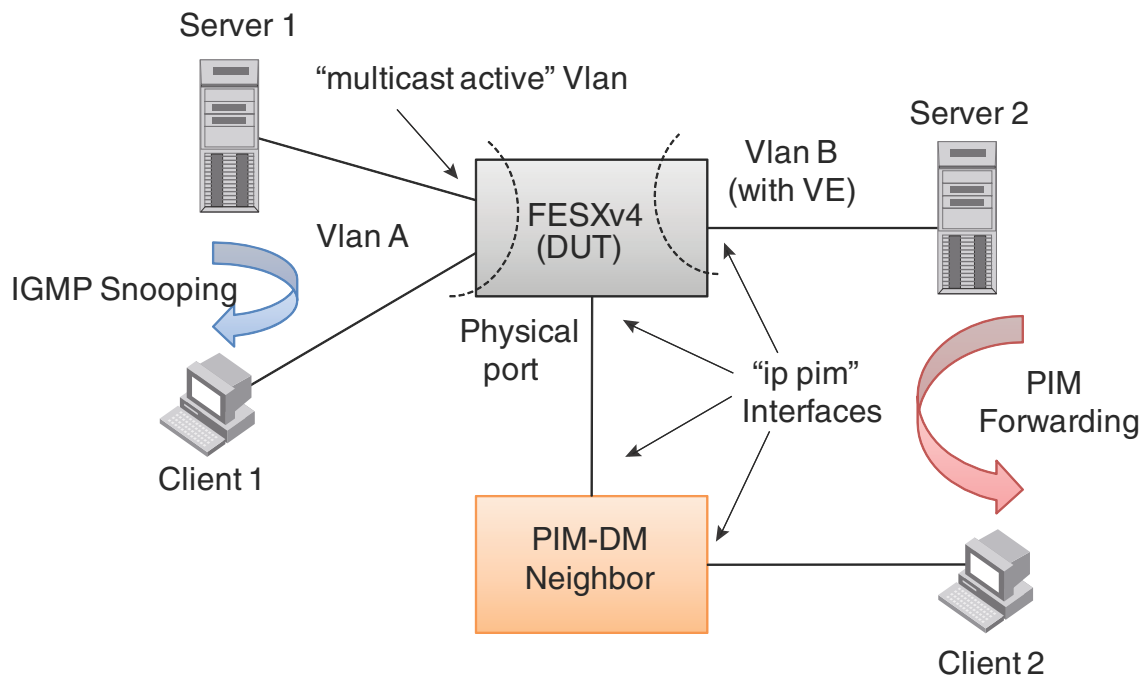
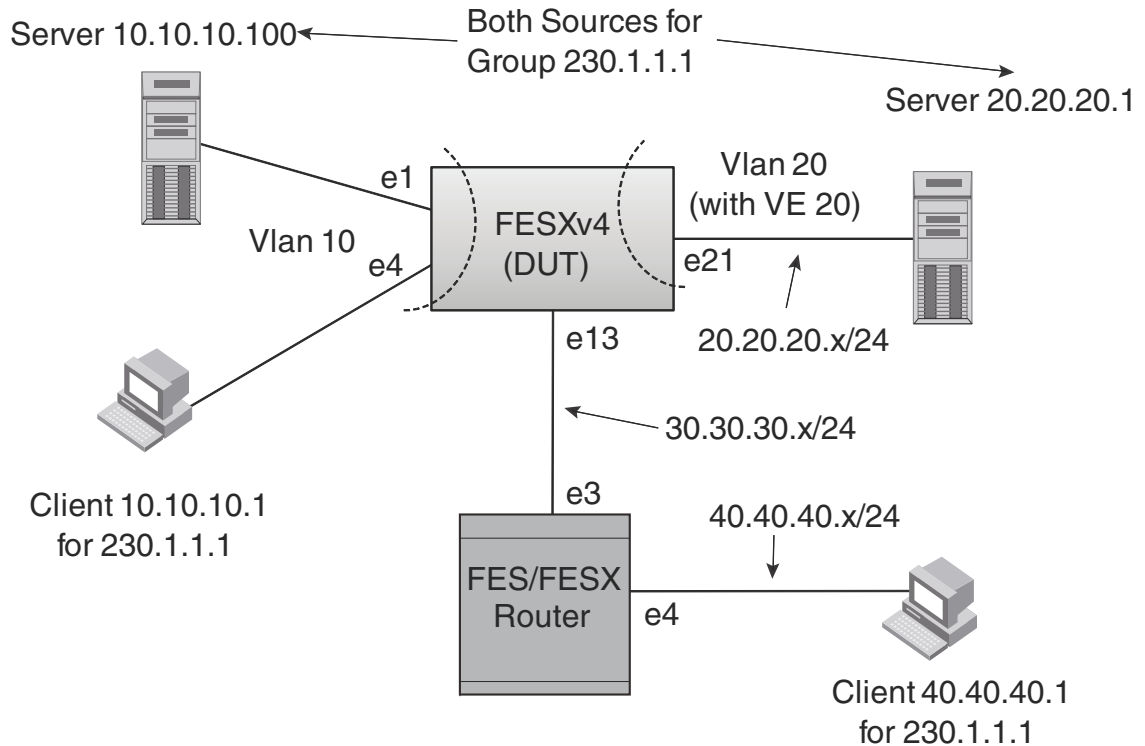


FIGURE 183 Example 2: IGMP Snooping and PIM Forwarding



IP multicast protocols and IGMP snooping CLI commands

The following are the CLI commands for the configuration example shown in [Figure 182](#) and [Figure 183](#).

1. On the FESXv4 device, configure IGMP Snooping on VLAN 10.

```
Brocade(config)#vlan 10 by port
Brocade(config-vlan-10)#untagged e 1 to 4
Added untagged port(s) ethe 1 to 4 to port-vlan 10.
Brocade(config-vlan-10)#router-interface ve 10
Brocade(config-vlan-10)#ip multicast active
Brocade(config-vlan-10)#interface ve 10
Brocade(config-vif-10)#ip address 10.10.10.10/24
```

2. On the FESXv4 device, enable PIM routing between VLAN/VE 20 and Interface e 13.

```
Brocade(config)#vlan 20 by port
Brocade(config-vlan-20)#untagged e 21 to 24
Added untagged port(s) ethe 21 to 24 to port-vlan 20.
Brocade(config-vlan-20)#router-interface ve 20
Brocade(config-vlan-20)#exit
Brocade(config)#router pim
Brocade(config-pim-router)#exit
Brocade(config)#interface ve 20
Brocade(config-vif-20)#ip address 20.20.20.10/24
Brocade(config-vif-20)#ip pim
```

```
Brocade(config-vif-20)#exit
Brocade(config)#interface e 13
Brocade(config-if-e1000-13)#ip address 30.30.30.10/24
Brocade(config-if-e1000-13)#ip pim
```

3. Configure the FES/FESX neighboring device.

```
Brocade(config)#ip route 20.20.20.0 255.255.255.0 30.30.30.10
Brocade(config)#router pim
Brocade(config-pim-router)#exit
Brocade(config)#interface ethernet 3
Brocade(config-if-e1000-3)#ip address 30.30.30.20/24
Brocade(config-if-e1000-3)#ip pim
Brocade(config-if-e1000-3)#interface ethernet 4
Brocade(config-if-e1000-4)#ip address 40.40.40.20/24
Brocade(config-if-e1000-4)#ip pim
```

IP multicast protocols and IGMP snooping on the same device

MLD Snooping on FastIron X Series Switches

[Table 273](#) lists the individual Brocade FastIron switches and the Multicast Listening Discovery (MLD) snooping features they support. These features are supported in the Layer 2, base Layer 3, and full Layer 3 software images.

TABLE 273 Supported MLD snooping features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
MLD V1/V2 snooping (global and local)	Yes	This chapter describes MLD snooping on the FESX, FSX 800 and FSX 1600. For information about MLD snooping on other FastIron devices, see Chapter 38 , “MLD Snooping on FastIron WS and Brocade FCX and ICX Switches”.			
MLD fast leave for V1	Yes				
MLD tracking and fast leave for V2	Yes				
Static MLD and IGMP groups with support for proxy	Yes				

MLD Snooping Overview

The default method a device uses to process an IPv6 multicast packet is to broadcast it to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to the CPU, which may result in some clients receiving unwanted traffic.

If a VLAN is not Multicast Listening Discovery (MLD) snooping-enabled, it floods IPv6 multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, MLD packets are trapped to the CPU. Data packets are mirrored to the CPU and flooded to the entire VLAN. The CPU then installs hardware resources so subsequent data packets can be hardware-switched to desired ports without going through the CPU. If there is no client report, the hardware resource drops the data stream.

MLD protocols provide a way for clients and a device to exchange messages, and allow the device to build a database indicating which port wants what traffic. Since the MLD protocols do not specify forwarding methods, MLD snooping or multicast protocols such as IPv6 PIM-Sparse Mode (PIM SM) are required to handle packet forwarding. PIM SM can route multicast packets within and outside a VLAN, while MLD snooping can switch packets only within a VLAN. FESX and FSX devices do not support PIM-SM routing.

MLD snooping provides multicast containment by forwarding traffic only to those clients that have MLD receivers for a specific multicast group (destination address). The device maintains the MLD group membership information by processing MLD reports and generating messages so traffic can be forwarded to ports receiving MLD reports. This is analogous to IGMP Snooping on Brocade Layer 3 switches.

An IPv6 multicast address is a destination address in the range of FF00::/8. A limited number of multicast addresses are reserved. Since packets destined for the reserved addresses may require VLAN flooding, FESX and FSX devices do not snoop in the FFOX::00X range (where X is from 00 to FF). Data packets destined to these addresses are flooded to the entire VLAN by hardware and mirrored to the CPU. Multicast data packets destined to addresses outside the FFOX::00X range are snooped. A client must send MLD reports in order to receive traffic.

An MLD device periodically broadcasts general queries and sends group queries upon receiving a leave message, to ensure no other clients at the same port still want this specific traffic before removing it. MLDv1 allows clients to specify which group (destination IPv6 address) will receive traffic. (MLDv1 cannot choose the source of the traffic.) MLDv2 deals with source-specific multicasts, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An MLDv2 device's port state can either be in INCLUDE or EXCLUDE mode.

There are different types of group records for client reports. Clients respond to general queries by sending a membership report containing one or more of the following records associated with a specific group:

- **Current-state record** - Indicates the sources from which the client wants to receive or not receive traffic. This record contains the source addresses of the clients and indicates whether or not traffic will be included (IS_IN) or excluded (IS_EX) from that source address.
- **Filter-mode-change record** - If the client changes its current state from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if a client current state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- **MLDv1 leave report** - Equivalent to a TO_IN (empty) record in MLDv2. This record means that no traffic from this group will be received, regardless of the source.
- **An MLDv1 group report** - Equivalent to an IS_EX (empty) record in MLDv2. This record means that all traffic from this group will be received, regardless of the source.
- **Source-list-change record** - If the client wants to add or remove traffic sources from its membership report, the report can include an ALLOW record, which contains a list of new sources from which the client wishes to receive traffic. The report can also contain a BLOCK record, which lists current traffic sources from which the client wants to stop receiving traffic.

How MLD snooping uses MAC addresses to forward multicast packets

Multicast Listening Discovery (MLD) snooping on Brocade devices is based on MAC address entries. When an IPv6 multicast data packet is received, the packet destination MAC is matched with the MAC address entries in the IPv6 multicast table. If a match is found, packets are sent to the ports associated with the MAC address. If a match is not found, packets are flooded to the VLAN and copied to the CPU.

For IPv6 multicast, the destination MAC address is in the format 0x33-33-xx-yy-zz-kk, where xx-yy-zz-kk are the 32 lowest bits of the IPv6 multicast group address. For example, the IPv6 group address 0xFF3E:40:2001:660:3007:123:1234:5678 maps to the IPv6 MAC address 0x33-33-12-34-56-78.

For two multicast traffic streams, Source_1 and Group1 (S1,G1) and Source_2 and Group2 (S2,G2), with the same or different source addresses, if the lowest 32 bits of the 128-bit IPv6 group address are the same, they would map to the same destination MAC. Because the FESX and FSX support MAC-based forwarding for MLD snooping, the final multicast MAC address entry would be a superset of all the IPv6 groups mapped to it. For example, consider the following three IPv6 multicast streams sent from port 5 of a Brocade device:

- (S1,G1) = (2060::5, ff1e::12), client port 1, port 2
- (S2,G2) = (2060::6, ff1e::12), client port 2, port 3
- (S3,G1) = (2060::7, ff1e::12), client port 4

Because the lowest 32 bits of the group address for G1 and G2 are the same, all three streams would use 0x33-33-00-00-00-12 as the destination MAC address. MLD snooping would build a MAC entry with the MAC address 0x33-33-00-00-00-12 on egress ports 1, 2, 3, and 4. As a result, all three streams would be sent to ports 1, 2, 3, and 4. Note that the above example assumes the following:

- The Brocade device is running MLD snooping on VLAN 10 and all three streams are in VLAN 10
- There are clients on port 1 and port 2 for (S1,G1)
- There are clients on port 2 and port 3 for (S2,G2)
- There are clients on port 4 for (S3,G1)

MLD snooping configuration notes and feature limitations

- Servers (traffic sources) are not required to send Multicast Listening Discovery (MLD) memberships.
- The default MLD version is V1, where the source address is not sensitive. In the example given in the preceding section ([“How MLD snooping uses MAC addresses to forward multicast packets”](#) on page 1612), (S1,G1) and (S3,G1) would be considered the same group as (*,G1).
- If MLDv2 is configured on any port of a VLAN, you can check the source information, but because MLD snooping is MAC based, (S,G) switching is not feasible.
- Hardware resources are installed only when there is data traffic.
- Up to 4K collective entries for IGMP Snooping, MLD snooping, and Multi-port Static MAC Addresses are supported.
- You can configure the maximum number of groups and the multicast cache (mcache) number.
- The device supports static groups applying to specific ports. The device acts as a proxy to send MLD reports for the static groups when receiving queries.
- A user can configure static router ports, forcing all multicast traffic to be sent to these ports.
- Brocade devices support fast leave for MLDv1, which stops traffic immediately to any port that has received a leave message.
- Brocade devices support tracking and fast leave for MLDv2, which tracks all MLDv2 clients. If the only client on a port leaves, traffic is stopped immediately.
- An MLD device can be configured as a querier (active) or non-querier (passive). Queriers send queries. Non-queriers listen for queries and forward them to the entire VLAN.
- Every VLAN can be independently configured as a querier or a non-querier.

- A VLAN that has a connection to an IPv6 PIM-enabled port on another router should be configured as a non-querier. When multiple snooping devices connect together and there is no connection to IPv6 PIM ports, only one device should be configured as the querier. If multiple devices are configured as active, only one will continue to send queries after the devices have exchanged queries. Refer to [“MLD snooping-enabled queriers and non-queriers”](#) on page 1614.
- An MLD device can be configured to rate-limit the forwarding of MLDv1 membership reports to queriers.
- Because an IPv6 link-local address as the source address when sending queries, a global address is not required.
- The MLD implementation allows snooping on some VLANs or on all VLANs. MLD can be enabled or disabled independently for each VLAN. In addition, individual ports of a VLAN can be configured as MLDv1 and MLDv2. In general, global configuration commands such as **ipv6 mld-snooping..** apply to all VLANs except those with a local **mld-snooping..** configuration, which supersedes the global configuration. Configuring the version on a port or a VLAN only affects the device sent query version. The device always processes all versions of client reports regardless of the version configured.
- MLD snooping requires hardware resources. If the device has insufficient resources, the data stream without a resource is mirrored to the CPU in addition to being VLAN flooded, which can cause high CPU usage. To avoid this situation, Brocade recommends that you avoid enabling snooping globally unless necessary.
- To receive data traffic, MLD snooping requires clients to send membership reports. If a client does not send reports, you must configure a static group to force traffic to client ports.
- Multicast Router Discovery (MRD) messages are useful for determining which nodes attached to a switch have multicast routing enabled. This capability is useful in a Layer 2 bridge domain with snooping switches. By utilizing MRD messages, Layer 2 switches can determine where to send multicast source data and group membership messages. Multicast source data and group membership reports must be received by all multicast routers on a segment. Using the group membership protocol Query messages to discover multicast routers is insufficient due to query suppression.

Since Brocade does not support MRD, this can lead to stream loss when non-Querier router ports age out on the Querier after the initial Query election. To avoid such stream loss, configure a static router port on the querier on each interface that connects to a non-querier snooping device.

MLD snooping-enabled queriers and non-queriers

An MLD snooping-enabled device can be configured as a querier (active) or non-querier (passive). An MLD querier sends queries; a non-querier listens for MLD queries and forwards them to the entire VLAN. When multiple MLD snooping devices are connected together, and there is no connection to an IPv6 PIM-enabled port, one of the devices should be configured as a querier. If multiple devices are configured as queriers, after multiple devices exchange queries, then all devices except the winner (the device with the lowest address) stop sending queries. Although the system works when multiple devices are configured as queriers, Brocade recommends that only one device, preferably the one with the traffic source, is configured as the querier.

VLANs can also be independently configured as queriers or non-queriers. If a VLAN has a connection to an IPv6 PIM-enabled port on another router, the VLAN should be configured as a non-querier.

Because non-queriers always forward multicast data traffic and MLD messages to router ports which receive MLD queries or IPv6 PIM hellos, Brocade recommends that you configure the devices with the data traffic source (server) as queriers. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether or not there are clients on the querier.

NOTE

In a topology with one or more connected devices, at least one device must be running PIM, or configured as active. Otherwise, no devices can send queries, and traffic cannot be forwarded to clients.

To configure the MLD mode (querier or non-querier) on an MLD snooping-enabled device, refer to [“Configuring the global MLD mode”](#) on page 1617. To configure the MLD mode on a VLAN, refer to [“Configuring the MLD mode for a VLAN”](#) on page 1619.

MLD and VLAN configuration

You can configure MLD snooping on some VLANs or all VLANs. Each VLAN can be independently enabled or disabled for MLD snooping, or can be configured with MLDv1 or MLDv2. In general, the **ipv6 mld-snooping...** commands apply globally to all VLANs except those configured with VLAN-specific **mld-snooping...** commands. VLAN-specific **mld-snooping** commands supersede global **ipv6 mld-snooping** commands.

MLDv1 with MLDv2

MLD snooping can be configured as MLDv1 or MLDv2 on individual ports on a VLAN. An interface or router sends queries and reports that include the MLD version with which it has been configured. The version configuration applies only to the sending of queries. The snooping device recognizes and processes MLDv1 and MLDv2 packets regardless of the version configured.

NOTE

To avoid version deadlock, when an interface receives a report with a lower version than that for which it has been configured, the interface does **not** automatically downgrade the running MLD version.

MLD snooping configuration

Configuring Multicast Listening Discovery (MLD) snooping on an IPv6 device consists of the following global and VLAN-specific tasks.

MLD snooping global tasks

- Configuring hardware and software resource limits
- Disabling transmission and receipt of MLD packets on a port
- Configuring the MLD mode: active or passive (must be enabled for MLD snooping)
- Modifying the age interval
- Modifying the interval for query messages (active MLD mode only)
- Specifying the global MLD version
- Enabling and disabling report control (rate limiting)

- Modifying the leave wait time
- Modifying the mcache age interval
- Disabling error and warning messages

MLD snooping VLAN-specific tasks:

- Configuring the MLD mode for the VLAN: active or passive
- Enabling or disabling MLD snooping for the VLAN
- Configuring the MLD version for the VLAN
- Configuring the MLD version for individual ports
- Configuring static groups to the entire VLAN or some ports
- Configuring static router ports
- Disabling proxy activity for a static group
- Enabling client tracking and the fast leave feature for MLDv2
- Configuring fast leave for MLDv1
- Configuring fast-convergence

Configuring the hardware and software resource limits

The system supports up to 8K of hardware-switched multicast streams. The default is 512. To define the maximum number of MLD snooping mcache entries, enter the **system-max mld-snoop-mcache** *<num>* command.

```
Brocade(config)#system-max mld-snoop-mcache 8000
```

Syntax: [no] **system-max mld-snoop-mcache** *<num>*

<num> is a value from 256 – 8192. The default is 512.

The system supports up to 32K of multicast groups. The default is 8192. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limits are not processed. To define the maximum number of multicast group addresses supported, enter the **system-max mld-max-group-addr** *<num>* command.

```
Brocade(config)#system-max mld-max-group-addr 4000
```

Syntax: [no] **system-max mld-snoop-mcache** *<num>*

<num> is a value from 256 – 32768. The default is 8192.

Disabling transmission and receipt of MLD packets on a port

When a VLAN is snooping-enabled, all MLD packets are trapped to the CPU without hardware VLAN flooding. The CPU can block MLD packets to and from a multicast-disabled port, and will not add that port to the output interfaces of hardware resources, which prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic for these groups is flooded to the entire VLAN, including to the disabled ports. Since the hardware cannot block traffic from disabled ports, hardware traffic is switched in the same way as traffic from enabled ports.

NOTE

This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

```
Brocade(config)#interface ethernet 1/3
Brocade(config-if-e1000-1/3)#ipv6-multicast-disable
```

Syntax: [no] **ipv6-multicast-disable**

Configuring the global MLD mode

You can configure a Brocade device for either active or passive (default) MLD mode. If you specify an MLD mode for a VLAN, the MLD mode overrides the global setting.

- **Active** – In active MLD mode, a device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- **Passive** – In passive MLD mode, the device forwards reports to the router ports which receive queries. MLD snooping in passive mode does not send queries, but does forward queries to the entire VLAN.

To globally set the MLD mode to active, enter the **ipv6 mld-snooping active** command.

```
Brocade(config)#ipv6 mld-snooping active
```

Syntax: [no] **ipv6 mld-snooping** [active | passive]

Omitting both the **active** and **passive** keywords is the same as entering **ipv6 mld-snooping passive**.

Modifying the age interval

When the device receives a group membership report, it makes an entry in the MLD group table for the group in the report. The age interval specifies how long the entry can remain in the table without the device receiving another group membership report. When multiple devices connect together, all devices should be configured with the same age interval. The age interval should be at least twice that of the query interval, so that missing one report will not stop traffic. For a non-querier, the query interval should equal that of the querier.

To modify the age interval, enter a command such as the following.

```
Brocade(config)#ipv6 mld-snooping age-interval 280
```

Syntax: [no] **ipv6 mld-snooping age-interval** <interval>

The <interval> parameter specifies the aging time. You can specify a value from 20 – 7200 seconds. The default is 140 seconds.

Modifying the query interval (active MLD snooping mode only)

If the MLD mode is set to active, you can modify the query interval, which specifies how often the Brocade device sends group membership queries. By default, queries are sent every 60 seconds. When multiple queriers connect together, all queriers should be configured with the same interval.

To modify the query interval, enter the **ipv6 mld-snooping query-interval** <num> command.

```
Brocade(config)#ipv6 mld-snooping query-interval 120
```

Syntax: [no] ipv6 mld-snooping query-interval <interval>

The <interval> parameter specifies the interval between queries. You can specify a value from 10 – 3600 seconds. The default is 60 seconds.

Configuring the global MLD version

The default version is MLDv1. You can specify the global MLD version on the device as either MLDv1 or MLDv2. For example, the following command configures the device to use MLDv2.

```
Brocade(config)#ipv6 mld-snooping version 2
```

Syntax: [no] ipv6 mld-snooping version 1 | 2

You can also specify the MLD version for individual VLANs, or individual ports within VLANs. If no MLD version is specified for a VLAN, then the globally configured MLD version is used. If an MLD version is specified for individual ports in a VLAN, those ports use that version instead of the version specified for the VLAN or the globally specified version. The default is MLDv1.

Configuring report control

When a device is in passive mode, it forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding for the same group to no more than once per 10 seconds. This rate limiting does not apply to the first report answering a group-specific query.

NOTE

This feature applies to MLDv1 only. The leave messages are not rate limited.

MLDv1 membership reports for the same group from different clients are considered to be the same, and are rate-limited. This alleviates the report storm caused by multiple clients answering the upstream router query.

To enable report-control, enter the **ipv6 mld-snooping report-control** command.

```
Brocade(config)#ipv6 mld-snooping report-control
```

Syntax: [no] ipv6 mld-snooping report-control

Modifying the wait time before stopping traffic when receiving a leave message

You can define the wait time before stopping traffic to a port when the device receives a leave message for that port. The device sends group-specific queries once per second to determine if any client on the same port still needs the group.

```
Brocade(config)#ipv6 mld-snooping leave-wait-time 1
```

Syntax: [no] ipv6 mld-snooping leave-wait-time <num>

where <num> is a value from 1 to 5. The default is 2. Because of the internal timer accuracy, the actual wait time is between n and (n+1) seconds, where n is the configured value.

Modifying the multicast cache (mcache) aging time

You can set the time for an mcache to age out when it does not receive traffic. Two seconds before an mcache is aged out, the device mirrors a packet of the mcache to the CPU to reset the age. If no data traffic arrives within two seconds, the mcache is deleted. Note that in FESX and FSX devices, more than one mcache can be mapped to the same destination MAC. Hence, when an mcache entry is deleted, the MAC entry may not be deleted. If you configure a lower value, the resource consumed by idle streams is quickly removed, but packets are mirrored to the CPU more frequently. Configure a higher value only when data streams are arriving consistently.

You can use the **show ipv6 mld-snooping mcache** command to view the currently configured mcache age. Refer to [“Displaying MLD snooping mcache information”](#) on page 1625.

To modify the multicast cache age out time, enter the **ipv6 mld-snooping mcache-age <num>** command.

```
Brocade(config)#ipv6 mld-snooping mcache-age 180
```

Syntax: [no] **ipv6 mld-snooping mcache-age <num>**

where <num> is a value from 60 to 3600 seconds, and the default is 60 seconds.

Disabling error and warning messages

Error or warning messages are printed when the device runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate limited. You can turn off these messages by entering the **ipv6 mld-snooping verbose-off** command.

```
Brocade(config)#ipv6 mld-snooping verbose-off
```

Syntax: [no] **ipv6 mld-snooping verbose-off**

Configuring the MLD mode for a VLAN

You can configure a VLAN for either the active or passive (default) MLD mode. The VLAN setting overrides the global setting:

- **Active** – In active MLD mode, the device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- **Passive** – In passive MLD mode, the device forwards reports to router ports that receive queries. MLD snooping in the passive mode does not send queries. However, it does forward queries to the entire VLAN.

To set the MLD mode for VLAN 20 to active, enter the following commands.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping active
```

Syntax: [no] **mld-snooping active | passive**

The default mode is passive.

Disabling MLD snooping for the VLAN

When MLD snooping is enabled globally, you can disable it for a specific VLAN. For example, the following commands disable MLD snooping for VLAN 20. This setting overrides the global setting for VLAN 20.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping disable-mld-snoop
```

Syntax: [no] mld-snooping disable-mld-snoop

Configuring the MLD version for the VLAN

You can specify the MLD version for a VLAN. For example, the following commands configure VLAN 20 to use MLDv2.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping version 2
```

Syntax: [no] mld-snooping version 1 | 2

When no MLD version is specified, the globally-configured MLD version is used. If an MLD version is specified for individual ports, these ports use that version, instead of the version specified for the VLAN.

Configuring the MLD version for individual ports

You can specify the MLD version for individual ports in a VLAN. For example, the following commands configure ports 1/4, 1/5, 1/6 and 2/1 to use MLDv2. The other ports use the MLD version specified with the **mld-snooping version** command, or the globally configured MLD version.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping port-version 2 ethe 2/1 ethe 1/4 to 1/6
```

Syntax: [no] mld-snooping port-version 1 | 2 ethernet <port-numbers>

Configuring static groups to the entire VLAN or to individual ports

NOTE

On FastIron X Series devices, you can configure MLD snooping static groups for individual ports in a VLAN, but not for the entire VLAN.

A snooping-enabled VLAN cannot forward multicast traffic to ports that do not receive MLD membership reports. To allow clients to send reports, you can configure a static group which applies to the entire VLAN, or to individual ports on the VLAN. The maximum number of supported static groups in a VLAN is 512, and the maximum number of supported static groups for individual ports in a VLAN is 256. The static group forwards packets to the static group ports even if they have no client membership reports. The static group for the entire VLAN is used in VLAN flooding because it uses fewer hardware resources than the static group for individual ports. Configure a static group for specific ports on VLAN 20 using commands similar to the following.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping static-group ff05::100 count 2 ethe 1/3 ethe
1/5 to 1/7
Brocade(config-vlan-20)#mld-snooping static-group ff10::200
```

Syntax: [no] mld-snooping static-group <ipv6-address> [count <num>] [<port-numbers>]

The **ipv6-address** parameter is the IPv6 address of the multicast group.

The **count** is optional, which allows a contiguous range of groups. Omitting the count <num> is equivalent to the count being 1.

Except on FastIron X Series devices, if there are no port-numbers, the static groups apply to the entire VLAN.

Configuring static router ports

All multicast control and data packets are forwarded to router ports that receive queries. Although router ports are learned, you can configure static router ports to force multicast traffic to specific ports, even though these ports never receive queries. To configure static router ports, enter commands such as the following.

```
Brocade(config)#vlan 70
Brocade(config-vlan-70)#mld-snooping router-port e 1/4 to 1/5 e 1/8
```

Syntax: [no] mld-snooping router-port ethernet <port-numbers>

Disabling static group proxy

A device with statically configured groups acts as a proxy and sends membership reports for its static groups when it receives general or group-specific queries. When a static group configuration is removed, the group is immediately deleted from the active group table. However, the device does not send leave messages to the querier. The querier should age out the group. The proxy activity can be disabled (the default is enabled).

Example

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping proxy-off
```

Syntax: [no] mld-snooping proxy-off

By default, MLD snooping proxy is enabled.

Enabling MLDv2 membership tracking and fast leave for the VLAN

MLDv2 provides membership tracking and fast leave services to clients. In MLDv1, only one client per interface must respond to a router queries; leaving some clients invisible to the router, which makes it impossible for the device to track the membership of all clients in a group. In addition, when a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before stopping the traffic. You can configure the wait time with the **ipv6 mld-snooping leave-wait-time** command. Refer to [“Modifying the wait time before stopping traffic when receiving a leave message”](#) on page 1618.

MLDv2 requires that every client respond to queries, allowing the device to track every client. When the tracking feature is enabled, the device immediately stops forwarding traffic to the interface if an MLDv2 client sends a leave message, and there is no other client. This feature requires the entire VLAN to be configured for MLDv2 and have no MLDv1 clients. If a client does not send a report during the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can track group membership only; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each is receiving traffic from different sources. Client A receives a traffic stream from (source_1, group1) and Client B receives a traffic stream from (source_2, group1). The device waits for the configured **leave-wait-time** before it stops the traffic because the two clients are in the same group. If the clients are in different groups, the waiting period is ignored and traffic is stopped immediately.

To enable tracking and fast leave for VLAN 20, enter the following commands.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping tracking
```

Syntax: [no] mld-snooping tracking

The membership tracking and fast leave features are supported for MLDv2 only. If a port or client is not configured for MLDv2, the **mld-snooping tracking** command is ignored.

Configuring fast leave for MLDv1

When a FESX or FSX device receives an MLDv1 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic to this port. Configuring fast-leave-v1 allows the device to stop forwarding traffic to a port immediately upon receiving a leave message. The device does not send group-specific queries. It is important that no snooping ports have multiple clients. When two devices connect, the querier device should not be configured for fast-leave-v1 because the port to the non-querier device could have multiple clients. The number of queries and the waiting period (in seconds) can be configured using the **ipv6 mld-snooping leave-wait-time** command. Refer to [“Modifying the wait time before stopping traffic when receiving a leave message”](#) on page 1618.

To configure fast leave for MLDv1, use commands such as the following.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping fast-leave-v1
```

Syntax: [no] mld-snooping fast-leave-v1

Enabling fast convergence

In addition to periodically sending general queries, an active (querier) FESX or FSX device sends out general queries when it detects a new port. However, since it does not recognize the other device port-up event, the multicast traffic might still use the query-interval time to resume after a topology change. Configuring fast-convergence allows the device to listen to topology change events in Layer 2 (L2) protocols, such as spanning tree, and send general queries to shorten the convergence time.

If the L2 protocol is unable to detect a topology change, the fast-convergence feature may not work. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this an optimization action, rather than a topology change. In this case, other devices will not receive topology change notifications and will be unable to send queries to speed up the convergence. The original spanning tree protocol does not recognize optimization actions, and fast-convergence works in all cases.

To enable fast-convergence, enter commands such as the following.

```
Brocade(config)#vlan 70
Brocade(config-vlan-70)#mld-snooping fast-convergence
```

Syntax: [no] mld-snooping fast-convergence

Displaying MLD snooping information

You can display the following MLD snooping information:

- MLD snooping error information
- Group and forwarding information for VLANs
- Information about MLD snooping mcache
- MLD memory pool usage
- Status of MLD traffic
- MLD information by VLAN

Displaying MLD snooping error information

To display information about possible MLD errors, enter the following command.

```
Brocade#show ipv6 mld-snooping error
snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: show ipv6 mld-snooping error

The following table describes the output from the **show ipv6 mld-snooping error** command.

Field	Description
SW processed pkt	The number of IPv6 multicast packets processed by MLD snooping.
up-time	The MLD snooping up time.

Displaying MLD group information

To display MLD group information, enter the **ipv6 mld-snooping group** command.

```
Brocade#show ipv6 mld-snooping group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 263 grp, 263 grp-port, tracking_enabled
      group                                p-port ST QR life mode source
1      ff0e::ef00:a0e3                    1/7   N  Y  120 EX   0
2      ff01::1:f123:f567                  1/9   N  Y      IN   1
```

NOTE

In this example, an MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed MLD group information, enter the following command.

```
Brocade#show ipv6 mld-snooping group ff0e::ef00:a096 detail
Display group ff0e::ef00:a096 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                p-port ST QR life mode source
1      ff0e::ef00:a096                    1/7    N  Y  100 EX    0
      group: ff0e::ef00:a096, EX, permit 0 (source, life):
      life=100, deny 0:
```

If tracking and fast leave are enabled, you can display the list of clients for a particular group by entering the following command.

```
Brocade#show ipv6 mld-snooping group ff0e::ef00:a096 tracking
Display group ff0e::ef00:a096 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                p-port ST QR life mode source
1      ff0e::ef00:a096                    1/7    N  Y   80 EX    0
      receive reports from 1 clients: (age)
      (fe80::1011:1213:1415 60)
```

Syntax: `show ipv6 mld-snooping group` [*<group-address>*] [**detail**] [**tracking**]

To receive a report for a specific multicast group, enter that group address for *<group-address>*.

Enter **detail** to display the source list of a specific VLAN.

Enter **tracking** for information on interfaces that are tracking-enabled.

The following table describes the information displayed by the **show ipv6 mld-snooping group** command.

Field.	Description
group	The address of the IPv6 group (destination IPv6 address).
p-port	The physical port on which the group membership was received.
ST	Yes indicates that the MLD group was configured as a static group; No means it was learned from reports.
QR	Yes means the port is a querier port; No means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the port.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE if it does not receive an IS_EX or TO_EX message during a specified period of time. The default is 140 seconds. There is no life displayed in INCLUDE mode.
mode	The current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If the interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.

Field.	Description
source	Identifies the source list that will be included or excluded on the interface. An MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from 0 (zero) source list, which actually means that all traffic sources are included.
group	If you requested a <i>detailed</i> report, the following information is displayed: <ul style="list-style-type: none"> • The multicast group address • The mode of the group • Sources from which traffic will be admitted (INCLUDE) or denied (EXCLUDE) on the interface. • The life of each source list. If you requested a <i>tracking/fast leave</i> report, the clients from which reports were received are identified.

Displaying MLD snooping mcache information

The MLD snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the **ipv6 mld-snooping mcache** command.

```
Brocade#show ipv6 mld-snooping mcache
Example: (S G) cnt=: (S G) are the lowest 32 bits, cnt: SW proc. count
        OIF: 1/22 TR(1/32,1/33), TR is trunk, 1/32 primary, 1/33 output
vlan 1, has 2 cache
1      (abcd:ef50 0:100), cnt=121
        OIF: 1/11 1/9
        age=0s up-time=120s vidx=4130 (ref-cnt=1)
2      (abcd:ef50 0:101), cnt=0
        OIF: entire vlan
        age=0s up-time=0s vidx=8191 (ref-cnt=1)
vlan 70, has 0 cache
```

Syntax: show ipv6 mld-snooping mcache

The following table describes the output from the **ipv6 mld-snooping mcache** command.

Field	Description
(abcd:ef50 0:100):	The lowest 32 bits of source and group. It is displayed in XXXX:XXXX hex format. Here XXXX is a 16-bit hex number.
cnt	The number of packets processed in software.
OIF	Output interfaces. <i>Entire vlan</i> means that static groups apply to the entire VLAN.
age	The mcache age in seconds. The mcache is reset to 0 if traffic continues to arrive, otherwise it is aged out when it reaches the time defined by ipv6 mld-snooping mcache-age .
uptime	The up time of this mcache in seconds.
vidx	The vidx is shared among mcache using the same output interfaces. The vidx specifies the output port list, which shows the index. Valid range is from 4096 to 8191.
ref-cnt	The number of mcache using this vidx.

Displaying software resource usage for VLANs

To display information about the software resources used, enter the following command.

```
Brocade#show ipv6 mld-snooping resource
          alloc in-use  avail get-fail    limit  get-mem  size init
mld group      512     9   503         0   32000    272   28  256
mld phy port   1024    16  1008         0  200000    279   21 1024
snoop group hash  512     9   503         0   59392    272   20  256
... Entries deleted
total pool memory 194432 bytes
has total 1 forwarding hash
Available vidx: 4061
```

Syntax: show ipv6 mld-snooping resource

The following table describes the output from the show **ipv6 mld-snooping resource** command.

This field...	Displays...
alloc	The allocated number of units.
in-use	The number of units which are currently used.
avail	The number of available units.
get-fail	The number of resource failures. NOTE: It is important to pay close attention to this field.
limit	The upper limit of this expandable field. The MLD group limit is configured using the system-max mld-max-group-addr command. The snoop mcache entry limit is configured using the system-max mld-snoop-mcache command.
get-mem	The current memory allocation. This number should continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. NOTE: This number can be increased. (More memory can be allocated if necessary.)
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying status of MLD snooping traffic

To display status information for MLD snooping traffic, enter the **ipv6 mld-snooping traffic** command.

```
Brocade#show ipv6 mld-snooping traffic
MLD snooping: Total Recv: 32208, Xmit: 166
Q: query, Qry: general Q, G-Qry: group Q, GSQry: group-source Q, Mbr: member
Recv      QryV1      QryV2      G-Qry      GSQry      MbrV1      MbrV2      Leave
VL1        0          0          0          0        31744        208        256
VL70       0          0          0          0          0          0          0
Recv      IsIN      IsEX      ToIN      ToEX      ALLOW      BLOCK      Pkt-Err
VL1       1473      31784      0          1          1          7          0
VL70      0          0          0          0          0          0          0
Send      QryV1      QryV2      G-Qry      GSQry      MbrV1      MbrV2
VL1        0          0         166         0          0          0
VL70       0          0          0          0          0          0
```

Syntax: show ipv6 mld-snooping traffic

The following table describes the information displayed by the **show ipv6 mld-snooping traffic** command.

Field	Description
Q	Query
Qry	General Query
QryV1	Number of general MLDv1 queries received or sent.
QryV2	Number of general MLDv2 snooping queries received or sent.
G-Qry	Number of group specific queries received or sent.
GSQry	Number of group source specific queries received or sent.
MBR	The membership report.
MbrV1	The MLDv1 membership report.
MbrV2	The MLDv2 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times additional source addresses were allowed on the interface.
BLK	Number of times sources were removed from an interface.
Pkt-Err	Number of packets having errors such as checksum errors.

Displaying MLD snooping information by VLAN

You can display MLD snooping information for all VLANs or for a specific VLAN. For example, to display MLD snooping information for VLAN 70, enter the **ipv6 mld-snooping vlan <num>** command.

```
Brocade#show ipv6 mld-snooping vlan 70
version=1, query-t=60, group-aging-t=140, max-resp-t=3, other-qr-present-t=123
VL70: cfg V2, vlan cfg passive, 2 grp, 0 (SG) cache, rtr ports,
  router ports: 1/36(120) fe80::2e0:52ff:fe00:9900,
  1/26 has 2 grp, non-QR (passive), cfg V1
  1/26 has 2 grp, non-QR (passive), cfg V1
  group: ff10:1234::5679, life = 100
  group: ff10:1234::5678, life = 100
  1/35 has 0 grp, non-QR (QR=fe80::2e0:52ff:fe00:9900, age=20), dft V2 trunk
```

Syntax: **show ipv6 mld-snooping vlan** [**<vlan-id>**]

If you do not specify a **vlan-id**, information for all VLANs is displayed.

The following table describes information displayed by the **show ipv6 mld-snooping vlan** command.

Field	Description
version	The MLD version number.
query-t	How often a querier sends a general query on the interface.
group-aging-t	Number of seconds membership groups can be members of this group before aging out.
rtr-port	The router ports which are the ports receiving queries. The display <code>router ports: 1/36(120) fe80::2e0:52ff:fe00:9900</code> means port 1/36 has a querier with fe80::2e0:52ff:fe00:9900 as the link-local address, and the remaining life is 120 seconds.
max-resp-t	The maximum number of seconds a client can wait before it replies to the query.
non-QR	Indicates that the port is a non-querier.
QR	Indicates that the port is a querier.

Clearing MLD snooping counters and mcache

The clear commands for MLD snooping should only be used in troubleshooting situations or when recovering from error conditions.

Clearing MLD counters on all VLANs

To clear MLD snooping error and traffic counters on all VLANs, enter the **ipv6 mld-snooping counters** command.

```
Brocade#clear ipv6 mld-snooping counters
```

Syntax: **clear ipv6 mld-snooping counters**

Clearing the mcache on all VLANs

To clear the mcache on all VLANs, enter the **clear ipv6 mld-snooping mcache** command.

```
Brocade#clear ipv6 mld-snooping mcache
```

Syntax: `clear ipv6 mld-snooping mcache`

Clearing the mcache on a specific VLAN

To clear the mcache on a specific VLAN, enter the **clear ipv6 mld-snooping vlan <num> mcache** command.

```
Brocade#clear ipv6 mld-snooping vlan 10 mcache
```

Syntax: `clear ipv6 mld-snooping vlan <vlan-id> mcache`

The <vlan-id> parameter specifies the specific VLAN from which to clear the cache.

Clearing traffic counters on a specific VLAN

To clear the traffic counters on a specific VLAN, enter the **clear ipv6 mld-snooping vlan <num> traffic** command.

```
Brocade#clear ipv6 mld-snooping vlan 10 traffic
```

Syntax: `clear ipv6 mld-snooping vlan <vlan-id> traffic`

The <vlan-id> parameter specifies the specific VLAN from which to clear the traffic counters.

MLD snooping configuration

MLD Snooping on FastIron WS and Brocade FCX and ICX Switches

[Table 274](#) lists the individual Brocade FastIron switches and the Multicast Listening Discovery (MLD) snooping features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images.

TABLE 274 Supported MLD snooping features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
MLD V1/V2 snooping (global and local)	This chapter describes MLD snooping on the FWS and FCX. For information about MLD snooping on other FastIron devices, see Chapter 37, “MLD Snooping on FastIron X Series Switches” .	Yes	Yes	Yes	Yes
MLD fast leave for V1		Yes	Yes	Yes	Yes
MLD tracking and fast leave for V2		Yes	Yes	Yes	Yes
Static MLD and IGMP groups with support for proxy		Yes	Yes	Yes	Yes

MLD snooping overview

The default method a device uses to process an IPv6 multicast packet is to broadcast it to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to CPU, which may result in some clients receiving unwanted traffic.

Multicast Learning Discovery (MLD) Snooping provides multicast containment by forwarding traffic only to those clients that have MLD receivers for a specific multicast group (destination address). The device maintains the MLD group membership information by processing MLD reports and generating messages so traffic can be forwarded to ports receiving MLD reports. This is analogous to IGMP Snooping on the Brocade Layer3 switches.

An IPv6 multicast address is a destination address in the range of FF00::/8. A limited number of multicast addresses are reserved. Since packets destined for the reserved addresses may require VLAN flooding, these devices do not snoop in the FFOX::000X range (where X is from 0 to F). Data packets destined to these addresses are flooded to the entire VLAN by hardware, and mirrored to CPU. Multicast data packets destined to addresses outside the FFOX::000X range are snooped. A client must send MLD reports in order to receive traffic. If an application outside the FFOX::000X range requires VLAN flooding, you must configure a static group for the entire VLAN.

An MLD device periodically broadcasts general queries, and sends group queries upon receiving a leave message to ensure no other clients at the same port still want this specific traffic before removing it. MLDv1 allows clients to specify which group (destination IPv6 address) on which to receive traffic. (MLDv1 cannot choose the source of the traffic.) MLDv2 deals with source-specific multicasts, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An MLDv2 device's port state can either be in INCLUDE or EXCLUDE mode. There are different types of group records for client reports.

The interfaces respond to general queries by sending a membership report containing one or more of the following records associated with a specific group:

- Current-state record - Indicates the sources from which the interface wants to receive or not receive traffic. This record contains the source addresses of the interfaces and whether or not traffic will be included (IS_IN) or excluded (IS_EX) from that source address.
- Filter-mode-change record - If the interface changes its current state from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if an interface current state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- MLDv1 leave report - Equivalent to a TO_IN (empty) record in MLDv2. This record means that no traffic from this group will be received regardless of the source.
- An MLDv1 group report - Equivalent to an IS_EX (empty) record in MLDv2. This record means that all traffic from this group will be received regardless of source.
- Source-list-change record - If the interface wants to add or remove traffic sources from its membership report, the report can include an ALLOW record, which contains a list of new sources from which the interface wishes to receive traffic. The report can also contain a BLOCK record, which lists current traffic sources from which the interface wants to stop receiving traffic.

MLD protocols provide a way for clients and a device to exchange messages, and allow the device to build a database indicating which port wants what traffic. Since the MLD protocols do not specify forwarding methods, MLD Snooping or multicast protocols such as IPv6 PIM-Sparse Mode (PIM SM) are required to handle packet forwarding. PIM SM can route multicast packets within and outside a VLAN, while MLD Snooping can switch packets only within a VLAN. These devices do not support PIM-SM routing.

If a VLAN is not MLD Snooping-enabled, it floods IPv6 multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, MLD packets are trapped to the CPU. Data packets are mirrored to the CPU and VLAN flooded. The CPU then installs hardware resources so subsequent data packets can be hardware-switched to desired ports without going through the CPU. If there is no client report, the hardware resource drops the data stream. The hardware can either match group addresses only (* G), or both source and group (S G) addresses in the data stream. If MLDv2 is configured in any port of a VLAN, the VLAN uses an (S G) match, otherwise it uses (* G). Because the hardware can match only the lowest 32 bits of a 128 bit IPv6 address, the output interfaces (OIF) of a hardware resource are the superset of the OIF of all data streams sharing the same lowest 32 bits. For example, if groups ff10::1234:5678:abcd and ff20::5678:abcd share the same hardware resource, then the OIF of the hardware matching (* 5678:abcd) is the superset of these two groups.

Stackable devices allocate 16K of hardware resources for MAC learning, IGMP, and MLD snooping. If a data packet does not match any of these resources, it might be sent to the CPU, increasing the CPU burden. This can happen if the device runs out of hardware resources, or is unable to install a resource for a specific matching address due to a hashing collision. Because the hardware hashes addresses into 16K entries, some addresses may be hashed into the same entry. If the collision number in an entry is more than the hardware chain length, the resource cannot be installed. The chain length can be configured using the **hash-chain-length** command, as follows.

```
Brocade(config)#hash-chain-length 8
```

Syntax: [no] **hash-chain-length** <num>

The `<num>` parameter range is 4 to 32, in multiples of 4. If the input value is not a multiple of 4, then it will be changed to the multiple of 4 lower than the input value (e.g. 11 will be changed to 8). The default hash chain length is 4. A chain length of more than 4 may affect line rate switching.

NOTE

For this command to take effect, you must save the configuration and reload the switch.

The hardware resource limit applies only to snooping-enabled VLANs. In VLANs where snooping is not enabled, multicast streams are switched in hardware without using any pre-installed resources.

The Brocade device supports up to 32K of MLD groups. They are produced by client membership reports.

Configuration notes for MLD snooping

- Servers (traffic sources) are not required to send MLD memberships.
- The default MLD version is V1.
- Hardware resources are installed only when there is data traffic. If a VLAN is configured for MLDv2, the hardware matches (S G), otherwise it matches (* G).
- You can configure the maximum number of groups and hardware-switched data streams.
- The device supports static groups applying to the entire VLAN, or to specific ports. The device acts as a proxy to send MLD reports for the static groups when receiving queries.
- A user can configure static router ports, forcing all multicast traffic to be sent to these ports.
- All devices support fast leave for MLDv1, which stops traffic immediately to any port that has received a leave message.
- All devices support tracking and fast leave for MLDv2, which tracks all MLDv2 clients. If the only client on a port leaves, traffic is stopped immediately.
- An MLD device can be configured as a querier (active) or non-querier (passive). Queriers send queries. Non-queriers listen for queries and forward them to the entire VLAN.
- Every VLAN can be independently configured as a querier or a non-querier.
- A VLAN that has a connection to an IPv6 PIM-enabled port on another router should be configured as a non-querier. When multiple snooping devices connect together and there is no connection to IPv6 PIM ports, only one device should be configured as the querier. If multiple devices are configured as active, only one will continue to send queries after the devices have exchanged queries. Refer to [“Configuring queriers and non-queriers on MLD snooping-enabled devices”](#) on page 1634.
- An MLD device can be configured to rate-limit the forwarding of MLDv1 membership reports to queriers.
- Because these devices use an IPv6 link-local address as the source address when sending queries, no global address is required.

The MLD implementation allows snooping on some VLANs or on all VLANs. MLD can be enabled or disabled independently for each VLAN. In addition, individual ports of a VLAN can be configured as MSLv1 and MLDv2. In general, global configuration commands such as **ipv6 mld-snooping...** apply to all VLANs except those with a local **mld-snooping..** configuration, which supersedes the global configuration. Configuring the version on a port or a VLAN only affects the device sent query version. The device always processes all versions of client reports regardless of the version configured.

MLD Snooping requires hardware resources. If the device has insufficient resources, the data stream without a resource is mirrored to the CPU in addition to being VLAN flooded, which can cause high CPU usage. To avoid this situation, Brocade recommends that you avoid enabling snooping globally unless necessary.

When any port of a VLAN is configured for MLDv2, the VLAN matches both source and group (S G) in hardware switching. If no port is configured for MLDv2, the VLAN matches group only (* G). Matching (S G) requires more hardware resources than (* G) when there are multiple servers sharing the same group. For example, two data streams from different sources to the same group require two (S G) entries in MLDv2, compared to only one (* G) in MLDv1. Brocade recommends that you use MLDv2 only in a source-specific application. Because each VLAN can be configured for the version independently, some VLANs might match (* G) while others match (S G).

To receive data traffic, MLD Snooping requires clients to send membership reports. If a client does not send reports, you must configure a static group to force traffic to client ports. The static group can either apply to some ports or to the entire VLAN.

Configuring queriers and non-queriers on MLD snooping-enabled devices

An MLD Snooping-enabled device can be configured as a querier (active) or non-querier (passive). An MLD querier sends queries; a non-querier listens for MLD queries and forwards them to the entire VLAN. VLANs can be independently configured as queriers or non-queriers. If a VLAN has a connection to an IPv6 PIM-enabled port on another router, the VLAN should be configured as a non-querier. When multiple MLD snooping devices are connected together, and there is no connection to an IPv6 PIM-enabled port, one of the devices should be configured as a querier. If multiple devices are configured as queriers, after multiple devices exchange queries, then all devices except the winner (the device with the lowest address) stop sending queries. Although the system works when multiple devices are configured as queriers, Brocade recommends that only one device, preferably the one with the traffic source, is configured as the querier.

Because non-queriers always forward multicast data traffic and MLD messages to router ports which receive MLD queries or IPv6 PIM hellos, Brocade recommends that you configure the devices with the data traffic source (server) as queriers. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether or not there are clients on the querier.

NOTE

In a topology with one or more connected devices, at least one device must be running PIM, or configured as active. Otherwise, no devices can send queries, and traffic cannot be forwarded to clients.

MLD snooping configuration on VLANs

You can configure MLD snooping on some VLANs or all VLANs. Each VLAN can be independently enabled or disabled for MLD snooping, or can be configured with MLDv1 or MLDv2. In general, the **ipv6 mld-snooping...** commands apply globally to all VLANs except those configured with **VLAN-specific mld-snooping...** commands. **VLAN-specific mld-snooping** commands supersede **global ipv6 mld-snooping** commands.

MLD snooping using MLDv1 with MLDv2

MLD snooping can be configured as MLDv1 or MLDv2 on individual ports on a VLAN. An interface or router sends queries and reports that include the MLD version with which it has been configured. The version configuration applies only to the sending of queries. The snooping device recognizes and processes MLDv1 and MLDv2 packets regardless of the version configured.

NOTE

To avoid version deadlock, when an interface receives a report with a lower version than that for which it has been configured, the interface does **not** automatically downgrade the running MLD version.

MLD snooping configuration

Configuring MLD Snooping on Stackable devices consists of the following global and VLAN-specific tasks.

Global tasks for MLD Snooping

- Configuring hardware and software resource limits
- Disabling transmission and receipt of MLD packets on a port
- Configuring the MLD mode: active or passive (must be enabled for MLD Snooping)
- Modifying the age interval
- Specifying the interval for query messages (active MLD mode only)
- Specifying the global MLD version
- Enabling and disabling report control (rate limiting)
- Modifying the leave-wait time
- Modifying the mcache age interval
- Disabling error and warning messages

VLAN-specific tasks for MLD Snooping

- Configuring the MLD mode for the VLAN: active or passive
- Enabling or disabling MLD Snooping for the VLAN
- Configuring the MLD version for the VLAN
- Configuring the MLD version for individual ports
- Configuring static groups to the entire VLAN or some ports
- Configuring static router ports
- Enabling client tracking and the fast leave feature for MLDv2
- Configuring fast leave for MLDv1
- Configuring fast-convergence

Configuring the hardware and software resource limits

The system supports up to 8K of hardware-switched multicast streams. The configurable range is from 256 to 8192 and the default is 512. However, for ICX 6430, the range is from 256 to 1024, and the default is 256, while for ICX 6450, the range is from 256 to 8192 with the default as 512. Enter the **system-max mld-snoop-mcache** *<num>* command to define the maximum number of MLD Snooping cache entries.

```
Brocade(config)#system-max mld-snoop-mcache 8000
```

Syntax: [no] **system-max mld-snoop-mcache** *<num>*

The system supports up to 32K of groups. The configurable range is 256 to 32768 and the default is 8192. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limits are not processed.

Disabling transmission and receipt of MLD packets on a port

When a VLAN is snooping-enabled, all MLD packets are trapped to the CPU without hardware VLAN flooding. The CPU can block MLD packets to and from a multicast-disabled port, and will not add that port to the output interfaces or hardware resources, which prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic for these groups is flooded to the entire VLAN, including to the disabled ports. Since the hardware cannot block traffic from disabled ports, hardware traffic is switched in the same way as traffic from enabled ports.

NOTE

This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

```
Brocade(config)#interface ethernet 0/1/3
Brocade(config-if-e1000-0/1/3)#ipv6-multicast-disable
```

Syntax: [no] **ipv6-multicast-disable**

Configuring the global MLD mode

You can configure a device for either active or passive (default) MLD mode. If you specify an MLD mode for a VLAN, the MLD mode overrides the global setting:

- **Active** – In active MLD mode, the device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- **Passive** – In passive MLD mode, the device forwards reports to the router ports which receive queries. MLD Snooping in passive mode does not send queries, but does forward queries to the entire VLAN.

To globally set the MLD mode to active for the device, enter the **ipv6 mld-snooping active** command.

```
Brocade(config)#ipv6 mld-snooping active
```

Syntax: [no] **ipv6 mld-snooping** [active | passive]

Omitting both the **active** and **passive** keywords is the same as entering **ipv6 mld-snooping passive**.

Modifying the age interval

When the device receives a group membership report, it makes an entry in the MLD group table for the group in the report. The age interval specifies how long the entry can remain in the table without the device receiving another group membership report. When multiple devices connect together, all devices should be configured with the same age interval. The age interval should be at least twice that of the query interval, so that missing one report will not stop traffic. For a non-querier, the query interval should equal that of the querier.

To modify the age interval, enter the **ipv6 mld-snooping age-interval <num>** command.

```
Brocade(config)#ipv6 mld-snooping age-interval 280
```

Syntax: [no] **ipv6 mld-snooping age-interval <interval>**

The **<interval>** parameter specifies the aging time. You can specify a value from 20 – 7200 seconds. The default is 140 seconds.

Modifying the query interval (Active MLD snooping mode only)

If the MLD mode is set to active, you can modify the query interval, which specifies how often the device sends group membership queries. When multiple queriers connect together, all queriers should be configured with the same interval.

To modify the query interval, enter the **ipv6 mld-snooping query-interval <num>** command.

```
Brocade(config)#ipv6 mld-snooping query-interval 120
```

Syntax: [no] **ipv6 mld-snooping query-interval <interval>**

The **<interval>** parameter specifies the interval between queries. You can specify a value from 10 – 3600 seconds. The default is 60 seconds.

Configuring the global MLD version

The default version is MLDv1. You can specify the global MLD version on the device as either MLDv1 or MLDv2. For example, the **ipv6 mld-snooping version <num>** command configures the device to use MLDv2.

```
Brocade(config)#ipv6 mld-snooping version 2
```

Syntax: [no] **ipv6 mld-snooping version 1 | 2**

You can also specify the MLD version for individual VLANs, or individual ports within VLANs. If no MLD version is specified for a VLAN, then the globally configured MLD version is used. If an MLD version is specified for individual ports in a VLAN, those ports use that version instead of the version specified for the VLAN or the globally specified version. The default is MLDv1.

Configuring report control

When a device is in passive mode, it forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding for the same group to no more than once per 10 seconds. This rate limiting does not apply to the first report answering a group-specific query.

NOTE

This feature applies to MLDv1 only. The leave messages are not rate limited.

MLDv1 membership reports for the same group from different clients are considered to be the same, and are rate-limited. This alleviates the report storm caused by multiple clients answering the upstream router query. To enable report-control, use the **ipv6 mld-snooping report-control** command.

```
Brocade(config)#ipv6 mld-snooping report-control
```

Syntax: [no] ipv6 mld-snooping report-control

Modifying the wait time before stopping traffic when receiving a leave message

You can define the wait time before stopping traffic to a port when the device receives a leave message for that port. The device sends group-specific queries once per second to determine if any client on the same port still needs the group. The value range is from 1 to 5, and the default is 2. Due to the internal timer accuracy, the actual wait time is between n and $(n+1)$ seconds, where n is the configured value.

```
Brocade(config)#ipv6 mld-snooping leave-wait-time 1
```

Syntax: [no] ipv6 mld-snooping leave-wait-time <num>

Modifying the multicast cache (mcache) aging time

You can set the time for an mcache to age out when it does not receive traffic. The traffic is hardware-switched. One minute before an mcache is aged out, the device mirrors a packet of the mcache to the CPU to reset the age. If no data traffic arrives within one minute, the mcache is deleted. If you configure a lower value, the resource consumed by idle streams is quickly removed, but packets are mirrored to the CPU more frequently. Configure a higher value only when data streams are arriving consistently. The range is 60 to 3600 seconds, and the default is 60 seconds.

```
Brocade(config)#ipv6 mld-snooping mcache-age 180
```

Syntax: [no] ipv6 mld-snooping mcache-age <num>

Disabling error and warning messages

The device prints error or warning messages when it runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate limited. You can turn off these messages by entering the **ipv6 mld-snooping verbose-off** command.

```
Brocade(config)#ipv6 mld-snooping verbose-off
```

Syntax: [no] ipv6 mld-snooping verbose-off

Configuring the MLD mode for a VLAN

You can configure a VLAN for either the active or passive (default) MLD mode. The VLAN setting overrides the global setting:

- **Active** – In active MLD mode, a device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- **Passive** – In passive MLD mode, the device forwards reports to router ports which receive queries. MLD snooping in the passive mode does not send queries. However, it does forward queries to the entire VLAN.

To set the MLD mode for VLAN 20 to active, enter the following commands.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping active
```

Syntax: [no] mld-snooping active | passive

Disabling MLD snooping for the VLAN

When MLD snooping is enabled globally, you can disable it for a specific VLAN. For example, the following commands disable MLD snooping for VLAN 20. This setting overrides the global setting for VLAN 20.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping disable-mld-snoop
```

Syntax: [no] mld-snooping disable-mld-snoop

Configuring the MLD version for the VLAN

You can specify the MLD version for a VLAN. For example, the following commands configure VLAN 20 to use MLDv2.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping version 2
```

Syntax: [no] mld-snooping version 1 | 2

When no MLD version is specified, the globally-configured MLD version is used. If an MLD version is specified for individual ports, these ports use that version, instead of the version specified for the VLAN.

Configuring the MLD version for individual ports

You can specify the MLD version for individual ports in a VLAN. For example, the following commands configure ports 0/1/4, 0/1/5, 0/1/6 and 0/2/1 to use MLDv2. The other ports use the MLD version specified with the **mld-snooping version** command, or the globally configured MLD version.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping port-version 2 ethe 0/2/1 ethe 0/1/4 to
0/1/6
```

Syntax: [no]mld-snooping port-version 1 | 2 ethernet <stack-unit/slot/port> [ethernet <stack-unit/slot/port>] [to ethernet <stack-unit/slot/port>]

Configuring static groups to the entire VLAN or to individual ports

A snooping-enabled VLAN cannot forward multicast traffic to ports that do not receive MLD membership reports. To allow clients to send reports, you can configure a static group which applies to the entire VLAN, or to individual ports on the VLAN. The static group forwards packets to the static group ports even if they have no client membership reports. The static group for the entire VLAN is used in VLAN flooding because it uses fewer hardware resources than the static group for individual ports. Configure a static group for specific ports on VLAN 20 using commands similar to the following.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping static-group ff05::100 count 2 ethe 0/1/3
ethe 0/1/5 to 0/1/7
Brocade(config-vlan-20)#mld-snooping static-group ff10::200
```

Syntax: [no] mld-snooping static-group <ipv6-address> [count <num>] [<stack-unit/slot/port>]

The **ipv6-address** parameter is the IPv6 address of the multicast group.

The **count** is optional, which allows a contiguous range of groups. Omitting the count <num> is equivalent to the count being 1.

If there are no <stack-unit/slot/port> numbers, the static groups apply to the entire VLAN.

Configuring static router ports

A device always forwards all multicast control and data packets to router ports that receive queries. Although router ports are learned, you can configure static router ports to force multicast traffic to specific ports, even though these ports never receive queries. To configure static router ports, enter commands such as the following.

```
Brocade(config)#vlan 70
Brocade(config-vlan-70)#mld-snooping router-port e 0/1/4 to 0/1/5 e 0/1/8
```

Syntax: [no] mld-snooping router-port <stack-unit/slot/port>

Turning off static group proxy

A device with static groups configured acts as a proxy and sends membership reports for its static groups when it receives general or group-specific queries. When a static group configuration is removed, the group is deleted from active group table immediately. However, the device does not send leave messages to the querier. The querier should age the group out. The proxy activity can be turned off (the default is on). For example.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping proxy-off
```

Syntax: [no] mld-snooping proxy-off

Enabling MLDv2 membership tracking and fast leave for the VLAN

MLDv2 provides membership tracking and fast leave services to clients. In MLDv1, only one client per interface must respond to a router queries; leaving some clients invisible to the router, which makes it impossible for the device to track the membership of all clients in a group. In addition, when a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before stopping the traffic. You can configure the wait time with the **ipv6 mld-snooping leave-wait-time** command.

MLDv2 requires that every client respond to queries, allowing the device is able to track every client. When the tracking feature is enabled, the device immediately stops forwarding traffic to the interface if an MLDv2 client sends a leave message, and there is no other client. This feature requires the entire VLAN to be configured for MLDv2 and have no MLDv1 clients. If a client does not send a report during the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can track group membership only; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each is receiving traffic from different sources. Client A receives a traffic stream from (source_1, group1) and Client B receives a traffic stream from (source_2, group1). The device waits for the configured **leave-wait-time** before it stops the traffic because the two clients are in the same group. If the clients are in different groups, the waiting period is ignored and traffic is stopped immediately.

To enable tracking and fast leave for VLAN 20, enter the following commands.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping tracking
```

Syntax: [no] mld-snooping tracking

The membership tracking and fast leave features are supported for MLDv2 only. If a port or client is not configured for MLDv2, the **mld-snooping tracking** command is ignored.

Configuring fast leave for MLDv1

When a device receives an MLDv1 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic to this port. Configuring fast-leave-v1 allows the device to stop forwarding traffic to a port immediately upon receiving a leave message. The device does not send group-specific queries. It is important that no snooping ports have multiple clients. When two devices connect, the querier device should not be configured for fast-leave-v1 because the port to the non-querier device could have multiple clients. The number of queries and the waiting period (in seconds) can be configured using the **ipv6 mld-snooping leave-wait-time** command. The default is 2 seconds. To configure fast leave for MLDv1, use commands such as the following.

```
Brocade(config)#vlan 20
Brocade(config-vlan-20)#mld-snooping fast-leave-v1
```

Syntax: [no] mld-snooping fast-leave-v1

Enabling fast convergence

In addition to periodically sending general queries, an active (querier) device sends out general queries when it detects a new port. However, since it does not recognize the other device port-up event, the multicast traffic might still use the query-interval time to resume after a topology change. Configuring fast-convergence allows the device to listen to topology change events in L2 protocols, such as spanning tree, and send general queries to shorten the convergence time.

If the L2 protocol is unable to detect a topology change, the fast-convergence feature may not work. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this an optimization action, rather than a topology change. In this case, other devices will not receive topology change notifications and will be unable to send queries to speed up the convergence. The original spanning tree protocol does not recognize optimization actions, and fast-convergence works in all cases.

To enable fast-convergence, enter commands such as the following.

```
Brocade(config)#vlan 70
Brocade(config-vlan-70)#mld-snooping fast-convergence
```

Syntax: mld-snooping fast-convergence

Displaying MLD snooping information

You can display the following MLD Snooping information:

- MLD Snooping error information
- Information about VLANs
- Group and forwarding information for VLANs
- MLD memory pool usage
- Status of MLD traffic
- MLD information by VLAN

Displaying MLD snooping error information

To display information about possible MLD errors, enter commands such as the following.

```
Brocade#show ipv6 mld-snooping error
snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: show ipv6 mld-snooping error

The following table describes the output from the **show ipv6 mld-snooping error** command.

Field	Description
SW processed pkt	The number of IPv6 multicast packets processed by MLD snooping.
up-time	The time since the MLD snooping last occurred is enabled.

Displaying MLD group information

To display MLD group information, enter the **show ipv6 mld-snooping group** command.

```
Brocade#show ipv6 mld-snooping group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 263 grp, 263 grp-port, tracking_enabled
      group                                p-port ST QR life mode source
1      ff0e::ef00:a0e3                    0/1/7  N  Y  120  EX   0
2      ff01::1:f123:f567                  0/1/9  N  Y      IN   1
```

NOTE

In this example, an MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed MLD group information, enter the **show ipv6 mld-snooping group detail** command.

```
Brocade#show ipv6 mld-snooping group ff0e::ef00:a096 detail
Display group ff0e::ef00:a096 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                p-port ST QR life mode source
1      ff0e::ef00:a096                    0/1/7  N  Y  100  EX   0
      group: ff0e::ef00:a096, EX, permit 0 (source, life):
      life=100, deny 0:
```

If tracking and fast leave are enabled, you can display the list of clients for a particular group by entering the **show ipv6 mld-snooping group tracking** command.

```
Brocade#show ipv6 mld-snooping group ff0e::ef00:a096 tracking
Display group ff0e::ef00:a096 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                p-port ST QR life mode source
1      ff0e::ef00:a096                    0/1/7  N  Y   80  EX   0
      receive reports from 1 clients: (age)
      (fe80::1011:1213:1415 60)
```

Syntax: **show ipv6 mld-snooping group** [*<group-address>*] [**detail**] [**tracking**]

To receive a report for a specific multicast group, enter that group address for *<group-address>*.

Enter **detail** to display the source list of a specific VLAN.

Enter **tracking** for information on interfaces that are tracking-enabled.

The following table describes the information displayed by the **show ipv6 mld-snooping group** command.

Field	Description
group	The address of the IPv6 group (destination IPv6 address).
p-port	The physical port on which the group membership was received.
ST	Yes indicates that the MLD group was configured as a static group; No means it was learned from reports.

Field	Description
QR	Yes means the port is a querier port; No means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the port.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE if it does not receive an IS_EX or TO_EX message during a specified period of time. The default is 140 seconds. There is no <code>life</code> displayed in INCLUDE mode.
mode	The current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If the interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.
source	Identifies the source list that will be included or excluded on the interface. An MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from 0 (zero) source list, which actually means that all traffic sources are included.
group	If you requested a <i>detailed</i> report, the following information is displayed: <ul style="list-style-type: none"> • The multicast group address • The mode of the group • Sources from which traffic will be admitted (INCLUDE) or denied (EXCLUDE) on the interface. • The life of each source list. If you requested a <i>tracking/fast leave</i> report, the clients from which reports were received are identified.

Displaying MLD snooping mcache information

The MLD snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the **show ipv6 mld-snooping mcache** command.

```
Brocade#show ipv6 mld-snooping mcache
Example: (S G) cnt=: (S G) are the lowest 32 bits, cnt: SW proc. count
        OIF: 0/1/22 TR(0/1/32,0/1/33), TR is trunk, 0/1/32 primary, 0/1/33 output
vlan 1, has 2 cache
1  (abcd:ef50 0:100), cnt=121
   OIF: 0/1/11 0/1/9
   age=0s up-time=120m vidx=4130 (ref-cnt=1)
2  (abcd:ef50 0:101), cnt=0
   OIF: entire vlan
   age=0s up-time=0m vidx=8191 (ref-cnt=1)
vlan 70, has 0 cache
```

Syntax: show ipv6 mld-snooping mcache

The following table describes the output from the **ipv6 mld-snooping mcache** command.

Field	Description
(abcd:ef50 0:100):	The lowest 32 bits of source and group. It is displayed in XXXX:XXXX hex format. Here XXXX is a 16-bit hex number.
cnt	The number of packets processed in software. IPv6 packets are switched in software, causing this number to increase slowly.
OIF	Output interfaces. <i>Entire vlan</i> means that static groups apply to the entire VLAN.
age	The mcache age. The mcache is reset to 0 if traffic continues to arrive, otherwise it is aged out when it reaches the time defined by ipv6 mld-snooping mcache-age .

Field	Description
uptime	The up time of this mcache in minutes.
vidx	The vidx is shared among mcaches using the same output interfaces. The vidx specifies the output port list, which shows the index. Valid range is from 4096 to 8191.
ref-cnt	The number of mcaches using this vidx.

Displaying software resource usage for VLANs

To display information about the software resources used, enter the **show ipv6 mld-snooping resource** command.

```
Brocade#show ipv6 mld-snooping resource
          alloc in-use  avail get-fail    limit  get-mem  size init
mld group      512     9    503      0    32000    272    28  256
mld phy port   1024    16   1008      0   200000    279    21 1024
snoop group hash  512     9    503      0    59392    272    20  256
... Entries deleted
total pool memory 194432 bytes
has total 1 forwarding hash
Available vidx: 4061
```

Syntax: show ipv6 mld-snooping resource

The following table describes the output from the **show ipv6 mld-snooping resource** command.

Field	Description
alloc	The allocated number of units.
in-use	The number of units which are currently used.
avail	The number of available units.
get-fail	Displays the number of resource failures. NOTE: It is important to pay close attention to this field.
limit	The upper limit of this expandable field. The MLD group limit is configured using the system-max mld-max-group-addr command. The snoop mcache entry limit is configured using the system-max mld-snoop-mcache command.
get-mem	The number of memory allocation. This number should continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. NOTE: This number can be increased. More memory can be allocated if necessary.
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying status of MLD snooping traffic

To display status information for MLD snooping traffic, enter the **show ipv6 mld-snooping traffic** command.

```
Brocade#show ipv6 mld-snooping traffic
MLD snooping: Total Recv: 32208, Xmit: 166
Q: query, Qry: general Q, G-Qry: group Q, GSQry: group-source Q, Mbr: member
Recv      QryV1      QryV2      G-Qry      GSQry      MbrV1      MbrV2      Leave
VL1        0          0          0          0          31744      208        256
VL70       0          0          0          0          0          0          0
Recv      IsIN      IsEX      ToIN      ToEX      ALLOW      BLOCK      Pkt-Err
VL1       1473      31784      0          1          1          7          0
VL70      0          0          0          0          0          0          0
Send      QryV1      QryV2      G-Qry      GSQry      MbrV1      MbrV2
VL1        0          0          166        0          0          0
VL70       0          0          0          0          0          0
```

Syntax: show ipv6 mld-snooping traffic

The following table describes the information displayed by the **show ipv6 mld-snooping traffic** command.

This field	Displays
Q	Query
Qry	General Query
QryV1	Number of general MLDv1 queries received or sent.
QryV2	Number of general MLDv2 snooping queries received or sent.
G-Qry	Number of group specific queries received or sent.
GSQry	Number of group source specific queries received or sent.
MBR	The membership report.
MbrV1	The MLDv1 membership report.
MbrV2	The MLDv2 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times additional source addresses were allowed on the interface.
BLK	Number of times sources were removed from an interface.
Pkt-Err	Number of packets having errors such as checksum errors.

Displaying MLD snooping information by VLAN

You can display MLD snooping information for all VLANs or for a specific VLAN. For example, to display MLD snooping information for VLAN 70, enter the following command.

```
Brocade#show ipv6 mld-snooping vlan 70
version=1, query-t=60, group-aging-t=140, max-resp-t=3, other-qr-present-t=123
VL70: cfg V2, vlan cfg passive, 2 grp, 0 (SG) cache, rtr ports,
      router ports: 0/1/36(120) fe80::2e0:52ff:fe00:9900,
      0/1/26 has 2 grp, non-QR (passive), cfg V1
      0/1/26 has 2 grp, non-QR (passive), cfg V1
      group: ff10:1234::5679, life = 100
      group: ff10:1234::5678, life = 100
      0/1/35 has 0 grp, non-QR (QR=fe80::2e0:52ff:fe00:9900, age=20), dft V2 trunk
```

Syntax: `show ipv6 mld-snooping vlan [<vlan-id>]`

If you do not specify a vlan-id, information for all VLANs is displayed.

The following table describes information displayed by the `show ipv6 mld-snooping vlan` command.

This field	Displays
version	The MLD version number.
query-t	How often a querier sends a general query on the interface.
group-aging-t	Number of seconds membership groups can be members of this group before aging out.
rtr-port	The router ports which are the ports receiving queries. The display <code>router ports: 0/1/36(120) fe80::2e0:52ff:fe00:9900</code> means port 0/1/36 has a querier with fe80::2e0:52ff:fe00:9900 as the link-local address, and the remaining life is 120 seconds.
max-resp-t	The maximum number of seconds a client can wait before it replies to the query.
non-QR	Indicates that the port is a non-querier.
QR	Indicates that the port is a querier.

Clear MLD snooping commands

The clear commands for MLD snooping should only be used in troubleshooting situations or when recovering from error conditions.

Clearing MLD counters on VLANs

To clear MLD Snooping error and traffic counters on all VLANs, enter the `clear ipv6 mld-snooping counters` command.

```
Brocade#clear ipv6 mld-snooping counters
```

Syntax: `clear ipv6 mld-snooping counters`

Clearing MLD mcache

To clear the mcache on all VLANs, enter the `clear ipv6 mld-snooping mcache` command.

```
Brocade#clear ipv6 mld-snooping mcache
```

Syntax: `clear ipv6 mld-snooping mcache`

Clearing mcache on a specific VLAN

To clear the mcache on a specific VLAN, enter the **clear ipv6 mld-snooping vlan <vlan-id> mcache** command.

```
Brocade#clear ipv6 mld-snooping vlan 10 mcache
```

Syntax: **clear ipv6 mld-snooping vlan <vlan-id> mcache**

The <vlan-id> parameter specifies the specific VLAN from which to clear the cache.

Clearing traffic on a specific VLAN

To clear the traffic counters on a specific VLAN, enter the **clear ipv6 mld-snooping vlan <vlan-id> traffic** command.

```
Brocade#clear ipv6 mld-snooping vlan 10 traffic
```

Syntax: **clear ipv6 mld-snooping vlan <vlan-id> traffic**

The <vlan-id> parameter specifies the specific VLAN from which to clear the traffic counters.

VRRP and VRRP-E

Table 275 lists the individual Brocade FastIron switches and the Virtual Router Redundancy Protocol (VRRP) and Virtual Router Redundancy Protocol Extended (VRRP-E) features they support.

NOTE

VRRP is supported in the base Layer 3, edge Layer 3, and full Layer 3 codes. VRRP support in the base Layer 3 and edge Layer 3 code is the same as in the full Layer 3 code. VRRP-E is supported with premium and ADV FastIron devices that are running the edge Layer 3 or full Layer 3 code.

TABLE 275 Supported VRRP and VRRP-E features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6450
Virtual Router Redundancy Protocol (VRRP)	Yes	Yes	Yes	Yes	Yes
VRRP timer scaling	Yes	Yes	Yes	Yes	Yes
VRRP Extended (VRRP-E)	Yes	Yes	Yes	Yes	Yes
IPv6 VRRP-E	Yes*	No	Yes	Yes	No
IPv6 VRRP v3	Yes*	No	Yes	Yes	No
VRRP-E slow start timer	Yes	Yes	Yes	Yes	Yes
VRRP-E timer scale	Yes	Yes	Yes	Yes	Yes
Forcing a Master router to abdicate to a standby router	Yes	No	Yes	Yes	Yes
VRRP-E Extension for Server Virtualization	Yes	No	No	No	No

* Refers to support for only IPv6 modules for these devices.

This chapter describes how to configure Brocade Layer 3 switches with the following router redundancy protocols:

- Virtual Router Redundancy Protocol (VRRP) – The standard router redundancy protocol described in RFC 2338. The FastIron devices support VRRP version 2 (v2) and VRRP version 3 (v3). VRRP v2 supports the IPv4 environment, and VRRP v3 supports the IPv6 environment.
- VRRP Extended (VRRP-E) – An enhanced version of VRRP that overcomes limitations in the standard protocol. The FastIron devices support VRRP-E v2 and VRRP-E v3. VRRP-E v2 supports the IPv4 environment, and VRRP-E v3 supports the IPv6 environment.

NOTE

VRRP and VRRP-E are separate protocols. You cannot use them together.

NOTE

You can use a Brocade Layer 3 switch configured for VRRP with another Brocade Layer 3 switch or a third-party router that is also configured for VRRP. However, you can use a Brocade Layer 3 switch configured for VRRP-E only with another Brocade Layer 3 switch that also is configured for VRRP-E.

NOTE

The maximum number of supported VRRP or VRRP-E router instances is 254 for IPv4 environments. The maximum number of supported VRRP or VRRP-E router instances is 128 for IPv6 environments.

For a summary of how these two router redundancy protocols differ, refer to [“Comparison of VRRP and VRRP-E”](#) on page 1658.

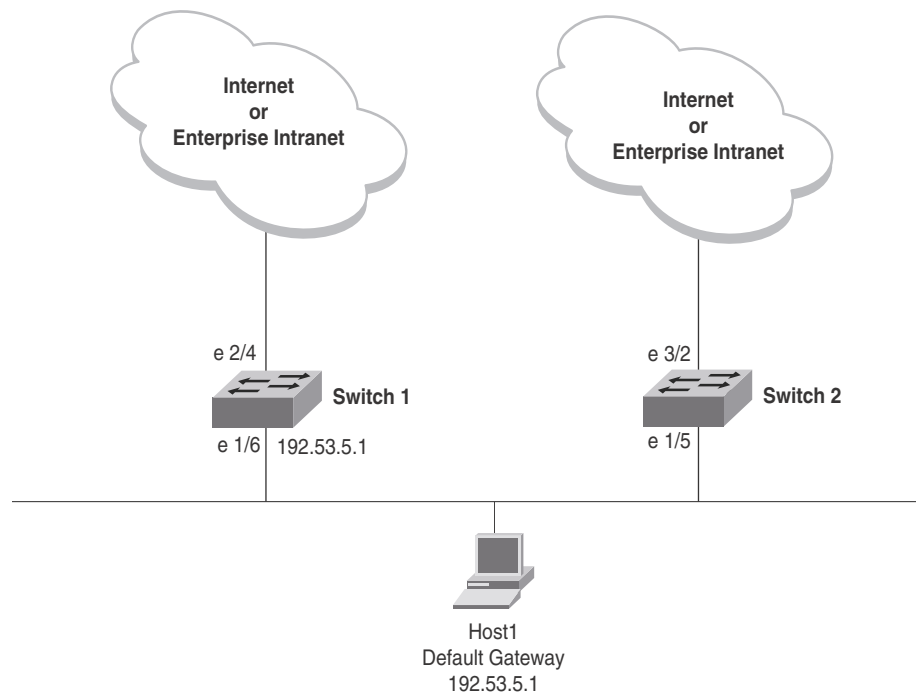
VRRP and VRRP-E overview

The following sections describe VRRP and VRRP-E. The protocols both provide redundant paths for IP addresses. However, the protocols differ in a few important ways. For clarity, each protocol is described separately.

VRRP overview

Virtual Router Redundancy Protocol (VRRP) provides redundancy to routers within a LAN. VRRP allows you to provide alternate router paths for a host without changing the IP address or MAC address by which the host knows its gateway. Consider the situation shown in [Figure 184](#).

FIGURE 184 Switch 1 is the Host1 default gateway but is a single point of failure

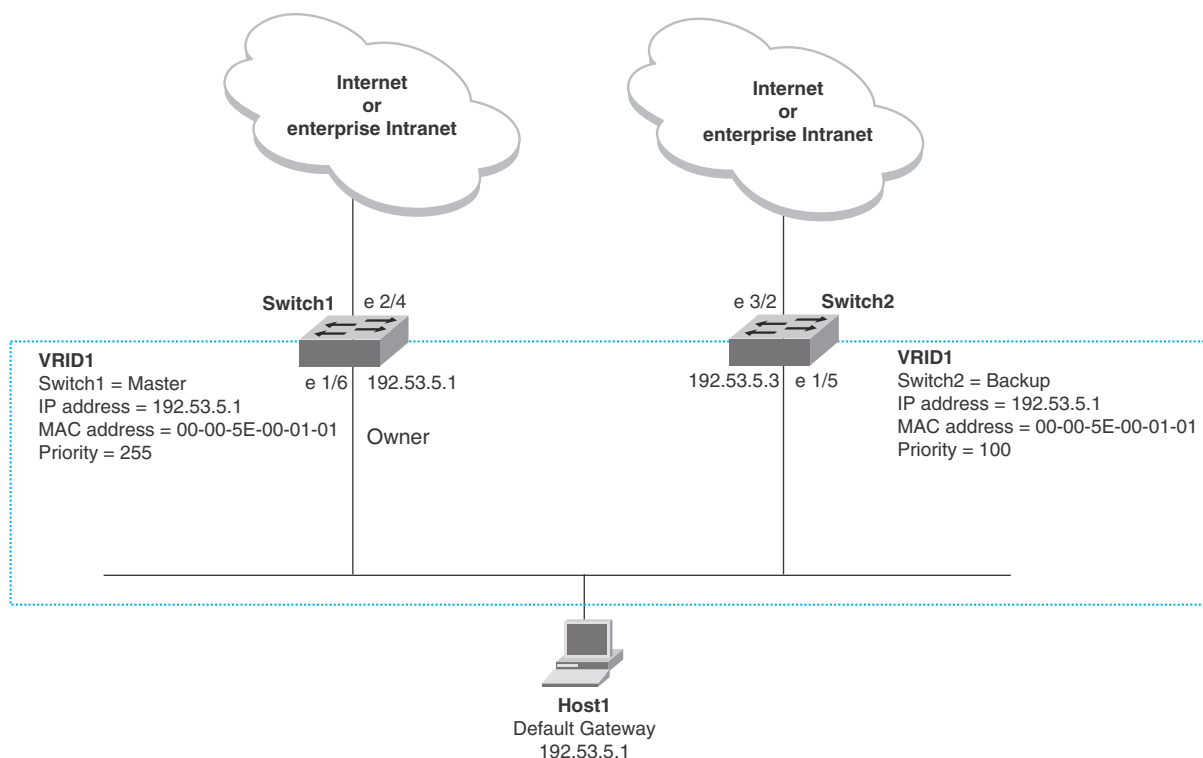


Switch 1 is the host default gateway out of the subnet. If this interface goes down, Host1 is cut off from the rest of the network. Switch 1 is thus a single point of failure for Host1's access to other networks.

If Switch 1 fails, you could configure Host1 to use Switch 2. Configuring one host with a different default gateway might not require too much extra administration. However, consider a more realistic network with dozens or even hundreds of hosts per subnet; reconfiguring the default gateways for all the hosts is impractical. It is much simpler to configure a VRRP virtual router on Switch 1 and Switch 2 to provide a redundant path for the hosts.

Figure 185 shows the same example network shown in Figure 184, but with a VRRP virtual router configured on Switch 1 and Switch 2.

FIGURE 185 Switch 1 and Switch 2 configured as VRRP virtual routers for redundant network access for Host1



The dashed box in Figure 185 represents a VRRP virtual router. When you configure a virtual router, one of the configuration parameters is the virtual router ID (VRID), which can be a number from 1 through 255. In this example, the VRID is 1.

NOTE

You can provide more redundancy by also configuring a second VRID with Switch 2 as the Owner and Switch 1 as the Backup. This type of configuration is sometimes called *Multigroup VRRP*.

Virtual router ID

A virtual router ID (VRID) consists of one Master router and one or more Backup routers. The Master router is the router that owns the IP addresses you associate with the VRID. For this reason, the Master router is sometimes called the “Owner”. Configure the VRID on the router that owns the default gateway interface. The other router in the VRID does not own the IP addresses associated with the VRID but provides the backup path if the Master router becomes unavailable.

Virtual router MAC address

Notice the MAC address associated with VRID1 in [Figure 185](#). The first five octets of the address are the standard MAC prefix for VRRP packets, as described in RFC 2338. The last octet is the VRID. The VRID number becomes the final octet in the virtual MAC address associated with the virtual router.

When you configure a VRID, the software automatically assigns its MAC address. When a VRID becomes active, the Master router broadcasts a gratuitous ARP request containing the virtual router MAC address for each IP address associated with the virtual router. In [Figure 185](#), Switch 1 sends a gratuitous ARP request with MAC address 00-00-5E-00-01-01 and IP address 192.53.5.1. Hosts use the virtual router MAC address in routed traffic they send to their default IP gateway (in this example, 192.53.5.1).

Virtual router IP address

VRRP does not use virtual IP addresses. Thus, there is no virtual IP address associated with a virtual router. Instead, you associate the virtual router with one or more real interface IP addresses configured on the router that owns the real IP addresses. In [Figure 185](#), the virtual router with VRID1 is associated with real IP address 192.53.5.1, which is configured on interface e1/6 on Switch 1. VRIDs are interface-level parameters, not system-level parameters, so the IP address you associate with the VRID must already be a real IP address configured on the Owner interface.

NOTE

You can associate a virtual router with a virtual interface. A virtual interface is a named set of physical interfaces.

When you configure the Backup router for the VRID, specify the same IP address as the one you specify on the Owner. This is the IP address used by the host as its default gateway. The IP address cannot also exist on the Backup router. The interface on which you configure the VRID on the Backup router must have an IP address in the same subnet.

NOTE

If you delete a real IP address used by a VRRP entry, the VRRP entry also is deleted automatically.

NOTE

When a Backup router takes over forwarding responsibilities from a failed Master router, the Backup forwards traffic addressed to the VRID MAC address, which the host believes is the MAC address of the router interface for its default gateway. However, the Backup router cannot reply to IP pings sent to the IP addresses associated with the VRID. Because the IP addresses are owned by the Owner, if the Owner is unavailable, the IP addresses are unavailable as packet destinations.

Master negotiation

The routers within a VRID use the VRRP priority values associated with each router to determine which router becomes the Master. When you configure the VRID on a router interface, you specify whether the router is the Owner of the IP addresses you plan to associate with the VRID or a Backup router. If you indicate that the router is the Owner of the IP addresses, the software automatically sets the router VRRP priority for the VRID to 255, the highest VRRP priority. The router with the highest priority becomes the Master.

Backup routers can have a priority from 3 through 254, which you assign when you configure the VRID on the Backup router interfaces. The default VRRP priority for Backup routers is 100.

Because the router that owns the IP addresses associated with the VRID always has the highest priority, when all the routers in the virtual router are operating normally, the negotiation process results in the Owner of the VRID IP addresses becoming the Master router. Thus, the VRRP negotiation results in the normal case, in which the host's path to the default route is to the router that owns the interface for that route.

Hello messages

Virtual routers use Hello messages for negotiation to determine the Master router. Virtual routers send Hello messages to IP Multicast address 224.0.0.18. The frequency with which the Master sends Hello messages is the Hello interval. Only the Master sends Hello messages. However, a Backup router uses the Hello interval you configure for the Backup router if it becomes the Master.

The Backup routers wait for a period of time called the dead interval for a Hello message from the Master. If a Backup router does not receive a Hello message by the time the dead interval expires, the Backup router assumes that the Master router is dead and negotiates with the other Backup routers to select a new Master router. The Backup router with the highest priority becomes the new Master.

Master and Owner backup routers

If the Owner becomes unavailable, but then comes back online, the Owner again becomes the Master router. The Owner becomes the Master router again because it has the highest priority. The Owner always becomes the Master again when the Owner comes back online.

NOTE

If you configure a track port on the Owner and the track port is down, the Owner priority is changed to the track priority. In this case, the Owner does not have a higher priority than the Backup router that is acting as the Master router and the Owner therefore does not resume its position as the Master router. For more information about track ports, refer to [“Track ports and track priority”](#) on page 1654.

By default, if a Backup is acting as the Master, and the original Master is still unavailable, another Backup can “preempt” the Backup that is acting as the Master. This can occur if the new Backup router has a higher priority than the Backup router that is acting as the Master. You can disable this behavior. When you disable preemption, a Backup router that has a higher priority than the router that is currently acting as the Master does not preempt the new Master by initiating a new Master negotiation. Refer to [“Backup preempt configuration”](#) on page 1678.

NOTE

Regardless of the setting for the preempt parameter, the Owner always becomes the Master again when it comes back online.

Track ports and track priority

The Brocade implementation of VRRP enhances the protocol by giving a VRRP router the capability to monitor the state of the interfaces on the other end of the route path through the router. For example, in [Figure 185](#) on page 1651, interface e1/6 on Switch 1 owns the IP address to which Host1 directs route traffic on its default gateway. The exit path for this traffic is through the Switch 1 e2/4 interface.

Suppose interface e2/4 goes down. Even if interface e1/6 is still up, Host1 is cut off from other networks. In conventional VRRP, Switch 1 would continue to be the Master router despite the unavailability of the exit interface for the path the router is supporting. However, if you configure interface e1/6 to track the state of interface e2/4, if e2/4 goes down, interface e1/6 responds by changing the Switch 1 VRRP priority to the value of the track priority. In the configuration shown in [Figure 185](#) on page 1651, the Switch 1 priority changes from 255 to 20. One of the parameters contained in the Hello messages the Master router sends to its Backup routers is the Master router priority. If the track port feature results in a change in the Master router priority, the Backup routers quickly become aware of the change and initiate a negotiation to become the Master router.

In [Figure 185](#) on page 1651, the track priority results in the Switch 1 VRRP priority becoming lower than the Switch 2 VRRP priority. As a result, when Switch 2 learns that it now has a higher priority than Switch 1, Switch 2 initiates negotiation to become the Master router and becomes the new Master router, thus providing an open path for the Host1 traffic. To take advantage of the track port feature, make sure the track priorities are always lower than the VRRP priorities. The default track priority for the router that owns the VRID IP addresses is 2. The default track priority for Backup routers is 1. If you change the track port priorities, make sure you assign a higher track priority to the Owner of the IP addresses than the track priority you assign on the Backup routers.

Suppression of RIP advertisements for backed-up interfaces

The Brocade implementation also enhances VRRP by allowing you to configure the protocol to suppress RIP advertisements for the backed-up paths from Backup routers. Normally, a VRRP Backup router includes route information for the interface it is backing up in RIP advertisements. As a result, other routers receive multiple paths for the interface and might sometimes unsuccessfully use the path to the Backup router rather than the path to the Master router. If you enable the Brocade implementation of VRRP to suppress the VRRP Backup routers from advertising the backed-up interface in RIP, other routers learn only the path to the Master router for the backed-up interface.

Authentication

The Brocade implementations of VRRP and VRRP-E can use simple passwords to authenticate VRRP and VRRP-E packets. VRRP-E can also use HMAC-MD5-96 to authenticate VRRP-E packets.

VRRP and VRRP-E authentication is configured on the router interfaces. The VRRP authentication configuration of every router interface must match. For example, if you want to use simple passwords to authenticate VRRP traffic within a router, you must configure VRRP simple password authentication with the same password on all of the participating router interfaces.

NOTE

The HMAC-MD5-96 authentication type is supported for VRRP-E, but not supported for VRRP.

NOTE

Authentication is not supported for VRRP v3.

Independent operation of VRRP alongside RIP, OSPF, and BGP4

VRRP operation is independent of RIP, OSPF, and BGP4; therefore, RIP, OSPF, and BGP4 are not affected if VRRP is enabled on one of these interfaces.

Dynamic VRRP configuration

All VRRP global and interface parameters take effect immediately. You do not need to reset the system to place VRRP configuration parameters into effect.

VRRP-E overview

The most important difference between VRRP and VRRP-E is that all VRRP-E routers are Backup routers; there is no Owner router. VRRP-E overcomes the limitations in standard VRRP by removing the Owner.

NOTE

VRRP-E is supported in the edge Layer 3 and full Layer 3 code only. It is not supported in the base Layer 3 code.

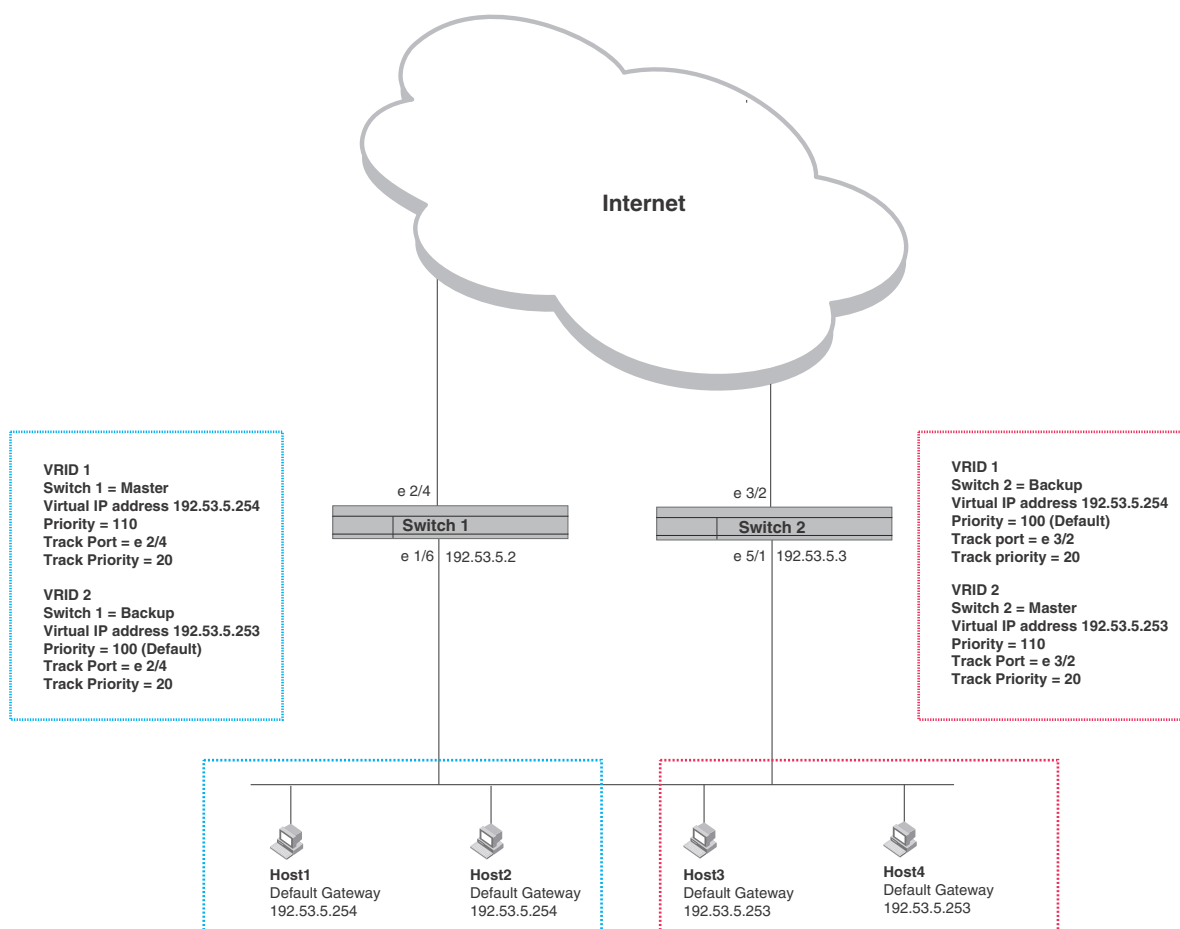
The following points explain how VRRP-E differs from VRRP:

- Owners and Backup routers
 - VRRP has an Owner and one or more Backup routers for each VRID. The Owner is the router on which the VRID's IP address is also configured as a real address. All the other routers supporting the VRID are Backup routers.
 - VRRP-E does not use Owners. All routers are Backup routers for a given VRID. The router with the highest priority becomes the Master. If there is a tie for highest priority, the router with the highest IP address becomes the Master. The elected Master owns the virtual IP address and answers pings and ARP requests.
- VRID's IP address
 - VRRP requires that the VRID's IP address also be a real IP address configured on the VRID's interface on the Owner.
 - VRRP-E requires only that the VRID be in the same subnet as an interface configured on the VRID's interface. VRRP-E does not allow you to specify a real IP address configured on the interface as the VRID IP address.

- VRID's MAC address
 - VRRP uses the source MAC address as a virtual MAC address defined as 00-00-5E-00-01-<vrid>, where <vrid> is the VRID. The Master owns the virtual MAC address.
 - VRRP-E uses the MAC address of the interface as the source MAC address. The MAC address is 02-04-80-<hash-value>-<vrid>, where <hash-value> is a two-octet hashed value for the IP address and <vrid> is the VRID.
- Hello packets
 - VRRP sends Hello messages to IP Multicast address 224.0.0.18.
 - VRRP-E uses UDP to send Hello messages in IP multicast messages. The Hello packets use the MAC address of the interface and the IP address as the source addresses. The destination MAC address is 01-00-5E-00-00-02, and the destination IP address is 224.0.0.2 (the well-known IP multicast address for "all routers"). Both the source and destination UDP port number is 8888. VRRP-E messages are encapsulated in the data portion of the packet.
- Track ports and track priority
 - VRRP changes the priority of the VRID to the track priority, which typically is lower than the VRID priority and lower than the VRID priorities configured on the Backup routers. For example, if the VRRP interface priority is 100 and a tracked interface with track priority 20 goes down, the software changes the VRRP interface priority to 20.
 - VRRP-E reduces the priority of a VRRP-E interface by the amount of a tracked interface priority if the tracked interface link goes down. For example, if the VRRP-E interface priority is 200 and a tracked interface with track priority 20 goes down, the software changes the VRRP-E interface priority to 180. If another tracked interface goes down, the software reduces the VRID priority again, by the amount of the tracked interface track priority.
- VRRP-E can use HMAC-MD5-96 for authenticating VRRP-E packets. VRRP can use only simple passwords.

Figure 186 shows an example of a VRRP-E configuration.

FIGURE 186 Switch 1 and Switch 2 are configured to provide dual redundant network access for the host



In this example, Switch 1 and Switch 2 use VRRP-E to load share as well as provide redundancy to the hosts. The load sharing is accomplished by creating two VRRP-E groups. Each group has its own virtual IP addresses. Half of the clients point to VRID 1's virtual IP address as their default gateway and the other half point to VRID 2's virtual IP address as their default gateway. This organization enables some of the outbound Internet traffic to go through Switch 1 and the rest to go through Switch 2.

Switch 1 is the Master router for VRID 1 (backup priority = 110) and Switch 2 is the Backup router for VRID 1 (backup priority = 100). Switch 1 and Switch 2 both track the uplinks to the Internet. If an uplink failure occurs on Switch 1, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the Internet is sent through Switch 2 instead.

Similarly, Switch 2 is the Master router for VRID 2 (backup priority = 110) and Switch 1 is the Backup router for VRID 2 (backup priority = 100). Switch 1 and Switch 2 are both tracking the uplinks to the Internet. If an uplink failure occurs on Switch 2, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the Internet is sent through Switch 1 instead.

ARP behavior with VRRP-E

In the VRRP-E implementation, the source MAC address of the gratuitous Address Resolution Protocol (ARP) request sent by the VRRP-E Master router is the VRRP-E virtual MAC address. When the router (either the Master or Backup router) sends an ARP request or reply packet, the sender's MAC address becomes the MAC address of the interface on the router. When an ARP request packet for the virtual router IP address is received by the Backup router, it is forwarded to the Master router to resolve the ARP request. Only the Master router answers the ARP request for the virtual router IP address.

Comparison of VRRP and VRRP-E

This section compares router redundancy protocols.

VRRP

VRRP is a standards-based protocol, described in RFC 2338. The Brocade implementation of VRRP contains the features in RFC 2338. The Brocade implementation also provides the following additional features:

- Track ports – A Brocade feature that enables you to diagnose the health of all the Layer 3 switch ports used by the backed-up VRID, instead of only the port connected to the client subnet. Refer to [“Track ports and track priority”](#) on page 1654.
- Suppression of RIP advertisements on Backup routers for the backed-up interface – You can enable the Layer 3 switches to advertise only the path to the Master router for the backed-up interface. Normally, a VRRP Backup router includes route information for the interface it is backing up in RIP advertisements.

Brocade Layer 3 switches configured for VRRP can interoperate with third-party routers using VRRP.

VRRP-E

VRRP-E is a Brocade protocol that provides the benefits of VRRP without the limitations. VRRP-E is unlike VRRP in the following ways:

- There is no “Owner” router. You do not need to use an IP address configured on one of the Layer 3 switches as the virtual router ID (VRID), which is the address you are backing up for redundancy. The VRID is independent of the IP interfaces configured in the Layer 3 switches. As a result, the protocol does not have an “Owner” as VRRP does.
- There is no restriction on which router can be the default Master router. In VRRP, the “Owner” (the Layer 3 switch on which the IP interface that is used for the VRID is configured) must be the default Master.

Brocade Layer 3 switches configured for VRRP-E can interoperate only with other Brocade Layer 3 switches.

Architectural differences between VRRP and VRRP-E

The protocols have the following architectural differences.

Management protocol

- VRRP – VRRP routers send VRRP Hello and Hello messages to IP Multicast address 224.0.0.18.
- VRRP-E – VRRP-E sends messages to destination MAC address 01-00-5E-00-00-02 and destination IP address 224.0.0.2 (the standard IP multicast address for “all routers”).

Virtual router IP address (the address you are backing up)

- VRRP – The virtual router IP address is the same as an IP address or virtual interface configured on one of the Layer 3 switches, which is the “Owner” and becomes the default Master.
- VRRP-E – The virtual router IP address is the gateway address you want to back up, but does not need to be an IP interface configured on one of the Layer 3 switch ports or a virtual interface.

Master and Backup routers

- VRRP – The “Owner” of the IP address of the VRID is the default Master and has the highest priority (255). The precedence of the Backup routers is determined by their priorities. The default Master is always the Owner of the IP address of the VRID.
- VRRP-E – The Master and Backup routers are selected based on their priority. You can configure any of the Layer 3 switches to be the Master by giving it the highest priority. There is no Owner.

VRRP and VRRP-E parameters

Table 276 lists the VRRP and VRRP-E parameters. Most of the parameters and default values are the same for both protocols. The exceptions are noted in the table.

TABLE 276 VRRP and VRRP-E parameters

Parameter	Description	Default	For more information
Protocol	The Virtual Router Redundancy Protocol (VRRP) based on RFC 2338 or VRRP-Extended, the Brocade-enhanced implementation of VRRP.	Disabled NOTE: Only one of the protocols can be enabled at a time.	page 1664 page 1668
VRRP or VRRP-E router	The Brocade Layer 3 switch active participation as a VRRP or VRRP-E router. Enabling the protocol does not activate the Layer 3 switch for VRRP or VRRP-E. You must activate the switch as a VRRP or VRRP-E router after you configure the VRRP or VRRP-E parameters.	Inactive	page 1664 page 1668
Virtual Router ID (VRID)	The ID of the virtual router you are creating by configuring multiple routers to back up an IP interface. You must configure the same VRID on each router that you want to use to back up the address.	None	page 1652 page 1664 page 1668
Virtual Router IP address	This is the address you are backing up. <ul style="list-style-type: none"> VRRP – The virtual router IP address must be a real IP address configured on the VRID interface on one of the VRRP routers. This router is the IP address Owner and is the default Master. VRRP-E – The virtual router IP address must be in the same subnet as a real IP address configured on the VRRP-E interface, but cannot be the same as a real IP address configured on the interface. 	None	page 1652 page 1664 page 1668
VRID MAC address	The source MAC address in VRRP or VRRP-E packets sent from the VRID interface, and the destination for packets sent to the VRID: <ul style="list-style-type: none"> VRRP – A virtual MAC address defined as 00-00-5E-00-01-<i><vrid></i> for IPv4 VRRP, and 00-00-5E-00-02-<i><vrid></i> for VRRP v3. The Master owns the virtual MAC address. VRRP-E – A virtual MAC address defined as 02-E0-52-<i><hash-value></i>-<i><vrid></i> for IPv4 VRRP-E and IPv6 VRRP-E, where <i><hash-value></i> is a two-octet hashed value for the IP address and <i><vrid></i> is the ID of the virtual router. 	Not configurable	page 1652

TABLE 276 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	For more information
Authentication type	<p>The type of authentication the VRRP or VRRP-E interfaces use to validate VRRP or VRRP-E packets.</p> <ul style="list-style-type: none"> No authentication – The interfaces do not use authentication. This is the VRRP default. Simple – The interface uses a simple text-string as a password in packets sent on the interface. If the interface uses simple password authentication, the VRID configured on the interface must use the same authentication type and the same password. HMAC-MD5-96 (VRRP-E only) – The interface uses HMAC-MD5-96 authentication for VRRP-E packets. <p>NOTE: HMAC-MD5-96 authentication is only supported for IPv4 or IPv6 VRRP-E. HMAC-MD5-96 is not supported by VRRP. Authentication is not supported for VRRP v3.</p>	No authentication	page 1654 page 1671
Router type	<p>Whether the router is an Owner or a Backup.</p> <ul style="list-style-type: none"> Owner (VRRP only) – The router on which the real IP address used by the VRID is configured. Backup – Routers that can provide routing services for the VRID but do not have a real IP address matching the VRID. 	<p>VRRP – The Owner is always the router that has the real IP address used by the VRID. All other routers for the VRID are Backups.</p> <p>VRRP-E – All routers for the VRID are Backups.</p>	page 1673
Backup priority	<p>A numeric value that determines a Backup router's preferability for becoming the Master for the VRID. During negotiation, the router with the highest priority becomes the Master.</p> <ul style="list-style-type: none"> VRRP – The Owner has the highest priority (255); other routers (backups) can have a priority from 3 through 254. VRRP-E – All routers are Backups and have the same priority by default. <p>If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.</p>	<p>VRRP v2 and IPv6 VRRP v3 - The value is 255 for the Owner and 100 for the Backups.</p> <p>VRRP-E v2 and IPv6 VRRP-E v3 - The value is 100 for all Backups.</p>	page 1673
Suppression of RIP advertisements	<p>A router that is running RIP normally advertises routes to a backed-up VRID even when the router is not currently the active router for the VRID. Suppression of these advertisements helps ensure that other routers do not receive invalid route paths for the VRID.</p> <p>NOTE: Suppression of RIP advertisements is not supported for VRRP v3 and VRRP-E v3.</p>	Disabled	page 1674
Hello interval	<p>The number of seconds or milliseconds between Hello messages from the Master to the Backups for a given VRID. The interval can be from 1 through 84 seconds for VRRP v2, VRRP-E v2, and IPv6 VRRP-E. The interval for VRRP v3 can be from 100 through 8400 milliseconds.</p>	<p>One second (VRRP v2 and VRRP-E v2, and IPv6 VRRP-E) 1000 milliseconds (VRRP v3).</p>	page 1653 page 1675

TABLE 276 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	For more information
Dead interval	<p>The number of seconds or milliseconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active.</p> <p>If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID.</p>	<p>The dead interval calculation for VRRP v6 or VRRP-E v6 is:</p> <p>Dead Interval: (3 X Hello Interval) + Skew Time</p> <p>Skew Time is 256 - Priority X Hello Interval / 256.</p> <p>For VRRP v3, the default is 3600 milliseconds. The configurable range is from 100 through 8400 milliseconds.</p> <p>For VRRP-E v3, the default is 3600 milliseconds. The configurable range is from 1 through 84 seconds.</p>	page 1653 page 1676
Backup Hello interval	<p>The number of seconds between Hello messages from a Backup to the Master.</p> <p>The message interval can be from 60 through 3600 seconds.</p> <p>You must enable the Backup to send the messages. The messages are disabled by default on Backups. The current Master (whether the VRRP Owner or a Backup) sends Hello messages by default.</p>	<p>Disabled</p> <p>60 seconds when enabled</p>	page 1653 page 1676
Track port	<p>Another Layer 3 switch port or virtual interface whose link status is tracked by the VRID interface.</p> <p>If the link for a tracked interface goes down, the VRRP or VRRP-E priority of the VRID interface is changed, causing the devices to renegotiate for the Master.</p> <p>NOTE: Track port is not supported by VRRP v3.</p>	None	page 1654 page 1677
Track priority	<p>A VRRP or VRRP-E priority value assigned to the tracked ports. If a tracked port link goes down, the VRID port VRRP or VRRP-E priority changes:</p> <ul style="list-style-type: none"> • VRRP – The priority changes to the value of the tracked port priority. • VRRP-E – The VRID port priority is reduced by the amount of the tracked port priority. <p>NOTE: Track priority is not supported by VRRP v3.</p>	<p>VRRP – 2</p> <p>VRRP-E – 5</p>	page 1654 page 1677
Backup preempt mode	Prevents a Backup with a higher VRRP priority from taking control of the VRID from another Backup that has a lower priority but has already assumed control of the VRID.	Enabled	page 1678
Timer scale	<p>Adjusts the timers for the Hello interval, Dead interval, Backup Hello interval, and Hold-down interval.</p> <p>NOTE: The timer scale is not supported for IPv6 VRRP v3.</p>	1	page 1678

TABLE 276 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	For more information
VRRP-E slow start timer	Causes a specified amount of time to elapse between the time the original Master is restored and when it takes over from the Backup. This interval allows time for OSPF convergence when the Master is restored. For VRRP-E only.	Disabled	page 1679
Short-path forwarding	Enables VRRP-E extension for server virtualization. If enabled, the traffic that is destined to the clients travels through the short-path forwarding path to reach the client (as shown in Figure 187 on page 1681). Any packets coming from the local subnet of the virtual IP address are routed to the VRRP-E Master router. For example, with VRRP-E extension for server virtualization enabled, the traceroute output shows one extra hop that displays the Backup router's interface IP address.	Disabled	page 1674

Note regarding disabling VRRP or VRRP-E

NOTE

Disabling VRRP or VRRP-E is supported by IPv4 VRRP v2, and IPv6 VRRP and IPv6 VRRP-E v3.

If you disable VRRP or VRRP-E, the Layer 3 switch removes all the configuration information for the disabled protocol from the running-config file. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

The CLI displays a warning message such as the following.

```
Brocade Router1(config-vrrp-router)#no router vrrp
router vrrp mode now disabled. All vrrp config data will be lost when writing to
flash!
```

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (for example, **router vrrp**). If you have already saved the configuration to the startup-config file and reloaded the software, the information is gone.

If you are testing a VRRP or VRRP-E configuration and are likely to disable and re-enable the protocol, you may want to make a backup copy of the startup-config file containing the protocol configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

Basic VRRP parameter configuration

To implement a simple VRRP configuration using all the default values, enter the commands shown in the following sections.

Configuration rules for VRRP

- The interfaces of all routers in a VRID must be in the same IP subnet.
- The IP addresses associated with the VRID must already be configured on the router that will be the Owner.
- An IP address associated with the VRID must be on only one router.
- The Hello interval must be set to the same value on the Owner and Backup routers for the VRID.
- The dead interval must be set to the same value on the Owner and Backup routers for the VRID.
- The track priority on a router must be lower than the router VRRP priority. Also, the track priority on the Owner must be higher than the track priority on the Backup routers.

NOTE

IPv6 VRRP and IPv6 VRRP-E v3 are supported on Brocade FESX, FSX 800, FSX 1600, and FCX IPv6 modules. IPv4 VRRP v3 is not supported on FCX devices.

NOTE

When you use the **router vrrp** command or the **ipv6 router vrrp** command to enter the VRRP configuration mode, the command prompt does not change and results in the following general configuration command prompt: `Brocade(config)#`. This differs from entering the VRRP extended mode, where entering the **router vrrp-extended** command results in a command prompt such as the following: `(config-VRRP-E-router)#`. For IPv6 VRRP extended mode, when entering the **ipv6 router vrrp-extended** command, this results in a command prompt such as the following: `(config-ipv6-VRRP-E-router)#`.

Configuring the Owner for IPv4 VRRP

To configure the VRRP Owner router for IPv4, enter the following commands on the router.

```
Brocade Router1(config)#router vrrp
Brocade Router1(config)#interface ethernet 1/6
Brocade Router1(config-if-1/6)#ip-address 192.53.5.1
Brocade Router1(config-if-1/6)#ip vrrp vrid 1
Brocade Router1(config-if-1/6-vrid-1)#owner
Brocade Router1(config-if-1/6-vrid-1)#ip-address 192.53.5.1
Brocade Router1(config-if-1/6-vrid-1)#activate
```

Syntax: [no] router vrrp

Syntax: [no] ip-address <ip-addr>

Syntax: [no] ip vrrp vrid <num>

Syntax: [no] owner [track-priority <value>]

Syntax: [no] activate

The `<ip-addr>` variable specifies the IPv4 address of the Owner router.

The IP address you assign to the Owner must be an IP address configured on an interface that belongs to the virtual router.

The `<num>` variable specifies the virtual router ID.

The **track-priority** `<value>` option changes the track-port priority for this interface and the VRID from the default (2) to a value from 1 through the maximum VRID supported by the device.

Configuring the Owner for IPv6 VRRP

To configure the VRRP Owner router for IPv6, enter the following commands on the router.

NOTE

You must first configure the **ipv6 unicast-routing** command at the global configuration level to enable IPv6 VRRP on the router.

```
Brocade Router1(config)# ipv6 unicast-routing
Brocade Router1(config)# ipv6 router vrrp
Brocade Router1(config)# interface ethernet 1/6
Brocade Router1(config-if-e10000-1/6)# ipv6-address 3013::1/64
Brocade Router1(config-if-e10000-1/6)# ipv6 vrrp vrid 1
Brocade Router1(config-if-e10000-1/6-vrid-1)# owner
Brocade Router1(config-if-e10000-1/6-vrid-1)# ipv6-address 3013::1
Brocade Router1(config-if-e10000-1/6-vrid-1)# activate
VRRP router 1 for this interface is activating
```

Syntax: `[no] ipv6 unicast-routing`

Syntax: `[no] ipv6 router vrrp`

Syntax: `[no] ipv6-address <ipv6-address>`

Syntax: `[no] ipv6 vrrp vrid <num>`

Syntax: `[no] owner`

Syntax: `[no] activate`

The `<num>` variable specifies the virtual router ID.

The `<ipv6-addr>` variable specifies the IPv6 address of the Owner router.

The IP address you assign to the Owner must be an IP address configured on an interface that belongs to the virtual router.

The **ipv6 router vrrp** command enables IPv6 VRRP v3 routing on the interface. All IPv6 VRRP router instances for a VRID are also enabled on the interface.

When the **no ipv6 router vrrp** command is enabled, all IPv6 VRRP router instances for a specific VRID are deleted from the interface, and the running configuration is lost when writing to flash. You must enable the **write memory** command to save your configuration. The following message is displayed when the **no ipv6 router vrrp** command is enabled.

```
Router1(config)#no ipv6 router vrrp
ipv6 router vrrp is disabled. All vrrp (ipv6) config data will be lost when
writing to flash!!
```

Configuring a Backup for IPv4 VRRP

To configure the VRRP Backup router for IPv4, enter the following commands.

```
Brocade Router2(config)#router vrrp
Brocade Router2(config)#interface ethernet 1/5
Brocade Router2(config-if-1/5)#ip-address 192.53.5.3
Brocade Router2(config-if-1/5)#ip vrrp vrid 1
Brocade Router2(config-if-1/5-vrid-1)#backup
Brocade Router2(config-if-1/5-vrid-1)#advertise backup
Brocade Router2(config-if-1/5-vrid-1)#ip-address 192.53.5.1
Brocade Router2(config-if-1/5-vrid-1)#activate
VRRP router 2 for interface is activating
```

Syntax: [no] router vrrp

Syntax: [no] ip-address <ip-addr>

Syntax: [no] ip vrrp vrid <num>

Syntax: [no] backup [priority <value>] [track-priority <value>]

Syntax: [no] advertise backup

Syntax: [no] activate

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

The <num> variable specifies the virtual router ID.

The **priority** <value> option specifies the VRRP priority for this virtual router. You can specify a value from 3 through 254. The default is 100.

The **track-priority** <value> option specifies that VRRP monitors the state of the interface. You can specify a value from 3 through 254. The default is 100.

By default, Backup routers do not send Hello messages to advertise themselves to the Master. The **advertise backup** command is used to enable a Backup router to send Hello messages to the Master.

Configuring a Backup for IPv6 VRRP

To configure the VRRP Backup router for IPv6, enter the following commands.

```
Brocade Router2(config)# ipv6 router vrrp
Brocade Router2(config)# interface ethernet 1/5
Brocade Router2(config-if-e10000-1/5)# ipv6-address 3013::3/64
Brocade Router2(config-if-e10000-1/5)# ipv6 vrrp vrid 1
Brocade Router2(config-if-e10000-1/5-vrid-1)# backup
Brocade Router2(config-if-e10000-1/5-vrid-1)# advertise backup
Brocade Router2(config-if-e10000-1/5-vrid-1)# ipv6-address 3013::1
Brocade Router2(config-if-e10000-1/5-vrid-1)# activate
```

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

Syntax: [no] ipv6 router vrrp

Syntax: [no] ipv6-address <ipv6-addr>

Syntax: [no] ipv6 vrrp vrid <num>

Syntax: [no] backup [priority <value>]

Syntax: [no] advertise backup

Syntax: [no] activate

The <ipv6-addr> variable specifies the IPv6 address of the Backup router.

The <num> variable specifies the virtual router ID.

The **priority** <value> option specifies the IPv6 VRRP priority for this virtual Backup router. You can specify a value from 3 through 254. The default is 100.

By default, Backup routers do not send Hello messages to advertise themselves to the Master. The **advertise backup** command is used to enable a Backup router to send Hello messages to the Master.

Configuration considerations for IPv6 VRRP v3 and IPv6 VRRP-E v3 support on Brocade devices

Consider the following when enabling IPv6 VRRP v3 mode and IPv6 VRRP-E v3 mode on FCX devices:

- You can configure only one protocol (Layer 3 VSRP, VRRP, or VRRP-E) on a router at a single time. However, VRRP or VRRP-E can be configured with IPv4 and IPv6 concurrently on a router.
- Scale timer configuration does not affect timer values, nor does it scale timer values for virtual routers configured with sub-second time values for IPv6 VRRP and IPv4 VRRP v3 modes.
- Abdication of a VRRP Owner router in an IPv6 environment is not supported. Abdication of an Owner router is a Brocade-specific enhancement to VRRP. Abdication of an Owner router is possible by changing the Owner's priority, or by configuring track ports for an Owner router.
 - For IPv6 VRRP v3 only, the tracking port configuration is not allowed if the router is configured as the VRRP Owner. This conforms to RFC 5798.
 - For the IPv6 VRRP v3 Owner router only, the priority configuration is not allowed. The Owner router priority is always 255. This conforms to RFC 5798.
- Hitless switchover is not supported for IPv6 VRRP and IPv6 VRRP-E environments.
- Interoperability is not supported for a VRID when VRRP routers are configured as VRRP v2 or v3.
- Brocade does not recommend that you re-use the same VRID across IPv6 VRRP-E and IPv4 VRRP-E if they are in the same broadcast domain.
- There is no specified restriction for configuring VRRP or VRRP-E instances if they are within the maximum VRID range. The maximum number of supported VRRP or VRRP-E router instances is 254 for IPv4 environments. The maximum number of supported VRRP or VRRP-E router instances is 128 for IPv6 environments.

Basic VRRP-E parameter configuration

The following sections describe the configuration of the parameters specific to IPv4 and IPv6 VRRP-E.

Configuration rules for VRRP-E

Consider the following rules when configuring VRRP-E:

- The interfaces of all routers in a VRID must be in the same IP subnet.
- The IP address associated with the VRID cannot be configured on any of the Layer 3 switches.
- The Hello interval must be set to the same value on all the Layer 3 switches.
- The dead interval must be set to the same value on all the Layer 3 switches.
- The track priority for a VRID must be lower than the VRRP-E priority.

Configuring IPv4 VRRP-E

VRRP-E is configured at the interface level. To implement a simple IPv4 VRRP-E configuration using all the default values, enter commands such as the following on each Layer 3 switch.

```
Brocade Router2(config)#router vrrp-extended
Brocade Router2(config)#interface ethernet 1/5
Brocade Router2(config-if-1/5)#ip-address 192.53.5.3
Brocade Router2(config-if-1/5)#ip vrrp-extended vrid 1
Brocade Router2(config-if-1/5-vrid-1)#backup
Brocade Router2(config-if-1/5-vrid-1)#advertise backup
Brocade Router2(config-if-1/5-vrid-1)#ip-address 192.53.5.254
Brocade Router2(config-if-1/5-vrid-1)#activate
```

Syntax: [no] router vrrp-extended

Syntax: [no] ip-address <ip-address>

Syntax: [no] ip vrrp-extended vrid <vrid>

Syntax: [no] backup [priority <value>] [track-priority <value>]

Syntax: [no] advertise backup

Syntax: [no] activate

The <vrid> variable specifies the virtual router ID.

The <ip-addr> variable specifies the IPv4 address of the router.

You must identify a VRRP-E router as a Backup before you can activate the virtual router on a Brocade device. However, after you configure the virtual router, you can use the **backup** command to change its priority or track priority.

The **priority** <value> option specifies the IPv4 VRRP-E priority for this virtual Backup router. You can specify a value from 3 through 254. The default is 100.

The **track-priority** <value> option changes the track port priority of a Backup router. You can specify a value from 1 through 254. The default is 2.

NOTE

You also can use the **enable** command to activate the configuration. This command does the same thing as the **activate** command.

Configuring IPv6 VRRP-E

To implement an IPv6 VRRP-E configuration using all the default values, enter the following commands.

NOTE

You must first configure the **ipv6 unicast-routing** command at the global configuration level to enable IPv6 VRRP-E on the router.

```
Brocade Router2(config)# ipv6 unicast-routing
Brocade Router2(config)# ipv6 router vrrp-extended
Brocade Router2(config-ipv6-VRRP-E-router)# interface ethernet 1/5
Brocade Router2(config-if-e10000-1/5)# ipv6-address 3013::2/64
Brocade Router2(config-if-e10000-1/5)# ipv6 vrrp-extended vrid 1
Brocade Router2(config-if-e10000-1/5-vrid-1)# backup priority 50 track-priority
10
Brocade Router2(config-if-e10000-1/5-vrid-1)# ipv6-address 3013::99
Brocade Router2(config-if-e10000-1/5-vrid-1)# activate
```

Syntax: [no] **ipv6 unicast-routing**

Syntax: **ipv6 router vrrp-extended**

Syntax: [no] **ipv6-address** <ipv6-addr>

Syntax: **ipv6 vrrp-extended vrid** <vrid>

Syntax: [no] **backup** [priority <value>] [track-priority <value>]

Syntax: [no] **activate**

The <vrid> variable specifies the virtual router ID.

The <ipv6-addr> variable specifies the IPv6 address of the router.

You must identify a VRRP-E router as a Backup before you can activate the virtual router on a Brocade device. However, after you configure the virtual router, you can use the **backup** command to change its priority or track priority.

The **priority** <value> option specifies the IPv6 VRRP-E priority for this virtual Backup router. You can specify a value from 3 through 254. The default is 100.

The **track-priority** <value> option changes the track port priority of a Backup router. You can specify a value from 1 through 254. The default is 2.

NOTE

You also can use the **enable** command to activate the configuration. This command does the same thing as the **activate** command.

When the **no ipv6 router vrrp-extended** command is enabled, all IPv6 VRRP-E instances for a specific VRID are deleted from the interface, and the running configuration is lost when writing to flash. You must enable the **write memory** command to save your configuration. The following message is displayed when the **no ipv6 router vrrp-extended** command is enabled.

```
Brocade Router2(config)#no ipv6 router vrrp-extended
ipv6 router VRRP-E is disabled. All VRRP-E (ipv6) config data will be lost when
writing to flash!!
```

Additional VRRP and VRRP-E parameter configuration

You can modify the following VRRP and VRRP-E parameters on an individual VRID basis. These parameters apply to both protocols:

- Authentication type (if the interfaces on which you configure the VRID use authentication)
- Router type (Owner or Backup)

NOTE

For VRRP, change the router type only if you have moved the real IP address from one router to another or you accidentally configured the IP address Owner as a Backup.

For VRRP-E, the router type is always Backup. You cannot change the type to Owner.

- Suppression of RIP advertisements on Backup routes for the backed-up interface
- Hello interval
- Dead interval
- Backup Hello messages and message timer (Backup advertisement)
- Track port
- Track priority
- Backup preempt mode
- Timer scale
- VRRP-E slow start timer
- VRRP-E extension for server virtualization (short-path forwarding)

Refer to [“VRRP and VRRP-E parameters”](#) on page 1660 for a summary of the parameters and their defaults.

VRRP and VRRP-E authentication types

This section describes VRRP and VRRP-E authentication parameters.

Configuring authentication type

The Brocade implementation of VRRP and VRRP-E supports the following authentication types for authenticating VRRP and VRRP-E traffic:

- No authentication – The interfaces do not use authentication. This is the default for VRRP and VRRP-E.
- Simple – The interfaces use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the VRID configured on the interfaces must use the same authentication type and the same password.

VRRP-E also supports the following authentication type:

- HMAC-MD5-96 – The interfaces use HMAC-MD5-96 authentication for VRRP-E packets.

NOTE

HMAC-MD5-96 authentication is not supported for VRRP.

To configure the VRID interface on Switch 1 for simple password authentication using the password “ourpword”, enter the following commands.

Configuring Switch 1

```
Brocade Switch1(config)#inter e 1/6
Brocade Switch1(config-if-1/6)#ip vrrp auth-type simple-text-auth ourpword
```

VRRP syntax

Syntax: `auth-type no-auth | simple-text-auth <auth-data>`

The **auth-type no-auth** option indicates that the VRID and the interface it is configured on do not use authentication.

The **simple-text-auth <auth-data>** option indicates that the VRID and the interface it is configured on use a simple text password for authentication. The `<auth-data>` variable is the password. If you use this variable, make sure all interfaces on all the routers supporting this VRID are configured for simple password authentication and use the same password.

NOTE

For VRRP v3, authentication is deprecated by RFC 5768.

VRRP-E syntax

For IPv4 VRRP-E:

Syntax: `ip vrrp-extended auth-type no-auth | simple-text-auth <auth-data> | md5-auth [0 | 1] <key>`

For IPv6 VRRP-E:

Syntax: `ipv6 vrrp-extended auth-type no-auth | simple-text-auth <auth-data> | md5-auth [0 | 1] <key>`

The values for the **no-auth** and **simple-text-auth** <auth-data> options are the same as for VRRP.

The **md5-auth** option configures the interface to use HMAC-MD5-96 for VRRP-E authentication.

The <key> variable is the MD5 encryption key, which can be up to 64 characters long. The optional [0 | 1] parameter configures whether the MD5 password is encrypted, as follows:

- If you do not enter this parameter and enter the key as clear text, the key appears encrypted in the device configuration and command outputs.
- If you enter **0** and enter the key as clear text, the key appears as clear text in the device configuration and command outputs.
- If you enter **1** and enter the key in encrypted format, the key appears in encrypted format in the device configuration and command outputs.

Syslog messages for VRRP-E HMAC-MD5-96 authentication

If an interface is configured with HMAC-MD5-96 authentication, all VRRP-E packets received on this interface are authenticated with the HMAC-MD5-96 algorithm using the shared secret key configured on the interface.

If a packet is received that fails this HMAC-MD5-96 authentication check, the packet gets dropped. Additionally, if syslog is enabled, a syslog message is generated to notify the administrator about an authentication failure. The message includes the VRID received in the packet's VRRP message and the interface on which the packet was received. These syslog messages will be rate limited to 20 log messages within a span of 5 minutes, starting from the first packet received that fails the HMAC-MD5-96 authentication check.

VRRP router type

A VRRP interface is either an Owner or a Backup router for a given VRID. By default, the Owner becomes the Master. A Backup router becomes the Master only if the Master becomes unavailable.

A VRRP-E interface is always a Backup router for its VRID. The Backup router with the highest VRRP priority becomes the Master.

This section describes how to specify the interface type, how to change the type for VRRP, and how to set or change the interface VRRP or VRRP-E priority and track priority for the VRID.

NOTE

You can force a VRRP Master router to abdicate (give away control) of the VRID to a Backup router by temporarily changing the Master VRRP priority to a value less than the Backup. Refer to [“Forcing a Master router to abdicate to a Backup router”](#) on page 1683.

NOTE

The Owner type is not applicable to VRRP-E.

NOTE

The IP addresses you associate with the Owner must be real IP addresses on the interface on which you configure the VRID.

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

Configuring Router 1 as VRRP VRID Owner

To configure Router1 as a VRRP VRID Owner, enter the following commands.

```
Brocade Router1(config)#interface ethernet 1/6
Brocade Router1(config-if-1/6)#ip vrrp vrid 1
Brocade Router1(config-if-1/6-vrid-1)#owner
```

Configuring Router 2 as VRRP Backup

To configure Router2 as a VRRP Backup for the same VRID, and set its VRRP priority, enter the following commands.

```
Brocade Router2(config)#interface ethernet 1/5
Brocade Router2(config-if-1/5)#ip vrrp vrid 1
Brocade Router2(config-if-1/5-vrid-1)#backup priority 120
Brocade Router2(config-if-1/5-vrid-1)#advertise backup
```

Configuring an IPv6 VRRP v3 interface as a Backup for a VRID

To configure an IPv6 VRRP v3 interface as a Backup for a VRID, and set its VRRP priority and track priority, enter commands such as the following.

```
Brocade Router2(config)# interface ethernet 1/5
Brocade Router2(config-if-e10000-1/1)# ipv6 vrrp vrid 1
Brocade Router2(config-if-e10000-1/1-vrid-1)# backup priority 50 track-priority 10
Brocade Router2(config-if-1/1-vrid-1)#advertise backup
```

Configuring an IPv6 VRRP-E v3 interface as a Backup for a VRID

To configure an IPv6 VRRP-E v3 interface as a Backup for a VRID, and set its VRRP-E priority and track priority, enter commands such as the following.

```
Brocade Router2(config)#interface ethernet 1/1
Brocade Router2(config-if-1/1)#ipv6 vrrp-extended vrid 1
Brocade Router2(config-if-1/1-vrid-1)#backup priority 50 track-priority 10
Brocade Router2(config-if-1/1-vrid-1)#advertise backup
```

VRRP v2 and IPv6 VRRP v3 syntax

Syntax: **owner** [**track-priority** <value>]

The **track-priority** <value> option changes the track port priority for this interface and VRID from the default (2) to a value from 1 through 254.

Syntax: **backup** [**priority** <value>] [**track-priority** <value>]

The **priority** <value> option specifies the VRRP priority for this interface and VRID. You can specify a value from 3 through 254. The default is 100.

The **track-priority** <value> option is the same as with the **owner** [**track-priority** <value>] command.

VRRP-E v2 and IPv6 VRRP-E v3 syntax

Syntax: **backup** [**priority** <value>] [**track-priority** <value>]

The software requires you to identify a VRRP-E interface as a Backup for its VRID before you can activate the interface for the VRID. However, after you configure the VRID, you can use this command to change its priority or track priority. The option values are the same as for VRRP.

Suppression of RIP advertisements

NOTE

Suppression of RIPng advertisements on Backup routers for the backup interface is not supported by IPv6 VRRP v3 and IPv6 VRRP-E v3.

Normally, a VRRP or VRRP-E Backup includes route information for the virtual IP address (the backed-up interface) in RIP advertisements. As a result, other routers receive multiple paths for the backed-up interface and might sometimes unsuccessfully use the path to the Backup router rather than the path to the Master.

You can prevent the Backup routers from advertising route information for the backed-up interface by enabling suppression of the advertisements.

Suppressing RIP advertisements for the backed-up interface in Router 2

To suppress RIP advertisements for the backed-up interface in Router 2, enter the following commands.

```
Brocade Router2(config)#router rip
Brocade Router2(config-rip-router)#use-vrrp-path
```

Syntax: `use-vrrp-path`

The syntax is the same for VRRP and VRRP-E.

Hello interval configuration

The Master periodically sends Hello messages to the Backup routers. The Backup routers use the Hello messages as verification that the Master is still online. If the Backup routers stop receiving the Hello messages for the period of time specified by the dead interval, the Backup routers determine that the Master router is dead. At this point, the Backup router with the highest priority becomes the new Master router.

NOTE

The default dead interval is three times the Hello interval plus the Skew time. Generally, if you change the Hello interval, you also should change the dead interval on the Backup routers.

To change the Hello interval on the Master to 10 seconds, enter the following commands.

```
Brocade Router1(config)#interface ethernet 1/6
Brocade Router1(config-if-1/6)#ip vrrp vrid 1
Brocade Router1(config-if-1/6-vrid-1)#hello-interval 10
```

Syntax: `[no] hello-interval <seconds>`

The `<seconds>` variable specifies the Hello interval value from 1 through 84 seconds for VRRP v2, VRRP-E v2, and IPv6 VRRP-E. The default is 1 second.

To change the Hello interval on the Master to 200 milliseconds for IPv6 VRRP v3, enter the following commands.

```
Brocade Router1(config)# interface ethernet 1/6
Brocade Router1(config-if-1/6)# ipv6 vrrp vrid 1
Brocade Router1(config-if-1/6-vrid-1)# hello-interval 200
```

Syntax: `[no] hello-interval <milliseconds>`

The `<milliseconds>` variable can be from 100 through 8400 milliseconds. The default is 1000 milliseconds.

Dead interval configuration

The dead interval is the number of seconds a Backup router waits for a Hello message from the Master before determining that the Master is dead. When Backup routers determine that the Master is dead, the Backup with the highest priority becomes the new Master.

If the value for the dead interval is not configured, then the current dead interval is equal to three times the Hello interval plus the Skew time (where Skew time is equal to $(256 - \text{priority})$ divided by 256).

To change the dead interval on a Backup to 30 seconds, enter the following commands.

```
Brocade Router2(config)#interface ethernet 1/5
Brocade Router2(config-if-e1000-1/5)#ip vrrp vrid 1
Brocade Router2(config-if-e1000-1/5-vrid-1)#dead-interval 30
```

Syntax: `dead-interval <value>`

The `<value>` variable is from 1 through 84 seconds for VRRP v2 and VRRP-E v2. For other versions, the `<value>` variable is from 100 through 8400 milliseconds. The default is 3600 milliseconds.

NOTE

If the `dead-interval` command is not configured on a VRRP v3 interface, then a zero value is displayed in the output of the `show ipv6 VRRP-Extended` command.

Backup Hello message state and interval

NOTE

The `advertise backup` command is supported by IPv4 VRRP v2, and IPv6 VRRP v3 and IPv6 VRRP-E v3.

By default, Backup routers do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup router to send Hello messages to the Master, enter the following commands.

```
Brocade(config)#router vrrp
Brocade(config)#interface ethernet 1/6
Brocade(config-if-1/6)#ip vrrp vrid 1
Brocade(config-if-1/6-vrid-1)#advertise backup
```

Syntax: `[no] advertise backup`

When you enable a Backup to send Hello messages, the Backup sends a Hello message to the Master every 60 seconds by default. You can change the interval to be up to 3600 seconds. To change the Hello message interval, enter the following commands.

```
Brocade(config)#router vrrp
Brocade(config)#interface ethernet 1/6
Brocade(config-if-1/6)#ip vrrp vrid 1
Brocade(config-if-1/6-vrid-1)#backup-hello-interval 180
```

Syntax: `[no] backup-hello-interval <num>`

The `<num>` variable specifies the message interval and can be from 60 through 3600 seconds. The default is 60 seconds.

The syntax is the same for VRRP v2 and IPv6 VRRP v3, and VRRP-E v2 and IPv6 VRRP-E v3.

Track port configuration

NOTE

Track port is not supported by VRRP v3.

You can configure the VRID on one interface to track the link state of another interface on the Layer 3 switch. This capability is quite useful for tracking the state of the exit interface for the path for which the VRID is providing redundancy. Refer to [“Track ports and track priority”](#) on page 1654.

To configure interface 1/6 on Router 1 to track interface 2/4, enter the following commands.

```
Brocade Router1(config)#interface ethernet 1/6
Brocade Router1(config-if-1/6)#ip vrrp vrid 1
Brocade Router1(config-if-1/6-vrid-1)#track-port ethernet 2/4
```

Syntax: **track-port ethernet** [<slotnum>/<portnum> | **ve** <num>]

The syntax is the same for VRRP and VRRP-E.

Track priority configuration

NOTE

Track priority is not supported by IPv6 VRRP v3.

When you configure a VRID to track the link state of other interfaces, and one of the tracked interfaces goes down, the software changes the VRRP or VRRP-E priority of the VRID interface:

- For VRRP, the software changes the priority of the VRID to the track priority, which typically is lower than the VRID priority and lower than the VRID priorities configured on the Backups. For example, if the VRRP interface priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VRRP interface priority to 60.
- For VRRP-E, the software reduces the VRID priority by the amount of the priority of the tracked interface that went down. For example, if the VRRP-E interface priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VRRP-E interface priority to 40. If another tracked interface goes down, the software reduces the VRID priority again, by the amount of the tracked interface track priority.

The default track priority for an Owner for VRRP v2, IPv6 VRRP v3, and VRRP-E v2 and IPv6 VRRP-E v3 is 2. The default track priority for Backup routers is 1.

You enter the track priority as a value with the **owner** or **backup** command. Refer to [“Track port configuration”](#) on page 1677.

Syntax: **owner** [track-priority <value>]

Syntax: **backup** [priority <value>] [track-priority <value>]

The syntax is the same for VRRP and VRRP-E.

Backup preempt configuration

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the VRID. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, because the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

NOTE

In VRRP, regardless of the setting for the preempt parameter, the Owner always becomes the Master again when it comes back online.

To disable preemption on a Backup, enter commands such as the following.

```
Brocade Router1(config)#interface ethernet 1/6
Brocade Router1(config-if-1/6)#ip vrrp vrid 1
Brocade Router1(config-if-1/6-vrid-1)#non-preempt-mode
```

Syntax: [no] non-preempt-mode

The syntax is the same for VRRP and VRRP-E.

Changing the timer scale

NOTE

Changing the timer scale is supported for IPv4 VRRP v2, IPv4 VRRP-E v2, and IPv6 VRRP-E v3. It is not supported for IPv6 VRRP v3.

To achieve sub-second failover times, you can shorten the duration of all scale timers for VSRP, VRRP, and VRRP-E by adjusting the timer scale. The timer scale is a value used by the software to calculate the timers. By default, the scale value is 1. If you increase the timer scale, each timer's value is divided by the scale value. Using the timer scale to adjust timer values enables you to easily change all the timers while preserving the ratios among their values. [Table 277](#) shows timer scale values.

TABLE 277 Time scale values

Timer	Timer scale	Timer value
Hello interval	1	1 second
	2	0.5 seconds
Dead interval	1	3 seconds
	2	1.5 seconds

TABLE 277 Time scale values (Continued)

Timer	Timer scale	Timer value
Backup Hello interval	1	60 seconds
	2	30 seconds
Hold-down interval	1	2 seconds
	2	1 second

If you configure the device to receive its timer values from the Master, the Backup also receives the timer scale value from the Master.

To change the timer scale, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# scale-timer 2
```

This command changes the scale to 2. All VSRP, VRRP, and VRRP-E timer values will be divided by 2.

Syntax: **[no] scale-timer** <num>

The <num> variable specifies the multiplier. You can specify a timer scale from 1 through 10. However, Brocade recommends the timer scale of 1 or 2 for VRRP and VRRP-E.

NOTE

Be cautious when configuring the **scale-timer** command in a VRRP or VRRP-E scaled environment. VSRP, VRRP, and VRRP-E are time-sensitive protocols and system behavior cannot be predicted when the timers are scaled.

VRRP-E slow start timer

In a VRRP-E configuration, if a Master router goes down, the Backup router with the highest priority takes over after expiration of the dead interval. When the original Master router comes back up again, it takes over from the Backup router (which became the Master router when the original Master router went down). By default, this transition from Backup back to Master takes place immediately. However, you can configure the VRRP-E slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup. This interval allows time for OSPF convergence when the Master is restored.

To set the IPv4 VRRP-E slow start timer to 30 seconds, enter the following commands.

```
Brocade(config)#router vrrp-extended
Brocade(config-VRRP-E-router)#slow-start 30
```

To set the IPv6 VRRP-E slow start timer to 60 seconds, enter the following commands.

```
Brocade(config)#ipv6 router vrrp-extended
Brocade(config-ipv6-VRRP-E-router)#slow-start 60
```

Syntax: **[no] slow-start** <seconds>

The <seconds> variable specifies a value from 1 through 255.

If the Master subsequently comes back up again, the amount of time specified by the VRRP-E slow start timer elapses (in the IPv4 example, 30 seconds) before the Master takes over from the Backup.

The VRRP-E slow start timer is effective only if the VRRP-E Backup router detects another VRRP-E Master (Standby) router. It is not effective during the initial bootup. The slow start timer is effective on a Backup router if the priority of the Backup router is equal to the configured priority on the Backup state router.

NOTE

The VRRP-E slow start timer applies only to VRRP-E configurations. It does not apply to VRRP configurations.

VRRP-E Extension for Server Virtualization

VRRP-E is enhanced with the VRRP-E Extension for Server Virtualization feature so that the Brocade device attempts to bypass the VRRP-E Master router and directly forward packets to their destinations through interfaces on the Backup router.

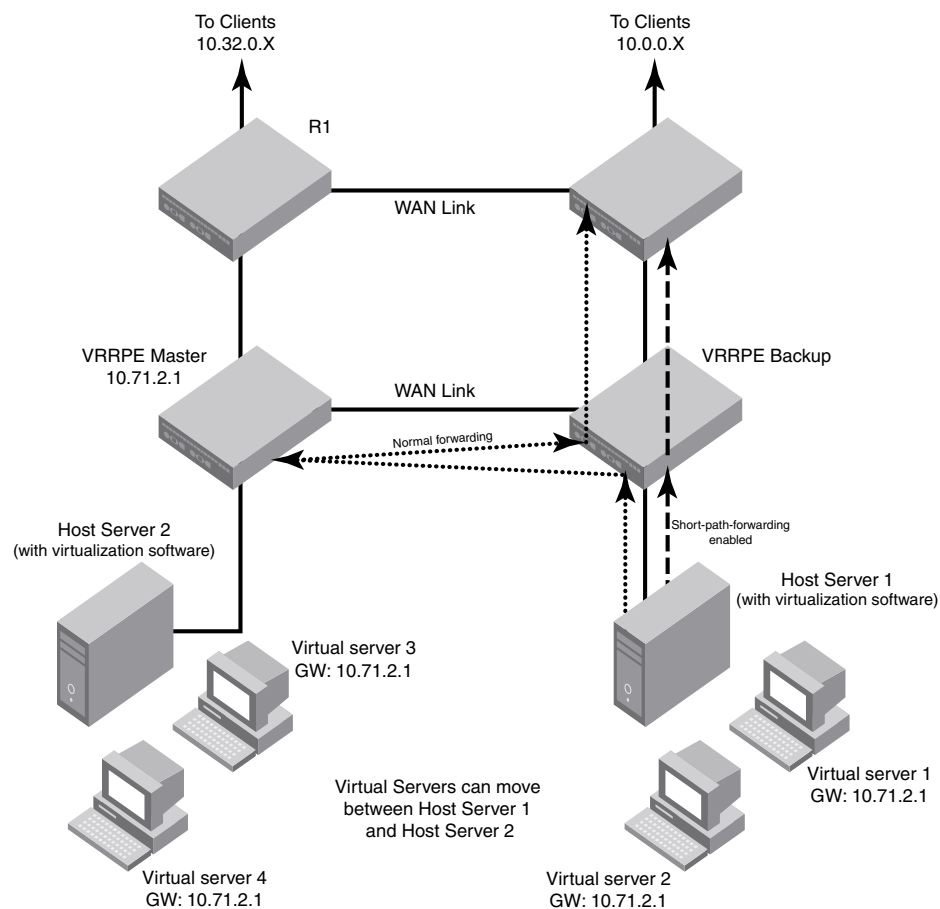
[Figure 187](#) shows an example of VRRP-E Extension for Server Virtualization. As shown, the virtual servers are dynamically moved between Host Server 1 and Host Server 2. Each time the virtual server is activated, it can be on a different Host Server, and sometimes the traffic crosses the WAN two times before it reaches the client. For example, in the VRRP-E implementation (without VRRP-E Extension for Server Virtualization), traffic from Virtual server 1 to the client at 10.0.0.X was switched to the VRRP-E Master router, then routed back to the VRRP-E Backup router, and then routed to the client (the normal forwarding path).

Short-path forwarding limitation

- Short-path forwarding (SPF) applies to IPv4 on Brocade FESX, FSX 800, and FSX 1600 platforms only. SPF is not supported on Brocade FWS, FCX, or ICX platforms.

Short-path forwarding configuration notes

- The VRRP-E Master router and Backup router must have routes to all destinations. You should utilize dynamic routing protocols such as Open Shortest Path First (OSPF) on all routers; otherwise, you must configure the static routes.
- Although it is not required, it is recommended that interfaces on different routers with the same VRID have the same SPF configuration. This ensures that the SPF behavior is retained after a failover. Different VRIDs, however, can have different SPF configurations.

FIGURE 187 VRRP-E Extension for short-path forwarding

VRRP-E Extension for short-path forwarding example

Under the VRRP-E VRID configuration level, there is an option to enable short-path forwarding. To enable short-path forwarding, enter the following commands.

```
Brocade (config)# router vrrp-extended
Brocade (config)# interface ve 10
Brocade (config-vif-10)# ip-address 10.10.10.25/24
Brocade (config-vif-10)# ip vrrp-extended vrid 10
Brocade (config-vif-10-vrid-10)# backup priority 50
Brocade (config-vif-10-vrid-10)# ip-address 10.10.10.254
Brocade (config-vif-10-vrid-10)# short-path-forwarding
Brocade (config-vif-10-vrid-10)# activate
```

Syntax: [no] short-path-forwarding [revert-priority <value>]

The **revert-priority** <value> parameter uses the priority value as the threshold to determine whether the short-path forwarding (SPF) behavior is effective. Typically, when short-path forwarding is enabled, the Backup router enforces SPF. For each port that goes down, the current priority of the VRRP-E router is lowered by the number specified in the **track-port** command. When the current priority is lower than the threshold, the SPF behavior is temporarily suspended and reverts back to the pre-SPF VRRP-E forwarding behavior. The value range is from 1 through 255.

Displaying short-path forwarding combinations

When short-path forwarding (SPF) is configured, the output of the following show commands include the SPF information:

- **show run**
- **show ip vrrp-e brief**
- **show ip vrrp-e vrid <vrid>**

The following example displays information about VRID 1 when only short-path forwarding is configured.

```
Brocade# show ip vrrp-e vrid 1
VRID 1
  Interface ethernet v100
  state backup
  administrative-status enabled
  priority 110
  current priority 90
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3500 msec
  preempt-mode true
  virtual ip address 100.1.1.3
  virtual mac address 02e0.5289.7001
  advertise backup: disabled
  master router 100.1.1.1 expires in 00:00:02.6
  track-port 1/13(down)
  short-path-forwarding enabled
```

The following example displays information about VRID 1 when short-path forwarding and revert-priority are configured.

```
Brocade# show ip vrrp-e vrid 1
VRID 1
  Interface ethernet v100
  state backup
  administrative-status enabled
  priority 110
  current priority 90
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3500 msec
  preempt-mode true
  virtual ip address 100.1.1.3
  virtual mac address 02e0.5289.7001
  advertise backup: disabled
  master router 100.1.1.1 expires in 00:00:02.7
  track-port 1/13(down)
  short-path-forwarding enabled <revertible priority 80 not reverted >
```


Forcing a Master router to abdicate to a Backup router

NOTE

Forcing a Master router to abdicate to a Backup router is not supported for IPv6 VRRP, IPv4 VRRP-E, and IPv6 VRRP-E. It is only supported for IPv4 VRRP.

You can force a VRRP Master to abdicate (give away control) of a VRID to a Backup router by temporarily changing the Master priority to a value less than that of the Backup router.

The VRRP Owner always has priority 255. You can use this feature to temporarily change the Owner priority to a value from 1 through 254.

NOTE

When you change the VRRP Owner priority, the change takes effect only for the current power cycle. The change is not saved to the startup-config file when you save the configuration and is not retained across a reload or reboot. Following a reload or reboot, the VRRP Owner again has priority 255.

To change the Master priority, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/6
Brocade(config-if-1/6)# ip vrrp vrid 1
Brocade(config-if-1/6-vrid-1)# owner priority 99
```

Syntax: [no] owner priority <num>

The <num> variable specifies the new priority and can be a number from 1 through 254.

When the command is enabled, the software changes the priority of the Master to the specified priority. If the new priority is lower than at least one Backup priority for the same VRID, the Backup router takes over and becomes the new Master until the next software reload or system reset.

To verify the change, enter the following command from any level of the CLI.

```
Brocade#show ip vrrp
Total number of VRRP routers defined: 1
Interface ethernet v3
auth-type simple text password
VRID 3
  state backup
  administrative-status enabled
  mode non-owner(backup)
  priority 110
  current priority 110
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3500 msec
  preempt-mode true
  ip-address 172.21.3.1
  virtual mac address 0000.5e00.0103
  advertise backup: enabled
  next hello sent in 00:00:26.1
  master router 172.21.3.1 expires in 00:00:02.7
```

This example shows that even though this Layer 3 switch is the Owner of the VRID (“mode owner”), the Layer 3 switch priority for the VRID is 110 and the state is now “backup” instead of “active”. In addition, the administrative status is “enabled”.

To change the Master priority back to the default Owner priority 255, enter **no** followed by the command you entered to change the priority. For example, to change the priority of a VRRP Owner back to 255 from 110, enter the following command.

```
Brocade(config-if-1/6-vrid-1)#no owner priority 110
```

You cannot set the priority to 255 using the **owner priority** command.

Displaying VRRP and VRRP-E information

You can display the following information for VRRP or VRRP-E:

- Summary configuration and status information
- Detailed configuration and status information
- VRRP and VRRP-E statistics
- CPU utilization statistics

Displaying summary information

To display summary information for a Layer 3 switch for VRRP, enter the **show ip vrrp brief** command at any level of the CLI.

```
Brocade#show ip vrrp brief
```

```
Total number of VRRP routers defined: 1
Interface VRID CurPri P State Master addr Backup addr VIP
1/6          1    255 P Init  192.53.5.1  192.53.5.3 192.53.5.1
```

To display summary information for IPv6 VRRP, enter the **show ipv6 vrrp brief** command at any level of the CLI.

```
Brocade#show ipv6 vrrp brief
```

```
Total number of VRRP routers defined: 1
Interface      VRID CurPri P State Master addr
Backup addr
VIP
1/5            1    255 P Master Master addr: Local
Backup addr: fe80::212:f2ff:fea8:3900
VIP           : 2001::1
```

To display summary information for IPv6 VRRP-E v3 , enter the **show ipv6 vrrp-extended brief** command at any level of the CLI.

```
Brocade#show ipv6 vrrp-extended brief
Total number of VRRP-Extended routers defined: 3
Interface      VRID CurPri P State Master addr
                Backup addr
                VIP
1/1/1          1    100   P Master Master addr: Local
                Backup addr: fe80::212:f2ff:fea8:5b00
                VIP      : 2000:1::100
1/1/2          2    150   P Master Master addr: Local
                Backup addr: fe80::212:f2ff:fea8:5b00
                VIP      : 2000:2::100
v51            100  100   P Master Master addr: Local
                Backup addr: fe80::212:f2ff:fea8:5b00
                VIP      : 2000:51::100
```

Syntax for IPv4 VRRP v2 and IPv6 VRRP v3:

Syntax: **show ip vrrp brief** | **ethernet** [**<slotnum>/<portnum>**] | **ve** **<num>** | **stat** | **vrid** **<num>**

Syntax: **show ipv6 vrrp brief** | **ethernet** [**<slotnum>/<portnum>**] | **ve** **<num>** | **stat** | **vrid** **<num>**

Syntax for IPv4 VRRP-E v2 and IPv6 VRRP-E v3:

Syntax: **show ip vrrp-extended brief** | **ethernet** [**<slotnum>/<portnum>**] | **ve** **<num>** | **stat** | **vrid** **<num>**

Syntax: **show ipv6 vrrp-extended brief** | **ethernet** [**<slotnum>/<portnum>**] | **ve** **<num>** | **stat** | **vrid** **<num>**

The **brief** option displays the summary information. If you do not use this option, detailed information is displayed instead. Refer to [“Displaying detailed information”](#) on page 1686.

The **ethernet** **<slotnum>** option is required on chassis devices if you specify a port number.

The **ethernet** **<portnum>** option specifies an Ethernet port. If you use this option, the command displays VRRP or VRRP-E information only for the specified port.

The **ve** **<num>** option specifies a virtual interface. If you use this option, the command displays VRRP or VRRP-E information only for the specified virtual interface.

The **stat** option displays statistics. Refer to [“Displaying statistics”](#) on page 1692.

The **vrid** **<num>** option specifies the virtual router ID. Enter a value from 1 through 255.

[Table 278](#) shows a description of the output for the **show ip vrrp brief** and **show ip vrrp-extended brief** commands.

TABLE 278 CLI display of VRRP or VRRP-E summary information

Field	Description
Total number of VRRP (or VRRP-Extended) routers defined	The total number of VRIDs configured on this Layer 3 switch. NOTE: The total applies only to the protocol the Layer 3 switch is running. For example, if the Layer 3 switch is running VRRP-E, the total applies only to VRRP-E routers.
Interface	The interface on which VRRP or VRRP-E is configured. If VRRP or VRRP-E is configured on multiple interfaces, information for each interface is listed separately.

TABLE 278 CLI display of VRRP or VRRP-E summary information (Continued)

Field	Description
VRID	The VRID configured on this interface. If multiple VRIDs are configured on the interface, information for each VRID is listed in a separate row.
CurPri	The current VRRP or VRRP-E priority of this Layer 3 switch for the VRID.
P	Whether the backup preempt mode is enabled. If the backup preempt mode is enabled, this field contains a "P". If the mode is disabled, this field is blank.
State	<p>This Layer 3 switch VRRP or VRRP-E state for the VRID. The state can be one of the following:</p> <ul style="list-style-type: none"> Init – The VRID is not enabled (activated). If the state remains Init after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. <p>NOTE: If the state is Init and the mode is incomplete, make sure you have specified the IP address for the VRID.</p> <ul style="list-style-type: none"> Backup – This Layer 3 switch is a Backup for the VRID. Master – This Layer 3 switch is the Master for the VRID.
Master addr	IP address of the router interface that is currently Master for the VRID.
Backup addr	IP addresses of router interfaces that are currently Backups for the VRID.
VIP	The virtual IP address that is being backed up by the VRID.

Displaying detailed information

To display detailed VRRP or VRRP-E information, enter the **show ip vrrp** command at any level of the CLI.

```
Brocade#show ip vrrp

Total number of VRRP routers defined: 1
Interface ethernet v3
auth-type simple text password
VRID 3
  state master
  administrative-status enabled
  mode owner
  priority 255
  current priority 255
  track-priority 150
  hello-interval 1000 msec
  ip-address 172.21.3.1
  virtual mac address 0000.5e00.0103
  advertise backup: disabled
  next hello sent in 00:00:00.7
  backup router 172.21.3.2 expires in 00:02:41.3
  track-port 3/14(up)
```

The following example is for a VRRP Backup.

```
Brocade#show ip vrrp

Total number of VRRP routers defined: 1
Interface ethernet v3
auth-type simple text password
VRID 3
  state backup
  administrative-status enabled
  mode non-owner(backup)
  priority 110
  current priority 110
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3500 msec
  preempt-mode true
  ip-address 172.21.3.1
  virtual mac address 0000.5e00.0103
  advertise backup: enabled
  next hello sent in 00:00:26.1
  master router 172.21.3.1 expires in 00:00:02.7
  track-port 4/1-4/4(up)
```

The following example is for a VRRP-E Master.

```
Brocade#show ip vrrp-extended

Total number of VRRP-Extended routers defined: 50
Interface ethernet v201
auth-type simple text password
VRID 201
  state master
  administrative-status enabled
  priority 220
  current priority 220
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3100 msec
  preempt-mode true
  virtual ip address 201.201.201.5
  virtual mac address 02e0.52d7.82c9
  advertise backup: enabled
  next hello sent in 00:00:00.1
  backup router 201.201.201.4 expires in 00:02:45.2
  backup router 201.201.201.3 expires in 00:02:47.6
  track-port 1/1/25*2/1/24(up)
```

Syntax: `show ip vrrp brief | ethernet [<slotnum>]/<portnum> | ve <num> | stat`

Syntax: `show ip vrrp-extended brief | ethernet [<slotnum>]/<portnum> | ve <num> | stat`

The **brief** option displays summary information. Refer to [“Displaying summary information”](#) on page 1684.

The **ethernet <portnum>** option specifies an Ethernet port. If you use this option, the command displays VRRP or VRRP-E information only for the specified port. Also, you must specify the **<slotnum>** variable on chassis devices.

The **ve** <num> option specifies a virtual interface. If you use this option, the command displays VRRP or VRRP-E information only for the specified virtual interface.

The **stat** option displays statistics. Refer to “[Displaying statistics](#)” on page 1692.

[Table 279](#) shows a description of the output for the **show ip vrrp** and **show ip vrrp-extended** commands.

TABLE 279 CLI display of VRRP or VRRP-E detailed information

Field	Description
Total number of VRRP (or VRRP-Extended) routers defined	The total number of VRIDs configured on this Layer 3 switch. NOTE: The total applies only to the protocol the Layer 3 switch is running. For example, if the Layer 3 switch is running VRRP-E, the total applies only to VRRP-E routers.
Interface parameters	
Interface	The interface on which VRRP, VRRP v3, VRRP-E, or IPv6 VRRP-E is configured. If VRRP, VRRP v3, VRRP-E, or IPv6 VRRP-E is configured on multiple interfaces, information for each interface is listed separately.
auth-type	The authentication type enabled on the interface.
VRID parameters	
VRID	The VRID configured on this interface. If multiple VRIDs are configured on the interface, information for each VRID is listed separately.
state	This Layer 3 switch VRRP, VRRP v3, VRRP-E, or IPv6 VRRP-E state for the VRID. The state can be one of the following: <ul style="list-style-type: none"> initialize – The VRID is not enabled (activated). If the state remains “initialize” after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. NOTE: If the state is “initialize” and the mode is incomplete, make sure you have specified the IP address for the VRID. <ul style="list-style-type: none"> backup – This Layer 3 switch is a Backup for the VRID. master – This Layer 3 switch is the Master for the VRID.
administrative-status	The administrative status of the VRID. The administrative status can be one of the following: <ul style="list-style-type: none"> disabled – The VRID is configured on the interface but VRRP or VRRP-E has not been activated on the interface. enabled – VRRP, VRRP v3, VRRP-E, or IPv6 VRRP-E has been activated on the interface.
mode	Indicates whether the Layer 3 switch is the Owner or a Backup for the VRID. NOTE: If “incomplete” appears after the mode, configuration for this VRID is incomplete. For example, you might not have configured the virtual IP address that is being backed up by the VRID. NOTE: This field applies only to VRRP or VRRP v3. All Layer 3 switches configured for VRRP-E are Backups.
priority	The device preferability for becoming the Master for the VRID. During negotiation, the router with the highest priority becomes the Master. If two or more devices are tied with the highest priority, the Backup interface with the highest IP address becomes the active router for the VRID.

TABLE 279 CLI display of VRRP or VRRP-E detailed information (Continued)

Field	Description
current priority	<p>The current VRRP, VRRP v3, VRRP-E, or IPv6 VRRP-E priority of this Layer 3 switch for the VRID. The current priority can differ from the configured priority (refer to the priority field) for the following reason:</p> <p>The current priority can differ from the configured priority in the VRID if the VRID is configured with track ports and the link on a tracked interface has gone down. Refer to “Track ports and track priority” on page 1654.</p>
hello-interval	<p>The configured value for the Hello interval. This is the amount of time, in milliseconds, between Hello messages from the Master to the Backups for a given VRID.</p> <p>NOTE: In some VRRP command outputs, Hello interval timers are displayed in seconds instead of milliseconds.</p>
dead interval	<p>The configured value for the dead interval. This is the amount of time, in milliseconds, that a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active.</p> <p>If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID.</p> <p>NOTE: If the value is 0, then you have not configured this parameter.</p> <p>NOTE: This field does not apply to VRRP Owners.</p> <p>NOTE: All timer fields (Hello interval, dead interval, current dead interval, and so on) are displayed in milliseconds.</p>
current dead interval	<p>The current value of the dead interval. This value is equal to the value configured for the dead interval.</p> <p>If the value for the dead interval is not configured, then the current dead interval is equal to three times the Hello interval plus Skew time (where Skew time is equal to 256 minus priority divided by 256).</p> <p>NOTE: This field does not apply to VRRP Owners.</p>
preempt mode	<p>Whether the backup preempt mode is enabled.</p> <p>NOTE: This field does not apply to VRRP Owners.</p>
virtual ip address	The virtual IP addresses that this VRID is backing up. The address can be an IPv4 or IPv6 address.
virtual mac address	The virtual MAC addresses for the VRID. The MAC address can be an IPv4 or IPv6 address.
advertise backup	<p>The IP addresses of Backups that have advertised themselves to this Layer 3 switch by sending Hello messages.</p> <p>NOTE: Hello messages from Backups are disabled by default. You must enable the Hello messages on the Backup for the Backup to advertise itself to the current Master. Refer to “Hello messages” on page 1653.</p>
backup router <ip-addr> expires in <time>	<p>The IP addresses of Backups that have advertised themselves to this Master by sending Hello messages.</p> <p>The <time> value indicates how long before the Backup expires. A Backup expires if you disable the advertise backup option on the Backup or the Backup becomes unavailable. Otherwise, the Backup next Hello message arrives before the Backup expires. The Hello message resets the expiration timer.</p> <p>An expired Backup does not necessarily affect the Master. However, if you have not disabled the advertise backup option on the Backup, then the expiration may indicate a problem with the Backup.</p> <p>NOTE: This field applies only when Hello messages are enabled on the Backups (using the advertise backup option).</p>

TABLE 279 CLI display of VRRP or VRRP-E detailed information (Continued)

Field	Description
next hello sent in <time>	How long until the Backup sends its next Hello message. NOTE: This field applies only when this Layer 3 switch is the Master and the Backup is configured to send Hello messages (the advertise backup option is enabled).
master router <ip-addr> expires in <time>	The IP address of the Master and the amount of time until the Master dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this Layer 3 switch itself will become the Master. NOTE: This field applies only when this Layer 3 switch is a Backup.
track port	The interfaces that the VRID interface is tracking. If the link for a tracked interface goes down, the VRRP, VRRP v3, VRRP-E, or IPv6 VRRP-E priority of the VRID interface is changed, causing the devices to renegotiate for Master. NOTE: This field is displayed only if track interfaces are configured for this VRID.

Displaying detailed information for an individual VRID

You can display information about the settings configured for a specified VRRP Virtual Router ID (VRID). To display information about VRID 1, enter the **show ip vrrp vrid** command.

```
Brocade#show ip vrrp vrid 2
VRID 2
  Interface ethernet v2
  state master
  administrative-status enabled
  version v2
  mode non-owner(backup)
  priority 100
  current priority 100
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3600 msec
  preempt-mode true
  ip-address 1.1.1.5
  virtual mac address 0000.5e00.0102
  advertise backup: disabled
  next hello sent in 00:00:01.0
```

To display information about the settings configured for a specified IPv6 VRRP VRID, enter the **show ipv6 vrrp vrid** command.


```

Brocade#show ipv6 vrrp vrid 1
VRID 1
  Interface ethernet 5
  state backup
  administrative-status enabled
  version v3
  mode non-owner(backup)
  priority 100
  current priority 100
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3000 msec
  preempt-mode true
  ip-address a7a7:a7a7:a7a7::1
  virtual mac address 0000.5e00.0201
  advertise backup: enabled

```

Syntax: `show ip vrrp vrid <num> [ethernet <num> | ve <num>]`

Syntax: `show ipv6 vrrp vrid <num> [ethernet <num> | ve <num>]`

The `<num>` variable specifies the VRID.

The `ethernet <num> | ve <num>` options specify an interface on which the VRID is configured. If you specify an interface, VRID information is displayed for that interface only. Otherwise, information is displayed for all the interfaces on which the specified VRID is configured.

[Table 280](#) shows a description of the output for the `show ip vrrp vrid` command.

TABLE 280 Output from the `show ip vrrp vrid` command

Field	Description
VRID	The specified VRID.
Interface	The interface on which VRRP is configured.
State	<p>This Layer 3 switch VRRP state for the VRID. The state can be one of the following:</p> <ul style="list-style-type: none"> Init – The VRID is not enabled (activated). If the state remains Init after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. <p>NOTE: If the state is Init and the mode is incomplete, make sure you have specified the IP address for the VRID.</p> <ul style="list-style-type: none"> Backup – This Layer 3 switch is a Backup for the VRID. Master – This Layer 3 switch is the Master for the VRID.
priority	The configured VRRP priority of this Layer 3 switch for the VRID.
current priority	The current VRRP priority of this Layer 3 switch for the VRID.
track priority	The new VRRP priority that the router receives for this VRID if the interface goes down.
hello interval	How often the Master router sends Hello messages to the Backups.
dead interval	The amount of time a Backup waits for a Hello message from the Master before determining that the Master is dead.

TABLE 280 Output from the show ip vrrp vrid command (Continued)

Field	Description
current dead interval	The current value of the dead interval. This value is equal to the value configured for the dead interval. If the value for the dead interval is not configured, then the current dead interval is equal to three times the Hello interval plus Skew time (where Skew time is equal to 256 minus priority divided by 256). NOTE: This field does not apply to VRRP Owners.
preempt mode	Whether the backup preempt mode is enabled. If the backup preempt mode is enabled, this field contains "true". If the mode is disabled, this field contains "false".
advertise backup	Whether Backup routers send Hello messages to the Master.

Displaying statistics

To display statistics on most Brocade devices, enter the **show ip vrrp stat** command at any level of the CLI.

```
Brocade#show ip vrrp stat
```

```
Interface ethernet 1/5
  rxd vrrp header error count = 0
  rxd vrrp auth error count = 0
  rxd vrrp auth passwd mismatch error count = 0
  rxd vrrp vrid not found error count = 0
VRID 1
  rxd arp packet drop count = 0
  rxd ip packet drop count = 0
  rxd vrrp port mismatch count = 0
  rxd vrrp ip address mismatch count = 0
  rxd vrrp hello interval mismatch count = 0
  rxd vrrp priority zero from master count = 0
  rxd vrrp higher priority count = 0
  transitioned to master state count = 1
  transitioned to backup state count = 1
```

To display IPv6 VRRP-E v3 statistics on a device, enter the following command at any level of the CLI.

```
Brocade#show ipv6 vrrp-extended stat ve 51
Interface ethernet v51
  rxd vrrp header error count = 0
  rxd vrrp auth error count = 0
  rxd vrrp auth passwd mismatch error count = 0
  rxd vrrp vrid not found error count = 0
VRID 100
  rxd arp packet drop count = 0
  rxd ip packet drop count = 0
  rxd vrrp port mismatch count = 0
  rxd vrrp ip address mismatch count = 0
  rxd vrrp hello interval mismatch count = 0
  rxd vrrp priority zero from master count = 0
  rxd vrrp higher priority count = 0
```

Table 281 shows a description of the output for the **show ip vrrp stat** and **show ip vrrp-extended stat** commands.

TABLE 281 CLI display of VRRP or VRRP-E statistics

Field	Description
Interface statistics	
Interface	The interface on which VRRP or VRRP-E is configured. If VRRP or VRRP-E is configured on more than one interface, the display lists the statistics separately for each interface.
rxed vrrp header error count	The number of VRRP or VRRP-E packets received by the interface that had a header error.
rxed vrrp auth error count	The number of VRRP or VRRP-E packets received by the interface that had an authentication error.
rxed vrrp auth passwd mismatch error count	The number of VRRP or VRRP-E packets received by the interface that had a password value that does not match the password used by the interface for authentication.
rxed vrrp vrid not found error count	The number of VRRP or VRRP-E packets received by the interface that contained a VRID that is not configured on this interface.
VRID statistics	
rxed arp packet drop count	The number of ARP packets addressed to the VRID that were dropped.
rxed ip packet drop count	The number of IP packets addressed to the VRID that were dropped.
rxed vrrp port mismatch count	The number of packets received that did not match the configuration for the receiving interface.
rxed vrrp ip address mismatch count	The number of packets received that did not match the configured IP addresses.
rxed vrrp hello interval mismatch count	The number of packets received that did not match the configured Hello interval.
rxed vrrp priority zero from master count	Indicates that the current Master has resigned.
rxed vrrp higher priority count	The number of VRRP or VRRP-E packets received by the interface that had a higher backup priority for the VRID than this Layer 3 switch backup priority for the VRID.
transitioned to master state count	The number of times this Layer 3 switch has changed from the backup state to the master state for the VRID.
transitioned to backup state count	The number of times this Layer 3 switch has changed from the master state to the backup state for the VRID.

Clearing VRRP or VRRP-E statistics

To clear VRRP or VRRP-E statistics, enter the **clear ip vrrp-stat** command at the Privileged EXEC level or any configuration level of the CLI.

```
Brocade#clear ip vrrp-stat
```

Syntax: clear ip vrrp-stat

To clear IPv6 VRRP v3 or IPv6 VRRP-E v3 statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI.

```
Brocade#clear ipv6 vrrp-stat
```

Syntax: clear ipv6 vrrp-stat

Displaying CPU utilization statistics

To display CPU utilization statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI. In the following output example, IPv6 protocols are also displayed.

```
Brocade#show process cpu
```

Process Name	5Sec(%)	1Min(%)	5Min(%)	15Min(%)	Runtime(ms)
ARP	0.02	0.02	0.02	0.02	15532
BGP	0.00	0.00	0.00	0.00	0
DOT1X	0.00	0.00	0.00	0.00	0
GVRP	0.00	0.00	0.00	0.00	0
ICMP	0.01	0.01	0.01	0.01	8608
IP	0.09	0.12	0.11	0.08	72959
OSPF	0.05	0.07	0.07	0.06	85015
RIP	0.00	0.00	0.00	0.00	98
STP	0.68	0.86	0.78	0.57	568586
VRRP	0.42	0.54	0.50	0.37	357133

Process Name	5Sec(%)	1Min(%)	5Min(%)	15Min(%)	Runtime(ms)
IPv6	1.23	1.49	0.99	1.40	917973
ICMP6	0.04	0.05	0.06	0.04	38508
ND6	0.00	0.01	0.01	0.01	6691
RIPng	0.00	0.00	0.00	0.00	45
OSPFv3	0.00	0.00	0.00	0.00	1515
IPV6_RX	0.16	0.21	0.21	0.14	143506

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running, as shown in the following example.

```
Brocade#show process cpu
```

The system has only been up for 6 seconds.

Process Name	5Sec(%)	1Min(%)	5Min(%)	15Min(%)	Runtime(ms)
ARP	0.01	0.00	0.00	0.00	0
BGP	0.00	0.00	0.00	0.00	0
GVRP	0.00	0.00	0.00	0.00	0
ICMP	0.01	0.00	0.00	0.00	1
IP	0.00	0.00	0.00	0.00	0
OSPF	0.00	0.00	0.00	0.00	0
RIP	0.00	0.00	0.00	0.00	0
STP	0.00	0.00	0.00	0.00	0
VRRP	0.00	0.00	0.00	0.00	0

To display utilization statistics for a specific number of seconds, enter a command such as the following. In the following output example, IPv6 protocols are also displayed for a specific number of seconds.

```
Brocade#show process cpu 2
```

Statistics for last 1 sec and 999 ms

Process Name	Sec(%)	Time(ms)
ARP	0.01	0
BGP	0.00	0
DOT1X	0.00	0
GVRP	0.00	0
ICMP	0.01	0
IP	0.04	0
OSPF	0.07	1
RIP	0.00	0
STP	0.97	19
VRRP	0.53	10

Statistics for last 1 sec and 999 ms

Process Name	Sec(%)	Time(ms)
IPv6	0.09	1
ICMP6	0.05	1
ND6	0.00	0
RIPng	0.00	0
OSPFv3	0.00	0
IPV6_RX	0.04	0

When you specify how many seconds of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is for the previous 1 second plus 80 milliseconds.

Syntax: `show process cpu [<num>]`

The `<num>` variable specifies the number of seconds and can be a value from 1 through 900. If you use this variable, the command lists the usage statistics only for the specified number of seconds. If you do not use this variable, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying VRRP and VRRP-E information for IPv6

You can display information for IPv6 VRRP or VRRP-E v3.

Displaying detailed information for IPv6 VRRP v3 and IPv6 VRRP-E v3

To display information for an IPv6 VRRP Owner, enter the **show ipv6 vrrp** command at any level of the CLI.

```
Brocade#show ipv6 vrrp
Total number of VRRP routers defined: 25
Interface ethernet v52
auth-type no authentication
VRID 52
  state master
  administrative-status enabled
  version v3
  mode owner
  priority 255
  current priority 255
  track-priority 5
  hello-interval 1000 msec
  ipv6-address 2172:52::52:3
  virtual mac address 0000.5e00.0234
  advertise backup: disabled
  next hello sent in 00:00:00.1
  backup router fe80::224:38ff:fec8:5a40 expires in 00:02:03.1
```

To display information for an IPv6 VRRP Backup, enter the **show ipv6 vrrp** command at any level of the CLI.

```
Brocade#show ipv6 vrrp
Total number of VRRP routers defined: 26
Interface ethernet v52
auth-type no authentication
VRID 52
  state backup
  administrative-status enabled
  version v3
  mode non-owner(backup)
  priority 101
  current priority 20
  track-priority 20
  hello-interval 100 msec
  dead-interval 0 msec
  current dead-interval 300 msec
  preempt-mode true
  ipv6-address 2172:52::52:3
  virtual mac address 0000.5e00.0234
  advertise backup: enabled
  next hello sent in 00:00:36.5
  master router fe80::768e:f8ff:fe33:8600 expires in 00:00:00.2
  track-port 2/1/3*4/1/4(down) v41(up)
```

Syntax: `show ipv6 vrrp brief | ethernet <stack-unit>/<slotnum>/<portnum> | stat [ethernet <stack-unit>/<slotnum>/<portnum> | ve <num>] | vrid <num>`

To display detailed information for IPv6 VRRP-E, enter the `show ipv6 vrrp-extended` command at any level of the CLI.

```
Brocade#show ipv6 vrrp-extended
Total number of VRRP-Extended routers defined: 1
Interface ethernet v201
auth-type md5 authentication
VRID 201
  state master
  administrative-status enabled
  priority 100
  current priority 100
  hello-interval 1000 msec
  dead-interval 0 msec
  current dead-interval 3600 msec
  preempt-mode true
  virtual ipv6 address 2201:201::201:5
  virtual mac address 02e0.5202.bac9
  advertise backup: enabled
  next hello sent in 00:00:01.0
```

Syntax: `show ipv6 vrrp-extended brief | ethernet <stack-unit>/<slotnum>/<portnum> | stat [ethernet <stack-unit>/<slotnum>/<portnum> | ve <num>] | vrid <num>`

For more information on the field descriptions for the `show ipv6 vrrp` command and the `show ipv6 vrrp-extended` command, refer to [“CLI display of VRRP or VRRP-E detailed information”](#) on page 1688.

Configuration examples

The following sections contain the CLI commands for implementing the VRRP and VRRP-E configurations shown in [Figure 185](#) on page 1651 and [Figure 186](#) on page 1657.

VRRP example

To implement the VRRP configuration shown in [Figure 185](#) on page 1651, use the following method.

Configuring Switch 1

To configure VRRP Switch 1, enter the following commands.

```
Brocade Switch1(config)#switch vrrp
Brocade Switch1(config)#interface ethernet 1/6
Brocade Switch1(config-if-1/6)#ip-address 192.53.5.1
Brocade Switch1(config-if-1/6)#ip vrrp vrid 1
Brocade Switch1(config-if-1/6-vrid-1)#owner track-priority 20
Brocade Switch1(config-if-1/6-vrid-1)#track-port ethernet 2/4
Brocade Switch1(config-if-1/6-vrid-1)#ip-address 192.53.5.1
Brocade Switch1(config-if-1/6-vrid-1)#activate
```

NOTE

When you configure the Master (Owner), the address you enter with the **ip-address** command must already be configured on the interface.

Configuring Switch 2

To configure Switch 2 in [Figure 185](#) on page 1651 after enabling VRRP, enter the following commands.

```
Brocade Switch2(config)#switch vrrp
Brocade Switch2(config)#interface ethernet 1/5
Brocade Switch2(config-if-1/5)#ip-address 192.53.5.3
Brocade Switch2(config-if-1/5)#ip vrrp vrid 1
Brocade Switch2(config-if-1/5-vrid-1)#backup priority 100 track-priority 19
Brocade Switch2(config-if-1/5-vrid-1)#track-port ethernet 3/2
Brocade Switch2(config-if-1/5-vrid-1)#ip-address 192.53.5.1
Brocade Switch2(config-if-1/5-vrid-1)#activate
```

The **backup** command specifies that this router is a VRRP Backup for virtual router VRID1. The IP address entered with the **ip-address** command is the same IP address as the one entered when configuring Switch 1. In this case, the IP address cannot also exist on Switch 2, but the interface on which you are configuring the VRID Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this VRID on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router VRRP priority in relation to the other VRRP routers in this virtual router. The **track-priority** parameter specifies the new VRRP priority that the router receives for this VRID if the interface goes down. Refer to [“Track ports and track priority”](#) on page 1654.

The **activate** command activates the VRID configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: router vrrp

Syntax: ip vrrp vrid <vrid>

Syntax: owner [track-priority <value>]

Syntax: backup [priority <value>] [track-priority <value>]

Syntax: track-port ethernet [<slotnum>/]<portnum> | ve <num>

Syntax: ip-address <ip-addr>

Syntax: activate

VRRP-E example

To implement the VRRP-E configuration shown in [Figure 186](#) on page 1657, use the following CLI method.

Configuring Switch 1

To configure VRRP Switch 1 in [Figure 186](#) on page 1657, enter the following commands.

```
Brocade Switch1(config)#switch vrrp-extended
Brocade Switch1(config)#interface ethernet 1/6
Brocade Switch1(config-if-1/6)#ip-address 192.53.5.2/24
Brocade Switch1(config-if-1/6)#ip vrrp-extended vrid 1
Brocade Switch1(config-if-1/6-vrid-1)#backup priority 110 track-priority 20
Brocade Switch1(config-if-1/6-vrid-1)#track-port ethernet 2/4
Brocade Switch1(config-if-1/6-vrid-1)#ip-address 192.53.5.254
Brocade Switch1(config-if-1/6-vrid-1)#activate
VRRP Switch 1 for this interface is activating
Brocade Switch1(config-if-1/6-vrid-1)#exit
Brocade Switch1(config)#interface ethernet 1/6
Brocade Switch1(config-if-1/6)#ip vrrp-extended vrid 2
Brocade Switch1(config-if-1/6-vrid-1)#backup priority 100 track-priority 20
Brocade Switch1(config-if-1/6-vrid-1)#track-port ethernet 2/4
Brocade Switch1(config-if-1/6-vrid-1)#ip-address 192.53.5.253
Brocade Switch1(config-if-1/6-vrid-1)#activate
VRRP Switch 1 for this interface is activating
```

NOTE

The address you enter with the **ip-address** command cannot be the same as a real IP address configured on the interface.

Configuring Switch 2

To configure Switch 2, enter the following commands.

```
Brocade Switch2(config)#switch vrrp-extended
Brocade Switch2(config)#interface ethernet 5/1
Brocade Switch2(config-if-5/1)#ip-address 192.53.5.3/24
Brocade Switch2(config-if-5/1)#ip vrrp-extended vrid 1
Brocade Switch2(config-if-5/1-vrid-1)#backup priority 100 track-priority 20
Brocade Switch2(config-if-5/1-vrid-1)#track-port ethernet 3/2
Brocade Switch2(config-if-5/1-vrid-1)#ip-address 192.53.5.254
Brocade Switch2(config-if-5/1-vrid-1)#activate
VRRP Switch 2 for this interface is activating
```

```

Brocade Switch2(config-if-5/1-vrid-1)#exit
Brocade Switch2(config)#interface ethernet 5/1
Brocade Switch2(config-if-5/1)#ip vrrp-extended vrid 2
Brocade Switch2(config-if-5/1-vrid-1)#backup priority 110 track-priority 20
Brocade Switch2(config-if-5/1-vrid-1)#track-port ethernet 2/4
Brocade Switch2(config-if-5/1-vrid-1)#ip-address 192.53.5.253
Brocade Switch2(config-if-5/1-vrid-1)#activate
VRRP Switch 2 for this interface is activating

```

The **backup** command specifies that this router is a VRRP-E Backup for virtual router VRID1. The IP address entered with the **ip-address** command is the same IP address as the one entered when configuring Switch 1. In this case, the IP address cannot also exist on Switch 2, but the interface on which you are configuring the VRID Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this VRID on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router VRRP-E priority in relation to the other VRRP-E routers in this virtual router. The **track-priority** parameter specifies the new VRRP-E priority that the router receives for this VRID if the interface goes down. Refer to [“Track ports and track priority”](#) on page 1654.

The **activate** command activates the VRID configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP-E configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: router vrrp-extended

Syntax: ip vrrp-extended vrid <vrid>

Syntax: backup [priority <value>] [track-priority <value>]

Syntax: track-port ethernet [<slotnum>/]<portnum> | ve <num>

Syntax: ip-address <ip-addr>

Syntax: activate

Rule-Based IP ACLs

Table 282 and Table 283 list the individual Brocade FastIron switches and Access Control List (ACL) features they support. Table 282 lists the features supported on inbound traffic, while Table 283 lists the features supported on outbound traffic. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 282 Supported ACL features on inbound traffic

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Hardware-based ACLs	Yes	Yes	Yes	Yes	Yes
Standard named and numbered ACLs	Yes	Yes	Yes	Yes	Yes
Extended named and numbered ACLs	Yes	Yes	Yes	Yes	Yes
User input preservation for ACL TCP/UDP port numbers	Yes	Yes	Yes	Yes	Yes
ACL comment text	Yes	Yes	Yes	Yes	Yes
ACL logging of denied packets	Yes	Yes	Yes	Yes	Yes
ACL logging with traffic rate limiting (to prevent CPU overload)	Yes To enable, configure a traffic conditioner.	Yes	Yes	Yes	Yes
Strict control of ACL filtering of fragmented packets	Yes	Yes	Yes	Yes	Yes
ACL support for switched traffic in the router image	Yes To enable, use the bridged-routed parameter..	Yes	Yes	Yes	ICX 6450 only This feature is enabled by default on FWS, FCX and ICX devices. There is no CLI command to enable or disable it.
ACL filtering based on VLAN membership or VE port membership	Yes	Yes	Yes	Yes	Yes
ACLs to filter ARP packets	No	Yes	No	No	No
Filtering on IP precedence and ToS value	Yes	Yes	Yes	Yes	Yes
Combined DSCP and internal marking in one ACL rule	Yes	No	No	No	No
QoS options for IP ACLs	Yes	Yes	Yes	Yes	Yes ¹
DSCP CoS mapping	Yes	No	No	No	No
Priority mapping using ACLs	Yes	Yes	Yes	Yes	Yes

TABLE 282 Supported ACL features on inbound traffic

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Hardware usage statistics	Yes	Yes	Yes	Yes	Yes
Policy-based routing (PBR) (Supported in the full Layer 3 code only)	Yes	No	Yes	Yes	No

1. ICX 6430 devices have only four priority queues. See [“Queues for the ICX 6430 switch”](#) on page 1973 for more information.

TABLE 283 Supported ACL features on outbound traffic

Feature	FSX 800 FSX 1600 ¹	FESX	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Hardware-based ACLs	Yes	No	No	Yes	Yes	Yes
Standard named and numbered ACLs	Yes	No	No	Yes	Yes	Yes
Extended named and numbered ACLs	Yes	No	No	Yes	Yes	Yes
User input preservation for ACL TCP/UDP port numbers	Yes	No	No	Yes	Yes	Yes
ACL comment text	Yes	No	No	Yes	Yes	Yes
ACL logging of denied packets	No	No	No	No	No	No
ACL logging with traffic rate limiting (to prevent CPU overload)	No	No	No	No	No	No
Strict control of ACL filtering of fragmented packets	Yes	No	No	Yes	Yes	Yes
ACL support for switched traffic in the router image	Yes	No	No	Yes	Yes	ICX 6450 only
This feature is enabled by default for outbound ACLs on platforms that support outbound ACL support. There is no CLI command to enable or disable it.						
ACL filtering based on VLAN membership or VE port membership	Not applicable for outbound traffic.					
ACLs to filter ARP packets	Not applicable for outbound traffic.					
Filtering on IP precedence and ToS value	Yes	No	No	Yes	Yes	Yes
Combined DSCP and internal marking in one ACL rule	Not applicable for outbound traffic, as DSCP CoS mapping is not supported.					
QoS options for IP ACLs ²	Yes	No	No	Yes	Yes	Yes
DSCP CoS mapping	DSCP CoS mapping is not supported for outgoing traffic.					
Priority mapping using ACLs	Internal priority marking is not supported for outgoing traffic.					
Hardware usage statistics	Yes	No	No	Yes	Yes	Yes
Policy-based routing (PBR) (Supported in the full Layer 3 code only)	Not applicable for outbound traffic					No

1. ACL features for outbound traffic are only supported on specific FastIron SX 800 & FastIron SX 1600 modules. Please check with your Brocade Support representative for details.
2. DSCP CoS mapping is not supported for outgoing traffic.

This chapter describes how Access Control Lists (ACLs) are implemented and configured in the Brocade devices.

NOTE

For information about IPv6 ACLs, refer to [Chapter 41, “IPv6 ACLs”](#).

ACL overview

Brocade devices support **rule-based ACLs** (sometimes called hardware-based ACLs), where the decisions to permit or deny packets are processed in hardware and all permitted packets are switched or routed in hardware. All denied packets are also dropped in hardware. In addition, FastIron FWS devices support inbound ACLs only. Outbound ACLs are not supported on those devices. FSX, FCX, and ICX devices support both inbound and outbound ACLs. The ACL features supported on inbound and outbound traffic are as listed in [Table 282](#) and [Table 283](#) respectively and discussed in more detail in the rest of this chapter.

NOTE

FastIron devices do not support flow-based ACLs.

Rule-based ACLs program the ACL entries you assign to an interface into Content Addressable Memory (CAM) space allocated for the ports. The ACLs are programmed into hardware at startup (or as new ACLs are entered and bound to ports). Devices that use rule-based ACLs program the ACLs into the CAM entries and use these entries to permit or deny packets in the hardware, without sending the packets to the CPU for processing.

Rule-based ACLs are supported on the following interface types:

- Gbps Ethernet ports
- 10 Gbps Ethernet ports
- Trunk groups
- Virtual routing interfaces

Types of IP ACLs

You can configure the following types of IP ACLs:

- **Standard** – Permits or denies packets based on source IP address. Valid standard ACL IDs are 1 – 99 or a character string.
- **Extended** – Permits or denies packets based on source and destination IP address and also based on IP protocol information. Valid extended ACL IDs are a number from 100 – 199 or a character string.

ACL IDs and entries

ACLs consist of ACL IDs and ACL entries:

- **ACL ID** – An ACL ID is a number from 1 – 99 (for a standard ACL) or 100 – 199 (for an extended ACL) or a character string. The ACL ID identifies a collection of individual ACL entries. When you apply ACL entries to an interface, you do so by applying the ACL ID that contains the ACL entries to the interface, instead of applying the individual entries to the interface. This makes applying large groups of access filters (ACL entries) to interfaces simple. Refer to [“Numbered and named ACLs”](#) on page 1706.

NOTE

This is different from IP access policies. If you use IP access policies, you apply the individual policies to interfaces.

- **ACL entry** – Also called an **ACL rule**, this is a filter command associated with an ACL ID. The maximum number of ACL rules you can configure is a system-wide parameter and depends on the device you are configuring. You can configure up to the maximum number of entries in any combination in different ACLs. For example, on a FESX switch, you can configure 4095 entries in one ACL, 2046 entries in two ACLs, etc.. The total number of entries in all ACLs cannot exceed the system maximum listed in [Table 284](#).

TABLE 284 Maximum number of ACL entries

System	Maximum ACL rules per port region	Maximum ACL entries per system
FESX Layer 2 Switch	IPv4 devices = 1016	4096
FESX Layer 3 Switch	IPv6 devices = 1023	
FSX 800 and FSX 1600 Layer 2 Switch	1015	8192
FSX 800 and FSX 1600 Layer 3 Switch		
FWS base Layer 3 Switch	756 (24-port) 1512 (48-port)	756 (24-port) 4096 (48-port)
FCX Layer 2 or Layer 3 Switch	4093	8192
ICX 6610	3069	8192
ICX 6430	508	8192
ICX 6450	3068	8192

You configure ACLs on a global basis, then apply them to the incoming or outgoing traffic on specific ports. The software applies the entries within an ACL in the order they appear in the ACL configuration. As soon as a match is found, the software takes the action specified in the ACL entry (permit or deny the packet) and stops further comparison for that packet.

Numbered and named ACLs

When you configure an ACL, you can refer to the ACL by a numeric ID or by an alphanumeric name. The commands to configure numbered ACLs are different from the commands for named ACLs.

- **Numbered ACL** – If you refer to the ACL by a numeric ID, you can use 1 – 99 for a standard ACL or 100 – 199 for an extended ACL.
- **Named ACL** – If you refer to the ACL by a name, you specify whether the ACL is a standard ACL or an extended ACL, then specify the name.

You can configure up to 99 standard numbered IP ACLs and 100 extended numbered IP ACLs. You also can configure up to 99 standard named ACLs and 100 extended named ACLs by number.

Default ACL action

The default action when no ACLs are configured on a device is to permit all traffic. However, once you configure an ACL and apply it to a port, the default action for that port is to deny all traffic that is not explicitly permitted on the port:

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.
- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The software permits packets that are not denied by the deny entries.

How hardware-based ACLs work

When you bind an ACL to inbound or outbound traffic on an interface, the device programs the Layer 4 CAM with the ACL. Permit and deny rules are programmed. Most ACL rules require one Layer 4 CAM entry. However, ACL rules that match on more than one TCP or UDP application port may require several CAM entries. The Layer 4 CAM entries for ACLs do not age out. They remain in the CAM until you remove the ACL:

- If a packet received on the interface matches an ACL rule in the Layer 4 CAM, the device permits or denies the packet according to the ACL.
- If a packet does not match an ACL rule, the packet is dropped, since the default action on an interface that has ACLs is to deny the packet.

How fragmented packets are processed

The descriptions above apply to non-fragmented packets. The default processing of fragments by hardware-based ACLs is as follows:

- The first fragment of a packet is permitted or denied using the ACLs. The first fragment is handled the same way as non-fragmented packets, since the first fragment contains the Layer 4 source and destination application port numbers. The device uses the Layer 4 CAM entry if one is programmed, or applies the interface's ACL entries to the packet and permits or denies the packet according to the first matching ACL.
- For other fragments of the same packet, they are subject to a rule only if there is no Layer 4 information in the rule or in any preceding rules.

The fragments are forwarded even if the first fragment, which contains the Layer 4 information, was denied. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.

On 48GC, 8x10GC, 2x10GC and 24GF modules, the first IP fragment packet is also treated as an IP packet fragment.

For tighter control, you can configure the port to drop all packet fragments. Refer to [“Enabling strict control of ACL filtering of fragmented packets”](#) on page 1732.

Hardware aging of Layer 4 CAM entries

Rule-based ACLs use Layer 4 CAM entries. The device permanently programs rule-based ACLs into the CAM. The entries never age out.

ACL configuration considerations

- See [“ACL overview”](#) on page 1705 for details on which devices support inbound and outbound ACLs.
- Hardware-based ACLs are supported on the following devices:
 - Gbps Ethernet ports
 - 10 Gbps Ethernet ports
 - Trunk groups
 - Virtual routing interfaces

NOTE

Brocade FCX devices do not support ACLs on Group VEs, even though the CLI contains commands for this action.

- Inbound ACLs apply to all traffic, including management traffic. By default outbound ACLs are not applied to traffic generated by the CPU. This must be enabled using the `enable egress-acl-on-control-traffic` command. See [“Applying egress ACLs to Control \(CPU\) traffic”](#) on page 1725 for details.
- The number of ACLs supported per device is listed in [Table 284](#).
- Hardware-based ACLs support only one ACL per port. The ACL of course can contain multiple entries (rules). For example, hardware-based ACLs do not support ACLs 101 and 102 on port 1, but hardware-based ACLs do support ACL 101 containing multiple entries.
- For devices that support both, inbound ACLs and outbound ACLs can co-exist. When an inbound ACL and an outbound ACL are configured on the same port, the outbound ACL is applied only on outgoing traffic.
- ACLs are affected by port regions. For example, on the FESX and FSX, multiple ACL groups share 1016 ACL rules per port region. Each ACL group must contain one entry for the implicit *deny all IP traffic* clause. Also, each ACL group uses a multiple of 8 ACL entries. For example, if all ACL groups contain 5 ACL entries, you could add 127 ACL groups (1016/8) in that port region. If all your ACL groups contain 8 ACL entries, you could add 63 ACL groups, since you must account for the implicit deny entry.
- By default, the first fragment of a fragmented packet received by the Brocade device is permitted or denied using the ACLs, but subsequent fragments of the same packet are forwarded in hardware. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.
- ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled. Also, IP source guard and ACLs are supported together on the same port, as long as both features are configured at the port-level or per-port-per-VLAN level. Brocade ports do not support IP source guard and ACLs on the same port if one is configured at the port-level and the other is configured at the per-port-per-VLAN level.
- Ingress MAC filters can be applied to the same port as an outbound ACL.

- A DOS attack configuration on a port will only apply on the ingress traffic.
- Outbound ACLs cannot be configured through a RADIUS server as dynamic or user-based ACLs. However, outbound ACLs can still be configured with MAC-AUTH/DOT1X enabled, as they the two are configured in different directions.
- The following ACL features and options are not supported on the FastIron devices:
 - Applying an ACL on a device that has Super Aggregated VLANs (SAVs) enabled.
 - ACL logging of permitted packets– ACL logging is supported for packets that are sent to the CPU for processing (denied packets) for inbound traffic. ACL logging is not supported for packets that are processed in hardware (permitted packets).
 - Flow-based ACLs
 - Layer 2 ACLs
- You can apply an ACL to a port that has TCP SYN protection or ICMP smurf protection, or both, enabled.

Configuring standard numbered ACLs

This section describes how to configure standard numbered ACLs with numeric IDs and provides configuration examples.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard numbered ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a device, refer to [“ACL IDs and entries”](#) on page 1706.

Standard numbered ACL syntax

Syntax: [no] access-list <ACL-num> deny | permit <source-ip> | <hostname> <wildcard> [log]

or

Syntax: [no] access-list <ACL-num> deny | permit <source-ip>/<mask-bits> | <hostname> [log]

Syntax: [no] access-list <ACL-num> deny | permit host <source-ip> | <hostname> [log]

Syntax: [no] access-list <ACL-num> deny | permit any [log]

Syntax: [no] ip access-group <ACL-num> in | out

The <ACL-num> parameter is the access list number from 1 – 99.

The **deny** | **permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

The <source-ip> parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE

To specify the host name instead of the IP address, the host name must be configured using the DNS resolver on the Brocade device. To configure the DNS resolver name, use the **ip dns server-address...** command at the global CONFIG level of the CLI.

The `<wildcard>` parameter specifies the mask value to compare against the host address specified by the `<source-ip>` parameter. The `<wildcard>` is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the `<source-ip>`. Ones mean any value matches. For example, the `<source-ip>` and `<wildcard>` values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of “209.157.22.26 0.0.0.255” as “209.157.22.26/24”. The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into ones. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in “/`<mask-bits>`” format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** `<source-ip> | <hostname>` parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

The **log** argument configures the device to generate Syslog entries and SNMP traps for inbound packets that are denied by the access policy.

The **in | out** parameter applies the ACL to incoming or outgoing traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port, or virtual interface.

NOTE

If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface.

Configuration example for standard numbered ACLs

To configure a standard ACL and apply it to incoming traffic on port 1/1, enter the following commands.

```
Brocade(config)#access-list 1 deny host 209.157.22.26 log
Brocade(config)#access-list 1 deny 209.157.29.12 log
Brocade(config)#access-list 1 deny host IPHost1 log
Brocade(config)#access-list 1 permit any
Brocade(config)#int eth 1/1
Brocade(config-if-1/1)#ip access-group 1 in
Brocade(config)#write memory
```

The commands in this example configure an ACL to deny packets from three source IP addresses from being received on port 1/1. The last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries.

Standard named ACL configuration

This section describes how to configure standard named ACLs with alphanumeric IDs. This section also provides configuration examples.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard named ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a device, refer to [“ACL IDs and entries”](#) on page 1706.

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL name with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

Standard named ACL syntax

Syntax: [no] ip access-list standard <ACL-name> | <ACL-num>

Syntax: deny | permit <source-ip> | <hostname> <wildcard> [log]

or

Syntax: deny | permit <source-ip>/<mask-bits> | <hostname> [log]

Syntax: deny | permit host <source-ip> | <hostname> [log]

Syntax: deny | permit any [log]

Syntax: [no] ip access-group <ACL-name> in | out

The <ACL-name> parameter is the access list name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, “ACL for Net1”).

The `<ACL-num>` parameter allows you to specify an ACL number if you prefer. If you specify a number, you can specify from 1 – 99 for standard ACLs.

NOTE

For convenience, the software allows you to configure numbered ACLs using the syntax for named ACLs. The software also still supports the older syntax for numbered ACLs. Although the software allows both methods for configuring numbered ACLs, numbered ACLs are always formatted in the startup-config and running-config files in using the older syntax, as follows.

```
access-list 1 deny host 209.157.22.26 log
access-list 1 deny 209.157.22.0 0.0.0.255 log
access-list 1 permit any
access-list 101 deny tcp any any eq http log
```

The **deny** | **permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

The `<source-ip>` parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE

To specify the host name instead of the IP address, the host name must be configured using the DNS resolver on the Brocade device. To configure the DNS resolver name, use the **ip dns server-address...** command at the global CONFIG level of the CLI.

The `<wildcard>` parameter specifies the mask value to compare against the host address specified by the `<source-ip>` parameter. The `<wildcard>` is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the `<source-ip>`. Ones mean any value matches. For example, the `<source-ip>` and `<wildcard>` values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of “209.157.22.26 0.0.0.255” as “209.157.22.26/24”. The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into ones. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in “/`<mask-bits>`” format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** <source-ip> | <hostname> parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

The **log** argument configures the device to generate Syslog entries and SNMP traps for inbound packets that are denied by the access policy.

NOTE

You can enable logging on inbound ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the **log** parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **in** | **out** parameter applies the ACL to incoming or outgoing traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or virtual interface.

NOTE

If the ACL is bound to a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. See [“Enabling ACL filtering based on VLAN membership or VE port membership”](#) on page 1734 for further details.

Configuration example for standard named ACLs

To configure a standard named ACL, enter commands such as the following.

```
Brocade(config)#ip access-list standard Net1
Brocade(config-std-nACL)#deny host 209.157.22.26 log
Brocade(config-std-nACL)#deny 209.157.29.12 log
Brocade(config-std-nACL)#deny host IPHost1 log
Brocade(config-std-nACL)#permit any
Brocade(config-std-nACL)#exit
Brocade(config)#int eth 1/1
Brocade(config-if-e1000-1/1)#ip access-group Net1 in
```

The commands in this example configure a standard ACL named “Net1”. The entries in this ACL deny packets from three source IP addresses from being forwarded on port 1. Since the implicit action for an ACL is “deny”, the last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries. For an example of how to configure the same entries in a numbered ACL, refer to [“Configuring standard numbered ACLs”](#) on page 1709.

Notice that the command prompt changes after you enter the ACL type and name. The “std” in the command prompt indicates that you are configuring entries for a standard ACL. For an extended ACL, this part of the command prompt is “ext”. The “nACL” indicates that you are configuring a named ACL.

Extended numbered ACL configuration

This section describes how to configure extended numbered ACLs.

Extended ACLs let you permit or deny packets based on the following information:

- IP protocol
- Source IP address or host name
- Destination IP address or host name
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)

The IP protocol can be one of the following well-known names or any IP protocol number from 0 – 255:

- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Gateway Routing Protocol (IGRP)
- Internet Protocol (IP)
- Open Shortest Path First (OSPF)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IP address to the website IP address.

Extended numbered ACL syntax

Syntax: `[no] access-list <ACL-num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-num> | <icmp-type>] <wildcard> [<tcp/udp comparison operator> <destination-tcp/udp-port>] [802.1p-priority-matching <0 - 7>] [dscp-cos-mapping] [dscp-marking <0-63>] [802.1p-priority-marking <0 - 7>... | dscp-cos-mapping] [dscp-matching <0-63>] [log] [precedence <name> | <0 - 7>] [tos <0 - 63> | <name>] [traffic policy <name>]`

Syntax: `[no] access-list <ACL-num> deny | permit host <ip-protocol> any any`

Syntax: `[no] ip access-group <ACL-num> in | out`

The `<ACL-num>` parameter is the extended access list number. Specify a number from 100 – 199.

The **deny** | **permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The *ip-protocol* parameter indicates the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter “?” instead of a protocol to list the well-known names recognized by the CLI.

The *source-ip* | *hostname* parameter specifies the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.

The *wildcard* parameter specifies the portion of the source IP host address to match against. The *wildcard* is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet's source address must match the *source-ip*. Ones mean any value matches. For example, the *source-ip* and *wildcard* values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in *"/ mask-bits "* format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The *destination-ip | hostname* parameter specifies the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any**.

The *icmp-type | icmp-num* parameter specifies the ICMP protocol type:

- This parameter applies only if you specified **icmp** as the *ip-protocol* value.
- If you use this parameter, the ACL entry is sent to the CPU for processing.
- If you do not specify a message type, the ACL applies to all types of ICMP messages.

The *icmp-num* parameter can be a value from 0 – 255.

The *icmp-type* parameter can have one of the following values, depending on the software version the device is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- log
- mask-reply
- mask-request
- parameter-problem
- redirect
- source-quench

- time-exceeded
- timestamp-reply
- timestamp-request
- traffic policy
- unreachable
- *num*

NOTE

The QoS options listed below are only available if a specific ICMP type is specified for the *icmp-type* parameter and cannot be used with the **any-icmp-type** option above. See [“QoS options for IP ACLs”](#) on page 1740 for more information on using ACLs to perform QoS.

The *tcp/udp comparison operator* parameter specifies a comparison operator for the TCP or UDP port number. This parameter applies only when you specify **tcp** or **udp** as the IP protocol. For example, if you are configuring an entry for HTTP, specify **tcp eq http**. You can enter one of the following operators:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to “1”) in the Control Bits field of the TCP packet header. Thus, the policy applies only to established TCP sessions, not to new sessions. Refer to Section 3.1, “Header Format”, in RFC 793 for information about this field.

NOTE

This operator applies only to destination TCP ports, not source TCP ports.

- **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – The policy applies to all TCP or UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The *tcp/udp-port* parameter specifies the TCP or UDP port number or well-known name. You can specify a well-known name for any application port whose number is less than 1024. For other application ports, you must enter the number. Enter “?” instead of a port to list the well-known names recognized by the CLI.

The **in | out** parameter specifies that the ACL applies to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or a virtual interface.

NOTE

If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. Refer to [“Configuring standard numbered ACLs”](#) on page 1709.

The **precedence** *name* | *num* parameter of the **ip access-list** command specifies the IP precedence. The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header. You can specify one of the following:

- **critical** or **5** – The ACL matches packets that have the critical precedence. If you specify the option number instead of the name, specify number 5.
- **flash** or **3** – The ACL matches packets that have the flash precedence. If you specify the option number instead of the name, specify number 3.
- **flash-override** or **4** – The ACL matches packets that have the flash override precedence. If you specify the option number instead of the name, specify number 4.
- **immediate** or **2** – The ACL matches packets that have the immediate precedence. If you specify the option number instead of the name, specify number 2.
- **internet** or **6** – The ACL matches packets that have the internetwork control precedence. If you specify the option number instead of the name, specify number 6.
- **network** or **7** – The ACL matches packets that have the network control precedence. If you specify the option number instead of the name, specify number 7.
- **priority** or **1** – The ACL matches packets that have the priority precedence. If you specify the option number instead of the name, specify number 1.
- **routine** or **0** – The ACL matches packets that have the routine precedence. If you specify the option number instead of the name, specify number 0.

The **tos** *name* | *num* parameter of the **ip access-list** command specifies the IP ToS. You can specify one of the following:

- **max-reliability** or **2** – The ACL matches packets that have the maximum reliability ToS. The decimal value for this option is 2.
- **max-throughput** or **4** – The ACL matches packets that have the maximum throughput ToS. The decimal value for this option is 4.
- **min-delay** or **8** – The ACL matches packets that have the minimum delay ToS. The decimal value for this option is 8.
- **min-monetary-cost** or **1** – The ACL matches packets that have the minimum monetary cost ToS. The decimal value for this option is 1.

NOTE

This value is not supported on 10 Gigabit Ethernet modules.

- **normal** or **0** – The ACL matches packets that have the normal ToS. The decimal value for this option is 0.
- *num* – A number from 0 – 15 that is the sum of the numeric values of the options you want. The ToS field is a four-bit field following the Precedence field in the IP header. You can specify one or more of the following. To select more than one option, enter the decimal value that is equivalent to the sum of the numeric values of all the ToS options you want to select. For example, to select the **max-reliability** and **min-delay** options, enter number 10. To select all options, select 15.

NOTE

The following QoS options are only available if a specific ICMP type is specified and cannot be used with the **any-icmp-type** option set for the *icmp-type* parameter. See [“QoS options for IP ACLs”](#) on page 1740 for more information on using ACLs to perform QoS.

The **802.1p-priority-matching** option inspects the 802.1p bit in the ACL that can be used with adaptive rate limiting. Enter a value from 0 – 7. For details, refer to [“Inspecting the 802.1p bit in the ACL for adaptive rate limiting”](#) on page 1781.

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE

The **dscp-cos-mapping** option overrides port-based priority settings.

NOTE

The **dscp-cos-mapping** option is **not** supported for FCX devices. The **dscp-cos-mapping** option is supported on FESX and FSX devices only.

The **dscp-marking** option enables you to configure an ACL that marks matching packets with a specified DSCP value. Enter a value from 0 – 63. Refer to [“Using an IP ACL to mark DSCP values \(DSCP marking\)”](#) on page 1742.

The **dscp-matching** option matches on the packet’s DSCP value. Enter a value from 0 – 63. This option does not change the packet’s forwarding priority through the device or mark the packet. Refer to [“DSCP matching”](#) on page 1744.

The **log** parameter enables SNMP traps and Syslog messages for inbound packets denied by the ACL:

- You can enable logging on inbound ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the **ACL** or **filter** command and add the **log** parameter to the end of the ACL or filter. The software replaces the **ACL** or **filter** command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **traffic-policy** option enables the device to rate limit inbound traffic and to count the packets and bytes per packet to which ACL permit or deny clauses are applied. For configuration procedures and examples, refer to the chapter [“Traffic Policies”](#) on page 1773.

Configuration examples for extended numbered ACLs

To configure an extended access control list that blocks all Telnet traffic received on port 1/1 from IP host 209.157.22.26, enter the following commands.

```
Brocade(config)#access-list 101 deny tcp host 209.157.22.26 any eq telnet log
Brocade(config)#access-list 101 permit ip any any
Brocade(config)#int eth 1/1
Brocade(config-if-e1000-1/1)#ip access-group 101 in
Brocade(config)#write memory
```

Here is another example of commands for configuring an extended ACL and applying it to an interface. These examples show many of the syntax choices. Notice that some of the entries are configured to generate log entries while other entries are not thus configured.

```

Brocade(config)#access-list 102 perm icmp 209.157.22.0/24 209.157.21.0/24
Brocade(config)#access-list 102 deny igmp host rkwong 209.157.21.0/24 log
Brocade(config)#access-list 102 deny igmp 209.157.21.0/24 host rkwong log
Brocade(config)#access-list 102 deny ip host 209.157.21.100 host 209.157.22.1 log
Brocade(config)#access-list 102 deny ospf any any log
Brocade(config)#access-list 102 permit ip any any

```

The first entry permits ICMP traffic from hosts in the 209.157.22.x network to hosts in the 209.157.21.x network.

The second entry denies IGMP traffic from the host device named “rkwong” to the 209.157.21.x network.

The third entry denies IGMP traffic from the 209.157.21.x network to the host device named “rkwong”.

The fourth entry denies all IP traffic from host 209.157.21.100 to host 209.157.22.1 and generates Syslog entries for packets that are denied by this entry.

The fifth entry denies all OSPF traffic and generates Syslog entries for denied traffic.

The sixth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 102 to the incoming traffic on port 1/2 and to the incoming traffic on port 4/3.

```

Brocade(config)#int eth 1/2
Brocade(config-if-1/2)#ip access-group 102 in
Brocade(config-if-1/2)#exit
Brocade(config)#int eth 4/3
Brocade(config-if-4/3)#ip access-group 102 in
Brocade(config)#write memory

```

Here is another example of an extended ACL.

```

Brocade(config)#access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
Brocade(config)#access-list 103 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
Brocade(config)#access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24 lt
telnet neq 5
Brocade(config)#access-list 103 deny udp any range 5 6 209.157.22.0/24 range 7 8
Brocade(config)#access-list 103 permit ip any any

```

The first entry in this ACL denies TCP traffic from the 209.157.21.x network to the 209.157.22.x network.

The second entry denies all FTP traffic from the 209.157.21.x network to the 209.157.22.x network.

The third entry denies TCP traffic from the 209.157.21.x network to the 209.157.22.x network, if the TCP port number of the traffic is less than the well-known TCP port number for Telnet (23), and if the TCP port is not equal to 5. Thus, TCP packets whose TCP port numbers are 5 or are greater than 23 are allowed.

The fourth entry denies UDP packets from any source to the 209.157.22.x network, if the UDP port number from the source network is 5 or 6 and the destination UDP port is 7 or 8.

The fifth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 103 to the incoming traffic on ports 2/1 and 2/2.

```
Brocade(config)#int eth 2/1
Brocade(config-if-2/1)#ip access-group 103 in
Brocade(config-if-2/1)#exit
Brocade(config)#int eth 0/2/2
Brocade(config-if-2/2)#ip access-group 103 in
Brocade(config)#write memory
```

Extended named ACL configuration

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL number with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

Extended ACLs let you permit or deny packets based on the following information:

- IP protocol
- Source IP address or host name
- Destination IP address or host name
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)

The IP protocol can be one of the following well-known names or any IP protocol number from 0 – 255:

- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Gateway Routing Protocol (IGRP)
- Internet Protocol (IP)
- Open Shortest Path First (OSPF)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IP address to the website's IP address.

Extended named ACL syntax

Syntax: `[no] ip access-list extended <ACL-name> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-num> | <icmp-type>] <wildcard> [<tcp/udp comparison operator> <destination-tcp/udp-port>] [802.1p-priority-matching <0 - 7>] [dscp-cos-mapping] [dscp-marking <0-63> [802.1p-priority-marking <0 - 7>... | dscp-cos-mapping]] [dscp-matching <0-63>] [log] [precedence <name> | <0 - 7>] [tos <0 - 63> | <name>] [traffic policy <name>]`

Syntax: `[no] ip access-group <num> in | out`

The *<ACL-name>* parameter is the access list name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, "ACL for Net1").

The **deny** | **permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The *ip-protocol* parameter indicates the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI.

The *source-ip* | *hostname* parameter specifies the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.

The *wildcard* parameter specifies the portion of the source IP host address to match against. The *wildcard* is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet's source address must match the *source-ip*. Ones mean any value matches. For example, the *source-ip* and *wildcard* values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in *" / mask-bits "* format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The *destination-ip | hostname* parameter specifies the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any**.

The *icmp-type | icmp-num* parameter specifies the ICMP protocol type:

- This parameter applies only if you specified **icmp** as the *ip-protocol* value.
- If you use this parameter, the ACL entry is sent to the CPU for processing.
- If you do not specify a message type, the ACL applies to all types of ICMP messages.

The *icmp-num* parameter can be a value from 0 – 255.

The *icmp-type* parameter can have one of the following values, depending on the software version the device is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- log
- mask-reply
- mask-request
- parameter-problem
- redirect
- source-quench
- time-exceeded
- timestamp-reply
- timestamp-request
- traffic policy
- unreachable
- *num*

NOTE

The QoS options listed below are only available if a specific ICMP type is specified for the *icmp-type* parameter and cannot be used with the **any-icmp-type** option above. See [“QoS options for IP ACLs”](#) on page 1740 for more information on using ACLs to perform QoS.

The *tcp/udp comparison operator* parameter specifies a comparison operator for the TCP or UDP port number. This parameter applies only when you specify **tcp** or **udp** as the IP protocol. For example, if you are configuring an entry for HTTP, specify **tcp eq http**. You can enter one of the following operators:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to “1”) in the Control Bits field of the TCP packet header. Thus, the policy applies only to established TCP sessions, not to new sessions. Refer to Section 3.1, “Header Format”, in RFC 793 for information about this field.

NOTE

This operator applies only to destination TCP ports, not source TCP ports.

- **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – The policy applies to all TCP or UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The *tcp/udp-port* parameter specifies the TCP or UDP port number or well-known name. You can specify a well-known name for any application port whose number is less than 1024. For other application ports, you must enter the number. Enter “?” instead of a port to list the well-known names recognized by the CLI.

The **in | out** parameter specifies that the ACL applies to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or a virtual interface.

NOTE

If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. Refer to [“Configuring standard numbered ACLs”](#) on page 1709.

The **precedence name | num** parameter of the **ip access-list** command specifies the IP precedence. The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet’s header. You can specify one of the following:

- **critical** or **5** – The ACL matches packets that have the critical precedence. If you specify the option number instead of the name, specify number 5.
- **flash** or **3** – The ACL matches packets that have the flash precedence. If you specify the option number instead of the name, specify number 3.
- **flash-override** or **4** – The ACL matches packets that have the flash override precedence. If you specify the option number instead of the name, specify number 4.
- **immediate** or **2** – The ACL matches packets that have the immediate precedence. If you specify the option number instead of the name, specify number 2.
- **internet** or **6** – The ACL matches packets that have the internetwork control precedence. If you specify the option number instead of the name, specify number 6.
- **network** or **7** – The ACL matches packets that have the network control precedence. If you specify the option number instead of the name, specify number 7.
- **priority** or **1** – The ACL matches packets that have the priority precedence. If you specify the option number instead of the name, specify number 1.
- **routine** or **0** – The ACL matches packets that have the routine precedence. If you specify the option number instead of the name, specify number 0.

The **tos name | num** parameter of the **ip access-list** command specifies the IP ToS. You can specify one of the following:

- **max-reliability** or **2** – The ACL matches packets that have the maximum reliability ToS. The decimal value for this option is 2.
- **max-throughput** or **4** – The ACL matches packets that have the maximum throughput ToS. The decimal value for this option is 4.
- **min-delay** or **8** – The ACL matches packets that have the minimum delay ToS. The decimal value for this option is 8.
- **min-monetary-cost** or **1** – The ACL matches packets that have the minimum monetary cost ToS. The decimal value for this option is 1.

NOTE

This value is not supported on 10 Gigabit Ethernet modules.

- **normal** or **0** – The ACL matches packets that have the normal ToS. The decimal value for this option is 0.
- *num* – A number from 0 – 15 that is the sum of the numeric values of the options you want. The ToS field is a four-bit field following the Precedence field in the IP header. You can specify one or more of the following. To select more than one option, enter the decimal value that is equivalent to the sum of the numeric values of all the ToS options you want to select. For example, to select the **max-reliability** and **min-delay** options, enter number 10. To select all options, select 15.

NOTE

The following QoS options are only available if a specific ICMP type is specified and cannot be used with the **any-icmp-type** option set for the *icmp-type* parameter. See [“QoS options for IP ACLs”](#) on page 1740 for more information on using ACLs to perform QoS.

The **802.1p-priority-matching** option inspects the 802.1p bit in the ACL that can be used with adaptive rate limiting. Enter a value from 0 – 7. For details, refer to [“Inspecting the 802.1p bit in the ACL for adaptive rate limiting”](#) on page 1781.

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE

The **dscp-cos-mapping** option overrides port-based priority settings.

NOTE

The **dscp-cos-mapping** option is **not** supported for FCX devices. The **dscp-cos-mapping** option is supported on FESX and FSX devices only.

The **dscp-marking** option enables you to configure an ACL that marks matching packets with a specified DSCP value. Enter a value from 0 – 63. Refer to [“Using an IP ACL to mark DSCP values \(DSCP marking\)”](#) on page 1742.

The **dscp-matching** option matches on the packet’s DSCP value. Enter a value from 0 – 63. This option does not change the packet’s forwarding priority through the device or mark the packet. Refer to [“DSCP matching”](#) on page 1744.

The **log** parameter enables SNMP traps and Syslog messages for inbound packets denied by the ACL:

- You can enable logging on inbound ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the **ACL** or **filter** command and add the **log** parameter to the end of the ACL or filter. The software replaces the **ACL** or **filter** command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **traffic-policy** option enables the device to rate limit inbound traffic and to count the packets and bytes per packet to which ACL permit or deny clauses are applied. For configuration procedures and examples, refer to the chapter [“Traffic Policies”](#) on page 1773.

Configuration example for extended named ACLs

To configure an extended named ACL, enter the **ip access-list extended <ACL_name>** command.

```
Brocade(config)#ip access-list extended "block Telnet"
Brocade(config-ext-nACL)#deny tcp host 209.157.22.26 any eq telnet log
Brocade(config-ext-nACL)#permit ip any any
Brocade(config-ext-nACL)#exit
Brocade(config)#int eth 1/1
Brocade(config-if-1/1)#ip access-group "block Telnet" in
```

The options at the ACL configuration level and the syntax for the **ip access-group** command are the same for numbered and named ACLs and are described in [“Extended numbered ACL configuration”](#) on page 1714 and [“Extended numbered ACL configuration”](#) on page 1714.

Applying egress ACLs to Control (CPU) traffic

By default, outbound ACLs are not applied to traffic generated by the CPU. This must be enabled using the **enable egress-acl-on-control-traffic** command.

Syntax: **enable egress-acl-on-control-traffic**

Preserving user input for ACL TCP/UDP port numbers

ACL implementations automatically display the TCP/UDP port name instead of the port number, regardless of user preference, unless the device is configured to preserve user input. When the option to preserve user input is enabled, the system will display either the port name or the number.

To enable this feature, enter the **ip preserve-ACL-user-input-format** command.

```
Brocade(config)#ip preserve-ACL-user-input-format
```

Syntax: **ip preserve-ACL-user-input-format**

The following example shows how this feature works for a TCP port (this feature works the same way for UDP ports). In this example, the user identifies the TCP port by number (80) when configuring ACL group 140. However, **show ip access-list 140** reverts back to the port name for the TCP port (http in this example). After the user issues the new **ip preserve-ACL-user-input-format** command, **show ip access-list 140** displays either the TCP port number or name, depending on how it was configured by the user.

```

Brocade(config)#access-list 140 permit tcp any any eq 80
Brocade(config)#access-list 140 permit tcp any any eq ftp
Brocade#show ip access-lists 140
Extended IP access list 140
permit tcp any any eq http
permit tcp any any eq ftp
Brocade(config)#ip preserve-ACL-user-input-format
Brocade#show ip access-lists 140
Extended IP access list 140
permit tcp any any eq 80
permit tcp any any eq ftp

```

ACL comment text management

ACL comment text describes entries in an ACL. The comment text appears in the output of **show** commands that display ACL information.

This section describes how to add, delete, and view ACL comments.

Adding a comment to an entry in a numbered ACL

To add comments to entries in a numbered ACL, enter commands such as the following.

```

Brocade(config)#access-list 100 remark The following line permits TCP packets
Brocade(config)#access-list 100 permit tcp 192.168.4.40/24 2.2.2.2/24
Brocade(config)#access-list 100 remark The following permits UDP packets
Brocade(config)#access-list 100 permit udp 192.168.2.52/24 2.2.2.2/24
Brocade(config)#access-list 100 deny ip any any

```

You can add comments to entries in a numbered ACL using the syntax for named ACLs. For example, using the same example configuration above, you could instead enter the following commands.

```

Brocade(config)#ip access-list extended 100
Brocade(config-ext-nACL)#remark The following line permits TCP packets
Brocade(config-ext-nACL)#permit tcp 192.168.4.40/24 2.2.2.2/24
Brocade(config-ext-nACL)#remark The following permits UDP packets
Brocade(config-ext-nACL)#permit udp 192.168.2.52/24 2.2.2.2/24
Brocade(config-ext-nACL)#deny ip any any

```

Syntax: [no] **access-list** <ACL-num> **remark** <comment-text>

or

Syntax: [no] **ip access-list** **standard** | **extended** <ACL-num>

Syntax: **remark** <comment-text>

For <ACL-num>, enter the number of the ACL.

The <comment-text> can be up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same **access-list** or **ip access-list** command. Also, in order for the remark to be displayed correctly in the output of **show** commands, the comment must be entered immediately before the ACL entry it describes. Note that an ACL comment is tied to the ACL entry immediately following the comment. Therefore, if the ACL entry is removed, the ACL comment is also removed.

The **standard** | **extended** parameter indicates the ACL type.

Adding a comment to an entry in a named ACL

To add comments to entries in a named ACL, enter commands such as the following.

```
Brocade(config)#ip access-list extended TCP/UDP
Brocade(config-ext-nACL)#remark The following line permits TCP packets
Brocade(config-ext-nACL)#permit tcp 192.168.4.40/24 2.2.2.2/24
Brocade(config-ext-nACL)#remark The following permits UDP packets
Brocade(config-ext-nACL)#permit udp 192.168.2.52/24 2.2.2.2/24
Brocade(config-ext-nACL)#deny ip any any
```

Syntax: [no] **access-list** **standard** | **extended** <ACL-name>

Syntax: **remark** <comment-text>

The **standard** | **extended** parameter indicates the ACL type.

For <ACL-name>, enter the name of the ACL.

The <comment-text> can be up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same **ip access-list** command. Also, in order for the remark to be displayed correctly in the output of show commands, the comment must be entered immediately before the ACL entry it describes. Note that an ACL comment is tied to the ACL entry immediately following the comment. Therefore, if the ACL entry is removed, the ACL comment is also removed.

Deleting a comment from an ACL entry

To delete a comment from an ACL entry, enter commands such as the following.

```
Brocade(config)#ip access-list standard 99
Brocade(config)#no remark The following line permits TCP packets
```

Syntax: **no remark** <comment-text>

Viewing comments in an ACL

You can use the following commands to display comments for ACL entries:

- **show running-config**
- **show access-list**
- **show ip access-list**

The following shows the comment text for a numbered ACL, ACL 100, in a **show running-config** display.

```
Brocade#show running-config
...
access-list 100 remark The following line permits TCP packets
access-list 100 permit tcp 192.168.4.40/24 2.2.2.2/24
access-list 100 remark The following line permits UDP packets
access-list 100 permit udp 192.168.2.52/24 2.2.2.2/24
access-list 100 deny ip any any
```

Syntax: show running-config

The following example shows the comment text for an ACL in a **show access-list** display. The output is identical in a **show ip access-list** display.

```
Brocade#show access-list
IP access list rate-limit 100 aaaa.bbbb.cccc
Extended IP access list TCP/UDP (Total flows: N/A, Total packets: N/A)
ACL Remark: The following line permits TCP packets
permit tcp 0.0.0.40 255.255.255.0 0.0.0.2 255.255.255.0 (Flows: N/A, Packets:
N/A)
ACL Remark: The following line permits UDP packets
permit udp 0.0.0.52 255.255.255.0 0.0.0.2 255.255.255.0 (Flows: N/A, Packets:
N/A)
deny ip any any (Flows: N/A, Packets: N/A)
```

Syntax: show access-list <ACL-num> | <ACL-name> | all

or

Syntax: show ip access-list <ACL-num> | <ACL-name> | all

Applying an ACL to a virtual interface in a protocol- or subnet-based VLAN

By default, when you apply an ACL to a virtual interface in a protocol-based or subnet-based VLAN, the ACL takes effect on all protocol or subnet VLANs to which the untagged port belongs. To prevent the Brocade device from denying packets on other virtual interfaces that do not have an ACL applied, configure an ACL that permits packets in the IP subnet of the virtual interface in all protocol-based or subnet-based VLANs to which the untagged port belongs. The following is an example configuration.

```
Brocade#conf t
Brocade(config)#vlan 1 name DEFAULT-VLAN by port
Brocade(config-vlan-1)#ip-subnet 192.168.10.0 255.255.255.0
Brocade(config-vlan-ip-subnet)#static ethe 1
Brocade(config-vlan-ip-subnet)#router-interface ve 10
Brocade(config-vlan-ip-subnet)#ip-subnet 10.15.1.0 255.255.255.0
Brocade(config-vlan-ip-subnet)#static ethe 1
Brocade(config-vlan-ip-subnet)#router-interface ve 20
Brocade(config-vlan-ip-subnet)#logging console
Brocade(config-vlan-ip-subnet)#exit
Brocade(config-vlan-1)#no vlan-dynamic-discovery
Vlan dynamic discovery is disabled
Brocade(config-vlan-1)#int e 2
Brocade(config-if-e1000-2)#disable
Brocade(config-if-e1000-2)#interface ve 10
Brocade(config-vif-10)#ip address 192.168.10.254 255.255.255.0
Brocade(config-vif-10)#int ve 20
Brocade(config-vif-20)#ip access-group test1 in
Brocade(config-vif-20)#ip address 10.15.1.10 255.255.255.0
Brocade(config-vif-20)#exit
Brocade(config)#ip access-list extended test1
```

```

Brocade(config-ext-nACL)#permit ip 10.15.1.0 0.0.0.255 any log
Brocade(config-ext-nACL)#permit ip 192.168.10.0 0.0.0.255 any log
Brocade(config-ext-nACL)#end
Brocade#

```

ACL logging

Brocade devices support ACL logging of inbound packets that are sent to the CPU for processing (denied packets).

NOTE

ACL logging is not supported for outbound packets or any packets that are processed in hardware (permitted packets).

You may want the software to log entries in the Syslog for packets that are denied by ACL filters. ACL logging is disabled by default; it must be explicitly enabled on a port.

When you enable logging for ACL entries, statistics for packets that match the deny conditions of the ACL entries are logged. For example, if you configure a standard ACL entry to deny all packets from source address 209.157.22.26, statistics for packets that are explicitly denied by the ACL entry are logged in the Syslog buffer and in SNMP traps sent by the Brocade device.

The first time an ACL entry denies a packet, the software immediately generates a Syslog entry and an SNMP trap. The software also starts a five-minute timer. The timer keeps track of all packets explicitly denied by the ACL entries. After five minutes, the software generates a single Syslog entry for each ACL entry that denied a packet. The Syslog entry (message) indicates the number of packets denied by the ACL entry during the previous five minutes. Note however that packet count may be inaccurate if the packet rate is high and exceeds the CPU processing rate.

If no ACL entries explicitly deny packets during an entire five-minute timer interval, the timer stops. The timer restarts when an ACL entry explicitly denies a packet.

NOTE

The timer for logging packets denied by MAC address filters is a different timer than the ACL logging timer.

Configuration notes for ACL logging

Note the following points before configuring ACL logging:

- ACL logging is supported for denied packets, which are sent to the CPU for logging. ACL logging is not supported for permitted packets.
- ACL logging is not supported for dynamic ACLs with multi-device port authentication and 802.1X.
- Packets that are denied by ACL filters are logged in the Syslog based on a sample time-period.
- You can enable ACL logging on physical and virtual interfaces.
- When ACL logging is disabled, packets that match the ACL rule are forwarded or dropped in hardware.
- ACL logging is supported on FCX and ICX devices for ACLs that are applied to network management access features such as Telnet, SSH, Web, and SNMP.

- When an ACL that includes an entry with a logging option is applied to a port that has logging enabled, and then the same ACL is applied to another port on the same system, traffic on the latter port is also logged, whether logging is explicitly enabled for that latter port or not.

On the other hand, when an ACL is applied to a port that has logging disabled, and then the same ACL is applied to another port on the same system, traffic on the latter port is also not logged, whether logging is explicitly enabled for that latter port or not.

NOTE

The above limitation applies only to IPv4 ACLs, it does not apply to the use of ACLs to log IPv6 traffic.

- When ACL logging is enabled on FastIron WS, Brocade FCX Series and ICX devices, packets sent to the CPU are automatically rate limited to prevent CPU overload.
- When ACL logging is enabled on FastIron X Series devices, Brocade recommends that you configure a traffic conditioner, then link the ACL to the traffic conditioner to prevent CPU overload. For example:

```
Brocade(config)#traffic-policy TPD1 rate-limit fixed 100 exceed-action drop
Brocade(config)#access-list 101 deny ip host 210.10.12.2 any traffic-policy
TPD1 log
```

- ACL logging is intended for debugging purposes. Brocade recommends that you disable ACL logging after the debug session is over.

Configuration tasks for ACL logging

To enable ACL logging, complete the following steps:

1. Create ACL entries with the log option
2. Enable ACL logging on individual ports

NOTE

The command syntax for enabling ACL logging is different on IPv4 devices than on IPv6 devices. See the configuration examples in the next section.

3. Bind the ACLs to the ports on which ACL logging is enabled

Example ACL logging configuration

The following shows an example ACL logging configuration on an IPv4 device.

```
Brocade(config)#access-list 1 deny host 209.157.22.26 log
Brocade(config)#access-list 1 deny 209.157.29.12 log
Brocade(config)#access-list 1 deny host IPHost1 log
Brocade(config)#access-list 1 permit any
Brocade(config)#interface e 1/4
Brocade(config-if-e1000-1/4)#ACL-logging
Brocade(config-if-e1000-1/4)#ip access-group 1 in
```

The above commands create ACL entries that include the log option, enable ACL logging on interface e 1/4, then bind the ACL to interface e 1/4. Statistics for packets that match the deny statements will be logged.

NOTE

The **ACL-logging** command shown above is **not** required for FWS devices.

Syntax: ACL-logging

The **ACL-logging** command applies to IPv4 devices only. For IPv6 devices, use the **logging-enable** command as shown in the following example.

The following shows an example configuration on an IPv6 device.

```
Brocade(config)#ipv6 acc ACL_log_v6
Brocade(config-ipv6-access-list ACL_log_v6)#logging-enable
Brocade(config-ipv6-access-list ACL_log_v6)# deny ipv6 host 2001::1 any log
Brocade(config-ipv6-access-list ACL_log_v6)#inter e 9/12
Brocade(config-if-e1000-9/12)#ipv6 traffic-filter ACL_log_v6 in
```

The above commands create ACL entries that include the log option, then bind the ACL to interface e 9/12. Statistics for packets that match the deny statement will be logged.

Syntax: logging-enable

NOTE

The **logging-enabled** command applies to IPv6 devices only. For IPv4 devices, use the **ACL-logging** command as shown in the previous example.

Displaying ACL Log Entries

The first time an entry in an ACL permits or denies a packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for packets permitted or denied by ACLs are at the warning level of the Syslog.

When the first Syslog entry for a packet permitted or denied by an ACL is generated, the software starts an ACL timer. After this, the software sends Syslog messages every five minutes. If an ACL entry does not permit or deny any packets during the timer interval, the software does not generate a Syslog entry for that ACL entry.

NOTE

For an ACL entry to be eligible to generate a Syslog entry for denied packets, logging must be enabled for the entry. The Syslog contains entries only for the ACL entries that deny packets and have logging enabled.

To display Syslog entries, enter the **show log** command from any CLI prompt:

Enabling strict control of ACL filtering of fragmented packets

```
Brocade#show log
Syslog logging: enabled (0 messages dropped, 2 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 9 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
              I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.6(0)(Ethernet 4 0000.0804.01
20.20.18.6(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.2(0)(Ethernet 4 0000.0804.01
20.20.18.2(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.4(0)(Ethernet 4 0000.0804.01
20.20.18.4(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.3(0)(Ethernet 4 0000.0804.01
20.20.18.3(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.5(0)(Ethernet 4 0000.0804.01
20.20.18.5(0), 1 event(s)
0d00h12m18s:I:ACL: 122 applied to port 4 by  from console session
0d00h10m12s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m56s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m38s:I:ACL: 122 removed from port 4 by  from console session
```

Syntax: show log

Enabling strict control of ACL filtering of fragmented packets

The default processing of fragments by hardware-based ACLs is as follows:

- The first fragment of a packet is permitted or denied using the ACLs. The first fragment is handled the same way as non-fragmented packets, since the first fragment contains the Layer 4 source and destination application port numbers. The device uses the Layer 4 CAM entry if one is programmed, or applies the interface's ACL entries to the packet and permits or denies the packet according to the first matching ACL.
- For other fragments of the same packet, they are subject to a rule only if there is no Layer 4 information in the rule or in any preceding rules.

The fragments are forwarded even if the first fragment, which contains the Layer 4 information, was denied. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.

For tighter control, you can configure the port to drop all packet fragments. To do so, enter commands such as the following.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#ip access-group frag deny
```

This option begins dropping all fragments received by the port as soon as you enter the command. This option is especially useful if the port is receiving an unusually high rate of fragments, which can indicate a hacker attack.

Syntax: [no] ip access-group frag deny

Enabling ACL support for switched traffic in the router image

NOTE

The **bridged-routed** CLI parameter applies to FSX and FESX devices only. For FWS, Brocade FCX Series and ICX devices, ACL support for switched traffic in the router image is enabled by default. There is no command to enable or disable it.

For outbound traffic, ACL support is enabled on switched traffic by default. The **bridged-routed** command is not applicable.

By default, when an ACL is applied to a physical or virtual routing interface, the Brocade Layer 3 device filters routed traffic only. It does not filter traffic that is switched from one port to another within the same VLAN or virtual routing interface, even if an ACL is applied to the interface.

You can enable the device to filter switched traffic within a VLAN or virtual routing interface. When filtering is enabled, the device uses the ACLs applied to inbound traffic to filter traffic received by a port from another port in the same virtual routing interface.

To enable this feature, enter a command such as the following.

```
Brocade(config)# access-list 101 bridged-routed
```

Applying the ACL rule above to an interface enables filtering of traffic switched within a VLAN or virtual routing interface.

Syntax: [no] **access-list** <ACL-ID> **bridged-routed**

The <ACL-ID> parameter specifies a standard or extended numbered or named ACL.

You can use the **bridged-routed** feature in conjunction with **enable ACL-per-port-per-vlan**, to assign an ACL to certain ports of a VLAN under the virtual interface configuration level. In this case, all of the Layer 3 traffic (bridged and routed) are filtered by the ACL. The following shows an example configuration.

```
Brocade(config)#vlan 101 by port
Brocade(config-vlan-101)#tagged ethernet 1 to 4
Brocade(config-vlan-101)#router-interface ve 101
Brocade(config-vlan-101)#exit
Brocade(config)#enable ACL-per-port-per-vlan
Brocade(config)#access-list 101 bridged-routed
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
...
Brocade(config-vif-101)#ip access group 1 in ethernet 1 ethernet 3 ethernet 4
```

NOTE

For FastIron X Series devices, the **enable ACL-per-port-per-vlan** command must be followed by the **write-memory** and **reload** commands to place the change into effect.

Enabling ACL filtering based on VLAN membership or VE port membership

NOTE

This section applies to IPv4 ACLs only. IPv6 ACLs do not support ACL filtering based on VLAN membership or VE port membership.

This feature is not applicable to outbound traffic.

You can apply an inbound IPv4 ACL to specific VLAN members on a port (Layer 2 devices only) or to specific ports on a virtual interface (VE) (Layer 3 Devices only). By default, this feature support is disabled. To enable it, enter the following commands at the Global CONFIG level of the CLI.

```
Brocade(config)#enable ACL-per-port-per-vlan
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
```

NOTE

For complete configuration examples, see [“Applying an IPv4 ACL to specific VLAN members on a port \(Layer 2 devices only\)”](#) on page 1734 and [“Applying an IPv4 ACL to a subset of ports on a virtual interface \(Layer 3 devices only\)”](#) on page 1735.

NOTE

For FastIron X Series devices, you must save the configuration and reload the software to place the change into effect.

Syntax: [no] enable ACL-per-port-per-vlan

Enter the **no** form of the command to disable this feature.

Configuration notes for ACL filtering

- Before enabling this feature on an IPv4 device, make sure the VLAN numbers are contiguous. For example, the VLAN numbers can be 201, 202, 203, and 204, but not 300, 401, 600, and 900.
- Brocade devices do not support a globally-configured PBR policy together with per-port-per-VLAN ACLs.
- IPv4 ACLs that filter based on VLAN membership or VE port membership (ACL-per-port-per-VLAN), are supported together with IPv6 ACLs on the same device, as long as they are not bound to the same port or virtual interface.

Applying an IPv4 ACL to specific VLAN members on a port (Layer 2 devices only)

NOTE

This section applies to IPv4 ACLs only. IPv6 ACLs do not support ACL filtering based on VLAN membership.

When you bind an IPv4 ACL to a port, the port filters all inbound traffic on the port. However, on a tagged port, there may be a need to treat packets for one VLAN differently from packets for another VLAN. In this case, you can configure a tagged port on a Layer 2 device to filter packets based on the packets' VLAN membership.

To apply an IPv4 ACL to a specific VLAN on a port, enter commands such as the following.

```
Brocade(config)#enable ACL-per-port-per-vlan
...
Brocade(config)#vlan 12 name vlan12
Brocade(config-vlan-12)#untag ethernet 5 to 8
Brocade(config-vlan-12)#tag ethernet 23 to 24
Brocade(config-vlan-12)#exit
Brocade(config)#access-list 10 deny host 209.157.22.26 log
Brocade(config)#access-list 10 deny 209.157.29.12 log
Brocade(config)#access-list 10 deny host IPhost1 log
Brocade(config)#access-list 10 permit
Brocade(config)#int e 1/23
Brocade(config-if-e1000-1/23)#per-vlan 12
Brocade(config-if-e1000-1/23-vlan-12)#ip access-group 10 in
```

NOTE

For FastIron X Series devices, the **enable ACL-per-port-per-vlan** command must be followed by the **write-memory** and **reload** commands to place the change into effect. .

The commands in this example configure port-based VLAN 12, and add ports e 5 – 8 as untagged ports and ports e 23 – 24 as tagged ports to the VLAN. The commands following the VLAN configuration commands configure ACL 10. Finally, the last three commands apply ACL 10 on VLAN 12 for which port e 23 is a member.

Syntax: **per-vlan** <VLAN ID>

Syntax: **[no] ip access-group** <ACL ID>

The <VLAN ID> parameter specifies the VLAN name or number to which you will bind the ACL.

The <ACL ID> parameter is the access list name or number.

Applying an IPv4 ACL to a subset of ports on a virtual interface (Layer 3 devices only)

NOTE

This section applies to IPv4 ACLs only. IPv6 ACLs do not support ACL filtering based on VE port membership.

You can apply an IPv4 ACL to a virtual routing interface. The virtual interface is used for routing between VLANs and contains all the ports within the VLAN. The IPv4 ACL applies to all the ports on the virtual routing interface. You also can specify a subset of ports within the VLAN containing a specified virtual interface when assigning an ACL to that virtual interface.

Use this feature when you do not want the IPv4 ACLs to apply to all the ports in the virtual interface VLAN or when you want to streamline IPv4 ACL performance for the VLAN.

To apply an ACL to a subset of ports within a virtual interface, enter commands such as the following.

```
Brocade(config)#enable ACL-per-port-per-vlan
...
Brocade(config)#vlan 10 name IP-subnet-vlan
Brocade(config-vlan-10)#untag ethernet 1/1 to 2/12
Brocade(config-vlan-10)#router-interface ve 1
Brocade(config-vlan-10)#exit
Brocade(config)#access-list 1 deny host 209.157.22.26 log
Brocade(config)#access-list 1 deny 209.157.29.12 log
Brocade(config)#access-list 1 deny host IPhost1 log
Brocade(config)#access-list 1 permit any
Brocade(config)#interface ve 1/1
Brocade(config-vif-1/1)#ip access-group 1 in ethernet 1/1 ethernet 1/3 ethernet
2/1 to 2/4
```

NOTE

For FastIron X Series devices, the **enable ACL-per-port-per-vlan** command must be followed by the **write-memory** and **reload** commands to place the change into effect.

The commands in this example configure port-based VLAN 10, add ports 1/1 – 2/12 to the VLAN, and add virtual routing interface 1 to the VLAN. The commands following the VLAN configuration commands configure ACL 1. Finally, the last two commands apply ACL 1 to a subset of the ports associated with virtual interface 1.

Syntax: [no] **ip access-group** <ACL ID> **in ethernet** <port> [to <port>]

The <ACL ID> parameter is the access list name or number.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

ACLs to filter ARP packets

NOTE

This feature is not applicable to outbound traffic.

You can use ACLs to filter ARP packets. Without this feature, ACLs cannot be used to permit or deny incoming ARP packets. Although an ARP packet contains an IP address just as an IP packet does, an ARP packet is not an IP packet; therefore, it is not subject to normal filtering provided by ACLs.

When a Brocade device receives an ARP request, the source MAC and IP addresses are stored in the device ARP table. A new record in the ARP table overwrites existing records that contain the same IP address. This behavior can cause a condition called "ARP hijacking", when two hosts with the same IP address try to send an ARP request to the Brocade device.

Normally ARP hijacking is not a problem because IP assignments are done dynamically; however, in some cases, ARP hijacking can occur, such as when a configuration allows a router interface to share the IP address of another router interface. Since multiple VLANs and the router interfaces that are associated with each of the VLANs share the same IP segment, it is possible for two hosts in two different VLANs to fight for the same IP address in that segment. ARP filtering using ACLs protects an IP host record in the ARP table from being overwritten by a hijacking host. Using ACLs to filter ARP requests checks the source IP address in the received ARP packet. Only packets with the permitted IP address will be allowed to be written in the ARP table; others are dropped.

Configuration considerations for filtering ARP packets

- This feature is available on devices running Layer 3 code. This filtering occurs on the management processor.
- The feature is available on physical interfaces and virtual routing interfaces. It is supported on the following physical interface types Ethernet and trunks.
- ACLs used to filter ARP packets on a virtual routing interface can be inherited from a previous interface if the virtual routing interface is defined as a follower virtual routing interface.

Configuring ACLs for ARP filtering

To implement the ACL ARP filtering feature, enter commands such as the following.

```
Brocade(config)# access-list 101 permit ip host 192.168.2.2 any
Brocade(config)# access-list 102 permit ip host 192.168.2.3 any
Brocade(config)# access-list 103 permit ip host 192.168.2.4 any
Brocade(config)# vlan 2
Brocade(config-vlan-2)# tag ethe 1/1 to 1/2
Brocade(config-vlan-2)# router-interface ve 2
Brocade(config-vlan-2)# vlan 3
Brocade(config-vlan-3)# tag ethe 1/1 to 1/2
Brocade(config-vlan-3)# router-int ve 3
Brocade(config-vlan-3)# vlan 4
Brocade(config-vlan-4)# tag ethe 1/1 to 1/2
Brocade(config-vlan-4)# router-int ve 4
Brocade(config-vlan-4)# interface ve 2
Brocade(config-ve-2)# ip access-group 101 in
Brocade(config-ve-2)# ip address 192.168.2.1/24
Brocade(config-ve-2)# ip use-ACL-on-arp 103
Brocade(config-ve-2)# exit
Brocade(config)# interface ve 3
Brocade(config-ve-3)# ip access-group 102 in
Brocade(config-ve-3)# ip follow ve 2
Brocade(config-ve-3)# ip use-ACL-on-arp
Brocade(config-ve-3)# exit
Brocade(config-vlan-4)# interface ve 4
Brocade(config-ve-4)# ip follow ve 2
Brocade(config-ve-4)# ip use-ACL-on-arp
Brocade(config-ve-4)# exit
```

Syntax: [no] ip use-ACL-on-arp [<access-list-number>]

When the **use-ACL-on-arp** command is configured, the ARP module checks the source IP address of the ARP request packets received on the interface. It then applies the specified ACL policies to the packet. Only the packet with the IP address that the ACL permits will be allowed to be written in the ARP table; those that are not permitted will be dropped.

The *<access-list-number>* parameter identifies the ID of the standard ACL that will be used to filter the packet. Only the source and destination IP addresses will be used to filter the ARP packet. You can do one of the following for *<access-list-number>*:

- Enter an ACL ID to explicitly specify the ACL to be used for filtering. In the example above, the line `FastIron(config-ve-2)# ip use-ACL-on-arp 103` specifies ACL 103 to be used as the filter.
- Allow the ACL ID to be inherited from the IP ACLs that have been defined for the device. In the example above, the line `FastIron(config-ve-4)# ip use-ACL-on-arp` allows the ACL to be inherited from IP ACL 101 because of the `ip follow` relationship between virtual routing interface 2 and virtual routing interface 4. Virtual routing interface 2 is configured with IP ACL 101; thus virtual routing interface 4 inherits IP ACL 101.

ARP requests will not be filtered by ACLs if one of the following conditions occur:

- If the ACL is to be inherited from an IP ACL, but there is no IP ACL defined.
- An ACL ID is specified for the `use-ACL-on-arp` command, but no IP address or “any any” filtering criteria have been defined under the ACL ID.

Displaying ACL filters for ARP

To determine which ACLs have been configured to filter ARP requests, enter a command such as the following.

```
Brocade(config)# show ACL-on-arp
Port ACL ID Filter Count
2 103 10
3 102 23
4 101 12
```

Syntax: `show ACL-on-arp [ethernet <port> | loopback [<num>] | ve [<num>]]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

If the *<port>* variable is not specified, all ports on the device that use ACLs for ARP filtering will be included in the display.

The Filter Count column shows how many ARP packets have been dropped on the interface since the last time the count was cleared.

Clearing the filter count

To clear the filter count for all interfaces on the device, enter a command such as the following.

```
Brocade(config)# clear ACL-on-arp
```

The above command resets the filter count on all interfaces in a device back to zero.

Syntax: `clear ACL-on-arp`

Filtering on IP precedence and ToS values

To configure an extended IP ACL that matches based on IP precedence, enter commands such as the following.

```
Brocade(config)#access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
precedence internet
Brocade(config)#access-list 103 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
precedence 6
Brocade(config)#access-list 103 permit ip any any
```

The first entry in this ACL denies TCP traffic from the 209.157.21.x network to the 209.157.22.x network, if the traffic has the IP precedence option “internet” (equivalent to “6”).

The second entry denies all FTP traffic from the 209.157.21.x network to the 209.157.22.x network, if the traffic has the IP precedence value “6” (equivalent to “internet”).

The third entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

To configure an IP ACL that matches based on ToS, enter commands such as the following.

```
Brocade(config)#access-list 104 deny tcp 209.157.21.0/24 209.157.22.0/24 tos
normal
Brocade(config)#access-list 104 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
tos 13
Brocade(config)#access-list 104 permit ip any any
```

The first entry in this IP ACL denies TCP traffic from the 209.157.21.x network to the 209.157.22.x network, if the traffic has the IP ToS option “normal” (equivalent to “0”).

The second entry denies all FTP traffic from the 209.157.21.x network to the 209.157.22.x network, if the traffic has the IP ToS value “13” (equivalent to “max-throughput”, “min-delay”, and “min-monetary-cost”).

The third entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

TCP flags - edge port security

The edge port security feature works in combination with IP ACL rules and can be combined with other ACL functions (such as dscp-marking and traffic policies), giving you greater flexibility when designing ACLs.

For details about the edge port security feature, refer to [“Using TCP Flags in combination with other ACL features”](#) on page 178.

QoS options for IP ACLs

Quality of Service (QoS) options enable you to perform QoS for packets that match the ACLs. Using an ACL to perform QoS is an alternative to directly setting the internal forwarding priority based on incoming port, VLAN membership, and so on. (This method is described in [“QoS priorities-to-traffic assignment”](#) on page 1978.)

The following QoS ACL options are supported:

- **dscp-cos-mapping** – This option is similar to the **dscp-matching** command (described below). This option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

By default, the Brocade device does the *802.1p* to CoS mapping. If you want to change the priority mapping to *DSCP* to CoS mapping, you must enter the following ACL statement.

```
permit ip any any dscp-cos-mapping
```

- **dscp-marking** – Marks the DSCP value in the outgoing packet with the value you specify.
- **internal-priority-marking** and **802.1p-priority-marking** – Supported with the DSCP marking option, these commands assign traffic that matches the ACL to a hardware forwarding queue (**internal-priority-marking**), and re-mark the packets that match the ACL with the 802.1p priority (**802.1p-priority-marking**).
- **dscp-matching** – Matches on the packet DSCP value. This option does not change the packet forwarding priority through the device or mark the packet.
- **802.1p-priority-matching** – Inspects the 802.1p bit in the ACL that can be used with adaptive rate limiting. For details, refer to [“Inspecting the 802.1p bit in the ACL for adaptive rate limiting”](#) on page 1781.

NOTE

These QoS options are only available if a specific ICMP type is specified for the *<icmp-type>* parameter while configuring extended ACLs, and cannot be used with the **any-icmp-type** option. See [“Extended numbered ACL syntax”](#) on page 1714 and [“Extended named ACL syntax”](#) on page 1721 for the syntax for configuring extended ACLs.

Configuration notes for QoS options on FCX and ICX devices

- These devices do not support marking and prioritization simultaneously with the same rule (and do not support DSCP CoS mapping at all). To achieve this, you need to create two separate rules. In other words, you can mark a rule with DSCP or 802.1p information, or you can prioritize a rule based on DSCP or 802.1p information. You can enable only one of the following ACL options per rule:
 - 802.1p-priority-marking
 - dscp-marking
 - internal-priority-marking

For example, any one of the following commands is supported.

```
FastIron(config)#access-list 101 permit ip any any dscp-marking 43
```

or

```
FastIron(config)#access-list 101 permit ip any any 802.1p-priority-marking
```

or

```
FastIron(config)#access-list 101 permit ip any any internal-priority-marking 6
```

The following command is not supported.

```
FastIron(config)#access-list 101 permit ip any any dscp-marking 43
802.1p-priority-marking 4 internal-priority-marking 6
```

Using an ACL to map the DSCP value (DSCP CoS mapping)

NOTE

The **dscp-cos-mapping** option is supported on FSX and FESX devices only. It is not supported on Stackable devices. This feature is not applicable to outbound traffic.

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE

The **dscp-cos-mapping** option overrides port-based priority settings.

By default, the Brocade device does the 802.1p to CoS mapping. If you want to change the priority mapping to DSCP to CoS mapping, you must enter the following ACL statement.

```
permit ip any any dscp-cos-mapping
```

The complete CLI syntax for this feature is shown in [“Extended numbered ACL configuration”](#) on page 1714 and [“Extended named ACL configuration”](#) on page 1720. The following shows the syntax specific to the DSCP Cos mapping feature.

Syntax: ... [**dscp-marking** <dscp-value> **dscp-cos-mapping**]

or

Syntax: ...[**dscp-cos-mapping**]

NOTE

The **dscp-cos-mapping** option should not be used when assigning an 802.1p priority. To assign an 802.1p priority to a specific DSCP (using **dscp-match**), re-assign the DSCP value as well. For example

```
FastIron(config)#access-list 100 permit ip any any dscp-match 0 dscp-marking 0
802.1p 0 internal 1
```

Using an IP ACL to mark DSCP values (DSCP marking)

The **dscp-marking** option for extended ACLs allows you to configure an ACL that marks matching packets with a specified DSCP value. You also can use DSCP marking to assign traffic to a specific hardware forwarding queue (refer to [“Using an ACL to change the forwarding queue”](#) on page 1743).

For example, the following commands configure an ACL that marks all IP packets with DSCP value 5. The ACL is then applied to incoming packets on interface 7. Consequently, all inbound packets on interface 7 are marked with the specified DSCP value.

```
Brocade(config)#access-list 120 permit ip any any dscp-marking 5 dscp-cos-mapping
Brocade(config)#interface 1/7
Brocade(config-if-e1000-1/7)#ip access-group 120 in
```

Syntax: ...**dscp-marking** <dscp-value>

The **dscp-marking** <dscp-value> parameter maps a DSCP value to an internal forwarding priority. The DSCP value can be from 0 – 63.

NOTE

The **dscp-cos-mapping** option is supported on FESX and FSX devices only.

Combined ACL for 802.1p marking

Brocade devices support a simple method for assigning an 802.1p priority value to packets without affecting the actual packet or the DSCP. In early IronWare software releases, users were required to provide DSCP-marking and DSCP-matching information in order to assign 802.1p priority values, which required the deployment of a 64-line ACL to match all possible DSCP values. Users were also required to configure an internal priority marking value. Now, users can easily specify 802.1p priority marking values directly, and change internal priority marking from *required* to *optional*.

NOTE

This feature is not applicable to outbound traffic.

On the following devices, if the user does not set a specific internal marking priority, the default value is the same as the 802.1p-priority marking value:

- FWS, FCX, FESX, and ICX devices
- FastIron SX modules, with the exception of SX-48GCPP modules, released prior to hardware release 07.3.00, including:
 - SX-FI624C
 - SX-FI624HF
 - SX-FI62XG
 - SX-FI42XG
 - SX-FI424C
 - SX-FI424F
 - SX-FI8GMR6
 - SX-FI2XGMR4

On the following devices, if the user does not set a specific internal marking priority, then the internal priority does not change:

- SX-48GCPP modules
- All FastIron SX modules released in hardware release 07.3.00 and later releases, including:
 - SX-FI24GPP
 - SX-FI24HF
 - SX-FI2XG
 - SX-FI8XG

Priority values range from 0 to 7.

Two new ACL parameters support this feature, one required for priority marking and one optional for internal priority marking. These parameters apply to IP, and TCP, and UDP.

For IP

```
Brocade(config)#acc 104 per ip any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value.

```
Brocade(config)#acc 104 per ip any any 802.1p-priority-marking 1
internal-priority-marking 5
```

Syntax: `access-list <num(100-199)> permit ip any any 802.1p-priority-marking <priority value (0-7)> [internal-priority-marking <value (0-7)>]`

For TCP

```
Brocade(config)#acc 105 per tcp any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value.

```
Brocade(config)#acc 105 per tcp any any 802.1p-priority-marking 1
internal-priority-marking 5
```

Syntax: `access-list <num(100-199)> permit tcp any any 802.1p-priority-marking <priority value (0-7)> [internal-priority-marking <value (0-7)>]`

For UDP

```
Brocade(config) #acc 105 per udp any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value.

```
Brocade(config) #acc 105 per udp any any 802.1p-priority-marking 1
internal-priority-marking 5
```

Syntax: `access-list <num(100-199)> permit udp any any 802.1p-priority-marking <priority value (0-7)> [internal-priority-marking <value (0-7)>]`

In each of these examples, in the first command the internal-priority value is not specified, which means it maintains a default value of 1 (equal to that of the 802.1p value). In the second command, the internal-priority value has been configured by the user to 5.

Using an ACL to change the forwarding queue

The **802.1p-priority-marking** <0 – 7> parameter re-marks the packets of the 802.1Q traffic that match the ACL with this new 802.1p priority, or marks the packets of the non-802.1Q traffic that match the ACL with this 802.1p priority, later at the outgoing 802.1Q interface.

The 802.1p priority mapping is shown in [Table 285](#).

The **internal-priority-marking** <0 – 7> parameter assigns traffic that matches the ACL to a specific hardware forwarding queue (qosp0 – qosp7>).

NOTE

The **internal-priority-marking** parameter overrides port-based priority settings.

On the FCX platform, using either **802.1p-priority-marking** or **802.1p-priority-marking** with **internal-priority-marking** performs both marking and internal prioritization.

In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1Q interface, this parameter maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority. [Table 285](#) lists the default mappings of hardware forwarding queues to 802.1p priorities on the FESX and FSX.

TABLE 285 Default mapping of forwarding queues to 802.1p priorities

Forwarding queue	qosp0	qosp1	qosp2	qosp3	qosp4	qosp5	qosp6	qosp7
802.1p	0	1	2	3	4	5	6	7

The complete CLI syntax for 802.1p priority marking and internal priority marking is shown in [“Extended numbered ACL configuration”](#) on page 1714 and [“Extended named ACL configuration”](#) on page 1720. The following shows the syntax specific to these features.

Syntax: ... **dscp-marking** <0 – 63> **802.1p-priority-marking** <0 – 7> **internal-priority-marking** <0 – 7>]

DSCP matching

The **dscp-matching** option matches on the packet DSCP value. This option does not change the packet forwarding priority through the device or mark the packet.

To configure an ACL that matches on a packet with DSCP value 29, enter a command such as the following.

```
Brocade(config)#access-list 112 permit ip 1.1.1.0 0.0.0.255 2.2.2.x 0.0.0.255
dscp-matching 29
```

The complete CLI syntax for this feature is shown in [“Extended numbered ACL configuration”](#) on page 1714 and [“Extended named ACL configuration”](#) on page 1720. The following shows the syntax specific to this feature.

Syntax: ...**dscp-matching** <0 – 63>

NOTE

For complete syntax information, refer to [“Extended numbered ACL syntax”](#) on page 1714.

ACL-based rate limiting

ACL-based rate limiting provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs. This feature is available in the Layer 2 and Layer 3 code.

NOTE

Brocade devices support ACL-based rate limiting for inbound traffic. This feature is not supported for outbound traffic.

For more details, including configuration procedures, refer to [Chapter 42, “Traffic Policies”](#).

ACL statistics

ACL statistics is a mechanism for counting the number of packets and the number of bytes per packet to which ACL filters are applied.

To see the configuration procedures for ACL statistics, refer to [Chapter 42, “Traffic Policies”](#).

NOTE

The terms *ACL statistics* and *ACL counting* are used interchangeably in this guide and mean the same thing.

ACLs to control multicast features

You can use ACLs to control the following multicast features:

- Limit the number of multicast groups that are covered by a static rendezvous point (RP)
- Control which multicast groups for which candidate RPs sends advertisement messages to bootstrap routers
- Identify which multicast group packets will be forwarded or blocked on an interface

For configuration procedures, refer to [Chapter 36, “IP Multicast Protocols”](#).

Enabling and viewing hardware usage statistics for an ACL

The number of configured ACL rules can affect the rate at which hardware resources are used. You can use the **show access-list hw-usage on** command to enable hardware usage statistics, followed by the **show access-list <access-list-id>** command to determine the hardware usage for an ACL. To gain more hardware resources, you can modify the ACL rules so that it uses less hardware resource.

To enable and view hardware usage statistics, enter commands such as the following:

```
Brocade#show access-list hw-usage on
Brocade#show access-list 100
Extended IP access list 100 (hw usage : 2)
deny ip any any (hw usage : 1
```

The first command enables hardware usage statistics, and the second command displays the hardware usage for IP access list 100.

NOTE

Hardware usage statistics for ACLs differ for SX 800 and SX 1600 devices with one or more SX-FI48GPP interface modules, compared to devices that do not have this interface module.

The following displays an example of the show output for an SX 800 device in which an SX-FI48GPP interface module is installed.

```
Brocade#show access-list all
```

```
Standard IP access list 1 (hw usage (if applied on 24GC modules) : 2) (hw usage
(if applied on 48GC modules) : 2)
permit any (hw usage (if applied on 24GC modules) : 1) (hw usage (if applied on
48GC modules) : 1)
```

```
Extended IP access list 100 (hw usage (if applied on 24GC modules) : 7) (hw
usage (if applied on 48GC modules) : 7)
deny tcp any range newacct src any (hw usage (if applied on 24GC modules) : 6)
(hw usage (if applied on 48GC modules) : 6)
```

```
FastIron SX 800 Router#sh mod
```

Module	Status	Ports	Starting MAC
F1: SX-FISF Switch Fabric	active		
F2: SX-FISF Switch Fabric	active		
S1:			
S2:			
S3: Configured as SX-FI648 48-port 100/1000 Copper			
S4: SX-FI648PP 48-port 100/1000 Copper	OK	48	0012.f227.7918
S5: SX-FI624C 24-port Gig Copper	OK	24	0012.f227.7960
S6:			
S7: SX-FI624C 24-port Gig Copper	OK	24	0012.f227.7990
S8:			
S9: SX-FIZMR6 0-port Management	Standby	0	
{ Status : OK }			
S10: SX-FIZMR6 0-port Management	Active	0	
{ Status : OK }			

Syntax: show access-list hw-usage on | off

Syntax: show access-list <access-list-id> | all

By default, hardware usage statistics are disabled. To disable hardware usage statistics after is has been enabled, use the **show access-list hw-usage off** command.

The <access-list-id> variable is a valid ACL name or number.

Displaying ACL information

To display the number of Layer 4 CAM entries used by each ACL, enter the following command.

```
Brocade#show access-list all
```

```
Extended IP access list 100 (Total flows: N/A, Total packets: N/A, Total rule cam
use: 3)
permit udp host 192.168.2.169 any (Flows: N/A, Packets: N/A, Rule cam use: 1)
permit icmp any any (Flows: N/A, Packets: N/A, Rule cam use: 1)
deny ip any any (Flows: N/A, Packets: N/A, Rule cam use: 1)
```

Syntax: show access-list <ACL-num> | <ACL-name> | all

The Rule cam use field lists the number of CAM entries used by the ACL or entry. The number of CAM entries listed for the ACL itself is the total of the CAM entries used by the ACL entries.

For flow-based ACLs, the Total flows and Flows fields list the number of Layer 4 session table flows in use for the ACL.

The Total packets and Packets fields apply only to flow-based ACLs.

Troubleshooting ACLs

Use the following methods to troubleshoot access control lists (ACLs):

- To display the number of Layer 4 CAM entries being used by each ACL, enter the **show access-list <ACL-num> | <ACL-name> | all** command. Refer to [“Displaying ACL information”](#) on page 1746.
- To determine whether the issue is specific to fragmentation, remove the Layer 4 information (TCP or UDP application ports) from the ACL, then reapply the ACL.

If you are using another feature that requires ACLs, either use the same ACL entries for filtering and for the other feature, or change to flow-based ACLs.

Policy-based routing (PBR)

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware. The ACLs classify the traffic. Route maps that match on the ACLs set routing attributes for the traffic.

A PBR policy specifies the next hop for traffic that matches the policy. Using standard ACLs with PBR, you can route IP packets based on their source IP address. With extended ACLs, you can route IP packets based on all of the clauses in the extended ACL.

You can configure the Brocade device to perform the following types of PBR based on a packet Layer 3 and Layer 4 information:

- Select the next-hop gateway.
- Send the packet to the null interface (null0).

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's direct routes or next hops are available, the packet is routed in the normal way.

Configuration considerations for policy-based routing

- PBR is supported in the full Layer 3 code only.
- PBR is not supported on FastIron WS devices.
- PBR is not supported together with ACLs on the same port.
- Global PBR is not supported when IP Follow is configured on an interface.
- Global PBR is not supported with per-port-per-VLAN ACLs.
- A PBR policy on an interface takes precedence over a global PBR policy.
- You cannot apply PBR on a port if that port already has ACLs, ACL-based rate limiting, DSCP-based QoS, MAC address filtering.

- The number of route maps that you can define is limited by the available system memory, which is determined by the system configuration and how much memory other features use. When a route map is used in a PBR policy, the PBR policy uses up to six instances of a route map, up to five ACLs in a matching policy of each route map instance, and up to six next hops in a set policy of each route map instance. Note that the CLI will allow you configure more than six next hops in a route map; however, the extra next hops will not be placed in the PBR database. The route map could be used by other features like BGP or OSPF, which may use more than six next hops.
- ACLs with the **log** option configured should not be used for PBR purposes.
- PBR ignores explicit or implicit **deny ip any any** ACL entries, to ensure that for route maps that use multiple ACLs, the traffic is compared to all the ACLs. PBR also ignores any deny clauses in an ACL. Traffic that matches a deny clause is routed normally using Layer 3 paths.
- PBR always selects the first next hop from the next hop list that is up. If a PBR policy's next hop goes down, the policy uses another next hop if available. If no next hops are available, the device routes the traffic in the normal way.
- PBR is not supported for fragmented packets. If the PBR ACL filters on Layer 4 information like TCP/UDP ports, fragmented packets are routed normally.
- You can change route maps or ACL definitions dynamically and do not need to rebind the PBR policy to an interface.

Configuring a PBR policy

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR globally or on individual interfaces. The device programs the ACLs into the packet processor on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps.

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR.
- Configure a route map that matches on the ACLs and sets the route information.
- Apply the route map to an interface.

Configuring the ACLs

PBR uses route maps to change the routing attributes in IP traffic. This section shows an example of how to configure a standard ACL to identify the source subnet for IP traffic.

To configure a standard ACL to identify a source subnet, enter a command such as the following.

```
Brocade(config)#access-list 99 permit 209.157.23.0 0.0.0.255
```

The command in this example configures a standard ACL that permits traffic from subnet 209.157.23.0/24. After you configure a route map that matches based on this ACL, the software uses the route map to set route attributes for the traffic, thus enforcing PBR.

NOTE

Do not use an access group to apply the ACL to an interface. Instead, use a route map to apply the ACL globally or to individual interfaces for PBR, as shown in the following sections.

Syntax: [no]access-list <num> deny | permit <source-ip> | <hostname> <wildcard>

or

Syntax: `[no]access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname>`

Syntax: `[no]access-list <num> deny | permit host <source-ip> | <hostname>`

Syntax: `[no]access-list <num> deny | permit any`

The `<num>` parameter is the access list number and can be from 1 – 99.

The **deny** | **permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

NOTE

If you are configuring the ACL for use in a route map, always specify **permit**. Otherwise, the Brocade device will ignore deny clauses and packets that match deny clauses are routed normally.

The `<source-ip>` parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE

To specify the host name instead of the IP address, the host name must be configured using the DNS resolver on the Brocade device. To configure the DNS resolver name, use the **ip dns server-address...** command at the global CONFIG level of the CLI.

The `<wildcard>` parameter specifies the mask value to compare against the host address specified by the `<source-ip>` parameter. The `<wildcard>` is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the `<source-ip>`. Ones mean any value matches. For example, the `<source-ip>` and `<wildcard>` values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of “209.157.22.26 0.0.0.255” as “209.157.22.26/24”. The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in “/`<mask-bits>`” format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** `<source-ip> | <hostname>` parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

NOTE

Do not use the **log** option in ACLs that will be used for PBR.

Configuring the route map

After you configure the ACLs, you can configure a PBR route map that matches based on the ACLs and sets routing information in the IP traffic.

NOTE

The match and set statements described in this section are the only route-map statements supported for PBR. Other route-map statements described in the documentation apply only to the protocols with which they are described.

To configure a PBR route map, enter commands such as the following.

```
Brocade(config)#route-map test-route permit 99
Brocade(config-route-map test-route)#match ip address 99
Brocade(config-route-map test-route)#set ip next-hop 192.168.2.1
Brocade(config-route-map test-route)#exit
```

The commands in this example configure an entry in a route map named “test-route”. The **match** statement matches on IP information in ACL 99. The **set** statement changes the next-hop IP address for packets that match to 192.168.2.1.

Syntax: [no]route-map <map-name> permit | deny <num>

The <map-name> is a string of characters that names the map. Map names can be up to 32 characters in length. You can define an unlimited number of route maps on the Brocade device, as long as system memory is available.

The **permit** | **deny** parameter specifies the action the Brocade device will take if a route matches a match statement:

- If you specify **deny**, the Brocade device does not apply a PBR policy to packets that match the ACLs in a match clause. Those packets are routed normally,
- If you specify **permit**, the Brocade device applies the match and set statements associated with this route map instance.

The <num> parameter specifies the instance of the route map you are defining. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

PBR uses up to six route map instances for comparison and ignores the rest.

Syntax: [no] match ip address <ACL-num-or-name>

The <ACL-num> parameter specifies a standard or extended ACL number or name.

Syntax: [no] set ip next hop <ip-addr>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Syntax: [no] set interface null0

This command sends the traffic to the null0 interface, which is the same as dropping the traffic.

Enabling PBR

After you configure the ACLs and route map entries, you can enable PBR globally, on individual interfaces, or both as described in this section. To enable PBR, you apply a route map you have configured for PBR globally or locally.

Enabling PBR globally

To enable PBR globally, enter a command such as the following at the global CONFIG level.

```
Brocade(config)#ip policy route-map test-route
```

This command applies a route map named “test-route” to all interfaces on the device for PBR.

Syntax: `ip policy route-map <map-name>`

Enabling PBR locally

To enable PBR locally, enter commands such as the following.

```
Brocade(config)#interface ve 1
Brocade(config-vif-1)#ip policy route-map test-route
```

The commands in this example change the CLI to the Interface level for virtual interface 1, then apply the “test-route” route map to the interface. You can apply a PBR route map to Ethernet ports or virtual interfaces.

Syntax: `ip policy route-map <map-name>`

Enter the name of the route map you want to use for the route-map <map-name> parameter.

Configuration examples for policy based routing

This section presents configuration examples for configuring and applying a PBR policy.

Basic example of policy based routing

The following commands configure and apply a PBR policy that routes HTTP traffic received on virtual routing interface 1 from the 10.10.10.x/24 network to 5.5.5.x/24 through next-hop IP address 1.1.1.1/24 or, if 1.1.1.x is unavailable, through 2.2.2.1/24.

```
Brocade(config)#access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq http 5.5.5.0
0.0.0.255
Brocade(config)#route-map net10web permit 101
Brocade(config-routemap net10web)#match ip address 101
Brocade(config-routemap net10web)#set ip next-hop 1.1.1.1
Brocade(config-routemap net10web)#set ip next-hop 2.2.2.2
Brocade(config-routemap net10web)#exit
Brocade(config)#vlan 10
Brocade(config-vlan-10)#tagged ethernet 1/1 to 1/4

Brocade(config-vlan-10)#router-interface ve 1
Brocade(config)#interface ve 1
Brocade(config-vif-1)#ip policy route-map net10web
```

Syntax: `[no] route-map <map-name> permit | deny <num>`

Syntax: `[no] set ip next hop <ip-addr>`

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Setting the next hop

The following commands configure the Brocade device to apply PBR to traffic from IP subnets 209.157.23.x, 209.157.24.x, and 209.157.25.x. In this example, route maps specify the next-hop gateway for packets from each of these subnets:

- Packets from 209.157.23.x are sent to 192.168.2.1.
- Packets from 209.157.24.x are sent to 192.168.2.2.
- Packets from 209.157.25.x are sent to 192.168.2.3.

The following commands configure three standard ACLs. Each ACL contains one of the ACLs listed above. Make sure you specify **permit** instead of deny in the ACLs, so that the Brocade device permits the traffic that matches the ACLs to be further evaluated by the route map. If you specify **deny**, the traffic that matches the deny statements are routed normally. Notice that these ACLs specify **any** for the destination address.

```
Brocade(config)#access-list 50 permit 209.157.23.0 0.0.0.255
Brocade(config)#access-list 51 permit 209.157.24.0 0.0.0.255
Brocade(config)#access-list 52 permit 209.157.25.0 0.0.0.255
```

The following commands configure three entries in a route map called “test-route”. The first entry (permit 50) matches on the IP address information in ACL 50 above. For IP traffic from subnet 209.157.23.0/24, this route map entry sets the next-hop IP address to 192.168.2.1.

```
Brocade(config)#route-map test-route permit 50
Brocade(config-route-map test-route)#match ip address 50
Brocade(config-route-map test-route)#set ip next-hop 192.168.2.1
Brocade(config-route-map test-route)#exit
```

The following commands configure the second entry in the route map. This entry (permit 51) matches on the IP address information in ACL 51 above. For IP traffic from subnet 209.157.24.0/24, this route map entry sets the next-hop IP address to 192.168.2.2.

```
Brocade(config)#route-map test-route permit 51
Brocade(config-route-map test-route)#match ip address 51
Brocade(config-route-map test-route)#set ip next-hop 192.168.2.2
Brocade(config-route-map test-route)#exit
```

The following commands configure the third entry in the test-route route map. This entry (permit 52) matches on the IP address information in ACL 52 above. For IP traffic from subnet 209.157.25.0/24, this route map entry sets the next-hop IP address to 192.168.2.3.

```
Brocade(config)#route-map test-route permit 52
Brocade(config-route-map test-route)#match ip address 52
Brocade(config-route-map test-route)#set ip next-hop 192.168.2.3
Brocade(config-route-map test-route)#exit
```

The following command enables PBR by globally applying the test-route route map to all interfaces.

```
Brocade(config)#ip policy route-map test-route
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the three source subnets identified in ACLs 50, 51, and 52, then apply route map test-route to the interface.

```
Brocade(config)#interface ve 1
Brocade(config-vif-1)#ip address 209.157.23.1/24
Brocade(config-vif-1)#ip address 209.157.24.1/24
Brocade(config-vif-1)#ip address 209.157.25.1/24
Brocade(config-vif-1)#ip policy route-map test-route
```

Setting the output interface to the null interface

The following commands configure a PBR policy to send all traffic from 192.168.1.204/32 to the null interface, thus dropping the traffic instead of forwarding it.

```
Brocade(config)#access-list 56 permit 209.168.1.204 0.0.0.0
```

The following commands configure an entry in a route map called “file-13”. The first entry (permit 56) matches on the IP address information in ACL 56 above. For IP traffic from the host 209.168.1.204/32, this route map entry sends the traffic to the null interface instead of forwarding it, thus sparing the rest of the network the unwanted traffic.

```
Brocade(config)#route-map file-13 permit 56
Brocade(config-routemap file-13)#match ip address 56
Brocade(config-routemap file-13)#set interface null0
Brocade(config-routemap file-13)#exit
```

The following command enables PBR by globally applying the route map to all interfaces.

```
Brocade(config)#ip policy route-map file-13
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the source subnet identified in ACL 56, then apply route map file-13 to the interface.

```
Brocade(config)#interface ethernet 3/11
Brocade(config-if-e10000-3/11)#ip address 192.168.1.204/32
Brocade(config-if-e10000-3/11)#ip policy route-map file-13
```

Trunk formation with PBR policy

When a trunk is formed, the PBR policy on the primary port applies to all the secondary ports. If a different PBR policy exists on a secondary port at the time of a trunk formation, that policy is overridden by the PBR policy on the primary port. If the primary port does not have a PBR policy, then the secondary ports will not have a PBR policy.

When a trunk is removed, the PBR policy that was applied to the trunk interface is unbound (removed) from former secondary ports. If global PBR is configured, the secondary ports adhere to the global PBR; otherwise, no PBR policy is bound to former secondary ports.

IPv6 ACLs

[Table 286](#) lists the individual Brocade FastIron switches and the IPv6 Access Control Lists (ACL) features they support. These features are supported in Brocade FastIron switches that can be configured as an IPv6 host in an IPv6 network, and in devices that support IPv6 routing. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 286 Supported IPv6 ACL features

Feature	FESX FSX 800 FSX 1600 ¹	FWS	FCX ¹	ICX 6610	ICX 6430 ICX 6450
IPv6 ACLs	Yes	No	Yes	Yes	Yes
Applying an IPv6 ACL to an interface	Yes	No	Yes	Yes	Yes
IPv6 ACL comment text	Yes	No	Yes	Yes	Yes
IPv6 ACL logging of denied packets	Yes	No	Yes	Yes	Yes

1. IPv6 ACLs are not supported on base Layer 3 software images on the FSX and FCX platforms

This chapter describes how Access Control Lists (ACLs) are implemented and configured on a Brocade FastIron IPv6 switch.

IPv6 ACL overview

Brocade devices support IPv6 Access Control Lists (ACLs) for inbound traffic filtering, as detailed in [Table 286](#). You can configure up to 100 IPv6 ACLs and, by default, up to a system-wide maximum of 4000 ACL rules. For example, you can configure one ACL with 4000 entries, two ACLs with 2000 and 2093 entries respectively (combining IPv4 and IPv6 ACLs), etc.

An IPv6 ACL is composed of one or more conditional statements that pose an action (permit or deny) if a packet matches a specified source or destination prefix. For FESX and FSX devices, there can be up to 1024 statements per port region, including IPv6, IPv4, MAC address filters, and default statements. For FCX devices, there can be up to 4096 statements per port region, including IPv6, IPv4, MAC address filters, and default statements. For ICX devices, there can be up to 1536 statements per port region, including IPv6, IPv4, MAC address filters, and default statements. When the maximum number of ACL rules allowed per port region is reached, an error message will display on the console.

In ACLs with multiple statements, you can specify a priority for each statement. The specified priority determines the order in which the statement appears in the ACL. The last statement in each IPv6 ACL is an implicit deny statement for all packets that do not match the previous statements in the ACL.

You can configure an IPv6 ACL on a global basis, then apply it to the incoming IPv6 packets on specified interfaces. You can apply only one IPv6 ACL to an interface. When an interface receives an IPv6 packet, it applies the statements within the ACL in their order of appearance to the packet. As soon as a match occurs, the Brocade device takes the specified action (permit or deny the packet) and stops further comparison for that packet.

IPv6 ACLs are supported on:

- Gbps Ethernet ports
- 10 Gbps Ethernet ports
- Trunk groups
- Virtual routing interfaces

NOTE

IPv6 ACLs are supported on inbound traffic and are implemented in hardware, making it possible for the Brocade device to filter traffic at line-rate speed on 10 Gbps interfaces.

IPv6 ACL traffic filtering criteria

The Brocade implementation of IPv6 ACLs enable traffic filtering based on the following information:

- IPv6 protocol
- Source IPv6 address
- Destination IPv6 address
- IPv6 message type
- Source TCP or UDP port (if the IPv6 protocol is TCP or UDP)
- Destination TCP or UDP port (if the IPv6 protocol is TCP or UDP)

IPv6 protocol names and numbers

The IPv6 protocol can be one of the following well-known names or any IPv6 protocol number from 0 – 255:

- Authentication Header (AHP)
- Encapsulating Security Payload (ESP)
- Internet Control Message Protocol (ICMP)
- Internet Protocol Version 6 (IPv6)
- Stream Control Transmission Protocol (SCTP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

NOTE

TCP and UDP filters will be matched only if they are listed as the first option in the extension header.

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IPv6 address to the website IPv6 address.

IPv6 ACLs also provide support for filtering packets based on DSCP.

IPv6 ACL configuration notes

- IPv4 ACLs that filter based on VLAN membership or VE port membership (ACL-per-port-per-VLAN), are supported together with IPv6 ACLs on the same device, as long as they are not bound to the same port or virtual interface.
- IPv4 source guard and IPv6 ACLs are supported together on the same device, as long as they are not configured on the same port or virtual interface.
- IPv6 ACLs do not support ACL filtering based on VLAN membership or VE port membership.
- IPv6 ACLs cannot be used with GRE
- IPv6 ACLs cannot be employed to implement a user-based ACL scheme
- If an IPv6 ACL has the implicit **deny** condition, make sure it also **permits** the IPv6 link-local address, in addition to the global unicast address. Otherwise, routing protocols such as OSPF will not work. To view the link-local address, use the **show ipv6 interface** command.
- IPv6 must be enabled on the interface before an ACL can be applied to it. If IPv6 is not enabled on the interface, the system will display the following error message.

```
Brocade(config-if-e1000-7)#ipv6 traffic-filter netw in
Error: IPv6 is not enabled for interface 7
```

To enable IPv6 on an interface, enter **ipv6 enable** at the Interface level of the CLI, or assign an IPv6 address to the interface as described in [“IPv6 configuration on each router interface”](#) on page 362 and further discussed in [“Enabling IPv6 on an interface to which an ACL will be applied”](#) on page 1768.

- You cannot disable IPv6 on an interface to which an ACL is bound. Attempting to do so will cause the system to return the following error message.

```
Brocade(config-if-e1000-7)#no ipv6 enable
Error: Port 7 has IPv6 ACL configured. Cannot disable IPv6
```

To disable IPv6, first remove the ACL from the interface.

- For notes on applying IPv6 ACLs to trunk ports, see [“Applying an IPv6 ACL to a trunk group”](#) on page 1769.
- For notes on applying IPv6 ACLs to virtual ports, see [“Applying an IPv6 ACL to a virtual interface in a protocol-based or subnet-based VLAN”](#) on page 1769.
- The **dscp-cos-mapping** option is supported on FESX and FSX devices only.

Configuring an IPv6 ACL

Follow the steps given below to configure an IPv6 ACL.

1. Create the ACL.
2. Enable IPv6 on the interface to which the ACL will be applied.
3. Apply the ACL to the interface.

Example IPv6 configurations

To configure an access list that blocks all Telnet traffic received on port 1/1 from IPv6 host 2000:2382:e0bb::2, enter the following commands.

```
Brocade(config)# ipv6 access-list fdry
Brocade(config-ipv6-access-list-fdry)# deny tcp host 2000:2382:e0bb::2 any eq
telnet
Brocade(config-ipv6-access-list-fdry)# permit ipv6 any any
Brocade(config-ipv6-access-list-fdry)# exit
Brocade(config)# int eth 1/1
Brocade(config-if-1/1)# ipv6 enable
Brocade(config-if-1/1)# ipv6 traffic-filter fdry in
Brocade(config)# write memory
```

The following is another example of commands for configuring an ACL and applying it to an interface.

```
Brocade(config)# ipv6 access-list netw
Brocade(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
Brocade(config-ipv6-access-list-netw)# deny ipv6 host 2000:2383:e0ac::2 host
2000:2383:e0aa:0::24
Brocade(config-ipv6-access-list-netw)# deny udp any any
Brocade(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first condition permits ICMP traffic from hosts in the 2000:2383:e0bb::x network to hosts in the 2001:3782::x network.

The second condition denies all IPv6 traffic from host 2000:2383:e0ac::2 to host 2000:2383:e0aa:0::24.

The third condition denies all UDP traffic.

The fourth condition permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming IPv6 traffic on the ports to which you assigned the ACL.

The following commands apply the ACL "netw" to the incoming traffic on port 1/2 and to the incoming traffic on port 4/3.

```
Brocade(config)# int eth 1/2
Brocade(config-if-1/2)# ipv6 enable
Brocade(config-if-1/2)# ipv6 traffic-filter netw in
Brocade(config-if-1/2)# exit
Brocade(config)# int eth 4/3
Brocade(config-if-4/3)# ipv6 enable
Brocade(config-if-4/3)# ipv6 traffic-filter netw in
Brocade(config)# write memory
```

Here is another example.

```
Brocade(config)# ipv6 access-list nextone
Brocade(config-ipv6-access-list rtr)# deny tcp 2001:1570:21::/24
2001:1570:22::/24
Brocade(config-ipv6-access-list rtr)# deny udp any range 5 6 2001:1570:22::/24
Brocade(config-ipv6-access-list rtr)# permit ipv6 any any
Brocade(config-ipv6-access-list rtr)# write memory
```

The first condition in this ACL denies TCP traffic from the 2001:1570:21::x network to the 2001:1570:22::x network.

The next condition denies UDP packets from any source with source UDP port in ranges 5 to 6 and whose destination is to the 2001:1570:22::/24 network.

The third condition permits all packets containing source and destination addresses that are not explicitly denied by the first two. Without this entry, the ACL would deny all incoming IPv6 traffic on the ports to which you assign the ACL.

A **show running-config** command displays the following.

```
Brocade(config)# show running-config
ipv6 access-list rtr
deny tcp 2001:1570:21::/24 2001:1570:22::/24
deny udp any range rje 6 2001:1570:22::/24
permit ipv6 any any
```

A **show ipv6 access-list** command displays the following.

```
Brocade(config)# sh ipv6 access-list rtr
ipv6 access-list rtr: 3 entries
10: deny tcp 2001:1570:21::/24 2001:1570:22::/24
20: deny udp any range rje 6 2001:1570:22::/24
30: permit ipv6 any any
```

The following commands apply the ACL “rtr” to the incoming traffic on ports 2/1 and 2/2.

```
Brocade(config)# int eth 2/1
Brocade(config-if-2/1)# ipv6 enable
Brocade(config-if-2/1)# ipv6 traffic-filter rtr in
Brocade(config-if-2/1)# exit
Brocade(config)# int eth 2/2
Brocade(config-if-2/2)# ipv6 enable
Brocade(config-if-2/2)# ipv6 traffic-filter rtr in
Brocade(config)# write memory
```

Default and implicit IPv6 ACL action

The default action when no IPv6 ACLs are configured on an interface is to permit all IPv6 traffic. However, once you configure an IPv6 ACL and apply it to an interface, the default action for that interface is to deny all IPv6 traffic that is not explicitly permitted on the interface.

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.

- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The permit entry permits packets that are not denied by the deny entries.

Every IPv6 ACL has the following implicit conditions as its last match conditions.

- **permit icmp any any nd-na** – Allows ICMP neighbor discovery acknowledgements.
- **permit icmp any any nd-ns** – Allows ICMP neighbor discovery solicitations.
- **deny ipv6 any any** – Denies IPv6 traffic. You must enter a **permit ipv6 any any** as the last statement in the access-list if you want to permit IPv6 traffic that were not denied by the previous statements.

NOTE

If an IPv6 ACL has the implicit **deny** condition, make sure it also **permits** the IPv6 link-local address, in addition to the global unicast address. Otherwise, routing protocols such as OSPF will not work. To view the link-local address, use the **show ipv6 interface** command.

The conditions are applied in the order shown above, with **deny ipv6 any any** as the last condition applied.

For example, if you want to deny ICMP neighbor discovery acknowledgement, then permit any remaining IPv6 traffic, enter commands such as the following.

```
Brocade(config)# ipv6 access-list netw
Brocade(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
Brocade(config-ipv6-access-list-netw)# deny icmp any any nd-na
Brocade(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first permit statement permits ICMP traffic from hosts in the 2000:2383:e0bb::x network to hosts in the 2001:3782::x network.

The deny statement denies ICMP neighbor discovery acknowledgement.

The last entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL will deny all incoming IPv6 traffic on the ports to which you assigned the ACL.

Furthermore, if you add the statement **deny icmp any any** in the access list, then all neighbor discovery messages will be denied. You must explicitly enter the **permit icmp any any nd-na** and **permit icmp any any nd-ns** statements just before the **deny icmp** statement if you want the ACLs to permit neighbor discovery as in the example below.

```
Brocade(config)#ipv6 access-list netw
Brocade(config-ipv6-access-list-netw)#permit icmp 2000:2383:e0bb::/64
2001:3782::/64
Brocade(config-ipv6-access-list-netw)#permit icmp any any nd-na
Brocade(config-ipv6-access-list-netw)#permit icmp any any nd-ns
Brocade(config-ipv6-access-list-netw)#deny icmp any any
Brocade(config-ipv6-access-list-netw)#permit ipv6 any any
```

Creating an IPv6 ACL

Before an IPv6 ACL can be applied to an interface, it must first be created, and then IPv6 must be enabled on that interface.

To create an IPv6 ACL, enter commands such as the following:

```
Brocade(config)# ipv6 access-list fdry
Brocade(config-ipv6-access-list-fdry)# deny tcp host 2000:2382:e0bb::2 any eq
telnet
Brocade(config-ipv6-access-list-fdry)# permit ipv6 any any
Brocade(config-ipv6-access-list-fdry)# exit
```

This creates an access list that blocks all Telnet traffic from IPv6 host 2000:2382:e0bb::2.

Syntax for creating an IPv6 ACL

NOTE

The following features are not supported:

- **ipv6-operator flow-label**
- **ipv6-operator fragments** when any protocol is specified. The option "fragments" can be specified only when "permit/deny ipv6" is specified. If you specify "tcp" or any other protocol instead of "ipv6" the keyword, "fragments" cannot be used.
- **ipv6-operator routing** when any protocol is specified. (Same limitation as for **ipv6-operator fragments**)

When creating ACLs, use the appropriate syntax below for the protocol you are filtering.

For IPv6 and supported protocols other than ICMP, TCP, or UDP

Syntax: [no] ipv6 access-list <ACL name>

Syntax: permit | deny <protocol>
 <ipv6-source-prefix/prefix-length> | any | host <source-ipv6_address>
 <ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address>
 [ipv6-operator [<value>]]
 [802.1p-priority-marking <number>]
 [dscp-marking <number> 802.1p-priority-marking <number> internal-priority-marking
 <number>] | [dscp-marking <dscp-value> dscp-cos-mapping] | [dscp-cos-mapping]

For ICMP

Syntax: [no] ipv6 access-list <ACL name>

Syntax: permit | deny icmp <ipv6-source-prefix/prefix-length> | any | host
 <source-ipv6_address>
 <ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address>
 [ipv6-operator [<value>]]
 [[<icmp-type>][<icmp-code>]] | [<icmp-message>]
 [dscp-marking <number>]

```
[dscp-marking <dscp-value> dscp-cos-mapping]
[dscp-cos-mapping]
```

For TCP

Syntax: [no] ipv6 access-list <ACL name>

Syntax: permit | deny <tcp>
<ipv6-source-prefix/prefix-length> | any | host <source-ipv6_address> [tcp-udp-operator
[source-port-number]]
<ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address>
[tcp-udp-operator [destination-port-number]]
[ipv6-operator [<value>]]
[802.1p-priority-matching <number>]
[dscp-marking <number> 802.1p-priority-marking <number> internal-priority-marking
<number>]
[dscp-marking <dscp-value> dscp-cos-mapping]
[dscp-cos-mapping]

For UDP

Syntax: [no] ipv6 access-list <ACL name>

Syntax: permit | deny <udp>
<ipv6-source-prefix/prefix-length> | any | host <source-ipv6_address> [tcp-udp-operator
[source port number]]
<ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address>
[tcp-udp-operator [destination port number]]
[ipv6-operator [<value>]]
[802.1p-priority-matching <number>]
[dscp-marking <number> 802.1p-priority-marking <number> internal-priority-marking
<number>]
[dscp-marking <dscp-value> dscp-cos-mapping]
[dscp-cos-mapping]

TABLE 287 Syntax descriptions

IPv6 ACL arguments	Description
ipv6 access-list <ACL name>	Enables the IPv6 configuration level and defines the name of the IPv6 ACL. The <ACL name> can contain up to 199 characters and numbers, but cannot begin with a number and cannot contain any spaces or quotation marks.
permit	The ACL will permit (forward) packets that match a policy in the access list.
deny	The ACL will deny (drop) packets that match a policy in the access list.
icmp	Indicates the you are filtering ICMP packets.
protocol	The type of IPv6 packet you are filtering. You can specify a well-known name for some protocols whose number is less than 255. For other protocols, you must enter the number. Enter “?” instead of a protocol to list the well-known names recognized by the CLI. IPv6 protocols include AHP – Authentication Header ESP – Encapsulating Security Payload IPv6 – Internet Protocol version 6 SCTP – Stream Control Transmission Protocol
<ipv6-source-prefix>/<prefix-length>	The <ipv6-source-prefix>/<prefix-length> parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-source-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.
<ipv6-destination-prefix>/<prefix-length>	The <ipv6-destination-prefix>/<prefix-length> parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-destination-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter
any	When specified instead of the <ipv6-source-prefix>/<prefix-length> or <ipv6-destination-prefix>/<prefix-length> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.
icmp-type	ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.
icmp code	ICMP packets, which are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255,
icmp-message	ICMP packets are filtered by ICMP messages. Refer to “ICMP message configurations” on page 1766 for a list of ICMP message types.
tcp	Indicates the you are filtering TCP packets.
udp	Indicates the you are filtering UDP packets.

TABLE 287 Syntax descriptions (Continued)

IPv6 ACL arguments	Description
<code><ipv6-source-prefix>/<prefix-length></code>	The <code><ipv6-source-prefix>/<prefix-length></code> parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <code><ipv6-source-prefix></code> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <code><prefix-length></code> parameter as a decimal value. A slash mark (/) must follow the <code><ipv6-prefix></code> parameter and precede the <code><prefix-length></code> parameter.
<code><ipv6-destination-prefix>/<prefix-length></code>	The <code><ipv6-destination-prefix>/<prefix-length></code> parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <code><ipv6-destination-prefix></code> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <code><prefix-length></code> parameter as a decimal value. A slash mark (/) must follow the <code><ipv6-prefix></code> parameter and precede the <code><prefix-length></code> parameter.
any	When specified instead of the <code><ipv6-source-prefix>/<prefix-length></code> or <code><ipv6-destination-prefix>/<prefix-length></code> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all 128 is implied.
tcp-udp-operator	<p>The <code><tcp-udp-operator></code> parameter can be one of the following:</p> <ul style="list-style-type: none"> • eq – The policy applies to the TCP or UDP port name or number you enter after eq. • gt – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after gt. Enter "?" to list the port names. • lt – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after lt. • neq – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after neq. • range – The policy applies to all TCP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the range parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following range 23 53. The first port number in the range must be lower than the last number in the range. <p>The <code><source-port number></code> and <code><destination-port number></code> for the tcp-udp-operator is the number of the port.</p>

TABLE 287 Syntax descriptions (Continued)

IPv6 ACL arguments	Description
ipv6-operator	<p>Allows you to filter the packets further by using one of the following options:</p> <ul style="list-style-type: none"> • dscp – The policy applies to packets that match the traffic class value in the traffic class field of the IPv6 packet header. This operator allows you to filter traffic based on TOS or IP precedence. You can specify a value from 0 – 63. • fragments – The policy applies to fragmented packets that contain a non-zero fragment offset. <p>NOTE: This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags.</p> <ul style="list-style-type: none"> • routing – The policy applies only to IPv6 source-routed packets. <p>NOTE This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags.</p>
802.1p-priority-matching <i><number></i>	<p>Enables the device to match only those packets that have the same 802.1p priorities as specified in the ACL. Enter 0 – 7.</p> <p>Use this option in conjunction with traffic policies to rate limit traffic for a specified 802.1p priority value. For details, refer to “Inspecting the 802.1p bit in the ACL for adaptive rate limiting” on page 1781.</p>
dscp-marking <i><number></i>	<p>Use the dscp-marking <i><number></i> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the DSCP value that you specify to the packet. Enter 0 – 63.</p>
802.1p-priority-marking <i><number></i>	<p>Use the 802.1p-priority-marking <i><number></i> parameter to specify a new QoS value to the packet (0-7). If a packet matches the filters in the ACL statement, the following actions happen:</p> <ul style="list-style-type: none"> • On FSX devices, this parameter assigns the 802.1p priority that you specify to the packet. • On all platforms other than FSX, this parameter assigns the priority that you specify to the 802.1p priority and the internal priority.
internal-priority-marking <i><number></i>	<p>Use the internal-priority-marking <i><number></i> parameter to specify a new QoS value to the packet (0-7). If a packet matches the filters in the ACL statement, the following actions happen:</p> <ul style="list-style-type: none"> • On FSX devices, this parameter assigns the internal priority that you specify to the packet. • On all platforms other than FSX, this parameter assigns the priority that you specify to the internal priority and the 802.1p priority. <p>NOTE: On all platforms other than FSX, configuring 802.1p-priority-marking alone or configuring both 802.1p-priority-marking and internal-priority-marking has the same functionality. That is, it assigns the priority that you specify to the 802.1p priority and the internal priority.</p>

TABLE 287 Syntax descriptions (Continued)

IPv6 ACL arguments	Description
dscp-marking <number>	Use the dscp-marking <number> dscp-cos-mapping parameters parameters to specify a DSCP value and map that value to an internal QoS table to obtain the packet new QoS value. The following occurs when you use these parameters. <ul style="list-style-type: none"> You enter 0 – 63 for the dscp-marking <number> parameter. The dscp-cos-mapping parameter takes the DSCP value you specified and compares it to an internal QoS table, which is indexed by DSCP values. The corresponding 802.1p priority, internal forwarding priority, and DSCP value is assigned to the packet.
dscp-cos-mapping ¹	Use dscp-cos-mapping if you want to use the DSCP value in the packet header to alter its QoS value. When you enter dscp-cos-mapping , the DSCP value in the packet header is compared to a column in the internal QoS table. The 802.1p priority, internal forwarding priority, and DSCP value that are mapped to the matching column is assigned to the packet.

1. **dscp-cos-mapping** is supported on FESX and FSX devices only.

ICMP message configurations

If you want to specify an ICMP message, you can enter one of the following ICMP message types:

- beyond-scope
- destination-unreachable
- echo-reply
- echo-request
- header
- hop-limit
- mld-query
- mld-reduction
- mld-report
- nd-na
- nd-ns
- next-header
- no-admin
- no-route
- packet-too-big
- parameter-option
- parameter-problem
- port-unreachable
- reassembly-timeout
- renum-command
- renum-result
- renum-seq-number
- router-advertisement

- router-renumbering
- router-solicitation
- time-exceeded
- unreachable

NOTE

If you do not specify a message type, the ACL applies to all types ICMP messages types.

Enabling IPv6 on an interface to which an ACL will be applied

Before an IPv6 ACL can be applied to an interface, it must first be created, and then IPv6 must be enabled on that interface.

To enable IPv6 on an interface, enter **ipv6 enable** at the Interface level of the CLI, or assign an IPv6 address to the interface, as described in [“IPv6 configuration on each router interface”](#) on page 362.

For example:

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-1/1)#ipv6 enable
```

These commands enable IPv6 on Ethernet interface 1/1 ready for an IPv6 ACL to be applied.

Syntax for enabling IPv6 on an interface

Syntax: **ipv6 enable**

When issued at the Interface Configuration level, this command enables IPv6 for a specific interface.

Applying an IPv6 ACL to an interface

As mentioned in [“IPv6 ACL overview”](#) on page 1755, IPv6 ACLs are supported on the following devices:

- Gbps Ethernet ports
- 10 Gbps Ethernet ports
- Trunk groups
- Virtual routing interfaces

To apply an IPv6 ACL to an interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e100-3/1)#ipv6 traffic-filter access1 in
```

This example applies the IPv6 ACL “access1” to incoming IPv6 packets on Ethernet interface 3/1. As a result, Ethernet interface 3/1 denies all incoming packets from the site-local prefix fec0:0:0:2::/64 and the global prefix 2001:100:1::/48 and permits all other incoming packets.

Syntax for applying an IPv6 ACL

NOTE

The **ipv6 traffic-filter** *<ipv6-ACL-name>* **in** command is supported on FCX, ICX 6610, ICX 6430, ICX 6450, and FESX devices only. The command is not supported on FSX, FLS, FGS, and FWS devices.

Syntax: **ipv6 traffic-filter** *<ipv6-ACL-name>* **in**

For the *<ipv6-ACL-name>* parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

The **in** keyword applies the specified IPv6 ACL to incoming IPv6 packets on the interface.

Applying an IPv6 ACL to a trunk group

When applying an IPv6 ACL to a trunk group, apply it to the primary port of the trunk, as described under [“Applying an IPv6 ACL to an interface”](#) on page 1768. IPv6 ACLs cannot be applied to secondary ports. When an IPv6 ACL is applied to a primary port in a trunk, it filters the traffic on the secondary ports of the trunk as well as the traffic on the primary port.

Applying an IPv6 ACL to a virtual interface in a protocol-based or subnet-based VLAN

As with IPv4 ACLs, by default, when you apply an IPv6 ACL to a virtual interface in a protocol-based or subnet-based VLAN, the ACL takes effect on all protocol or subnet VLANs to which the untagged port belongs. To prevent the Brocade device from denying packets on other virtual interfaces that do not have an ACL applied, configure an ACL that permits packets in the IP subnet of the virtual interface in all protocol-based or subnet-based VLANs to which the untagged port belongs.

Adding a comment to an IPv6 ACL entry

You can optionally add a comment to describe entries in an IPv6 ACL. The comment appears in the output of **show** commands that display ACL information.

You can add a comment by entering the **remark** command immediately preceding an ACL entry. For example, to enter comments preceding an ACL entry, enter commands such as the following.

```
Brocade(config)#ipv6 access-list rtr
Brocade(config-ipv6-access-list rtr)# remark This entry permits ipv6 packets from
3002::2 to any destination
Brocade(config-ipv6-access-list rtr)# permit ipv6 host 3000::2 any
Brocade(config-ipv6-access-list rtr)# remark This entry denies udp packets from
any source to any destination
Brocade(config-ipv6-access-list rtr)# deny udp any any
Brocade(config-ipv6-access-list rtr)# remark This entry denies IPv6 packets from
any source to any destination
Brocade(config-ipv6-access-list rtr)# deny ipv6 any any
Brocade(config-ipv6-access-list rtr)# write memory
```

Syntax: **remark** *<comment-text>*

The *<comment-text>* can be up to 256 characters in length.

Deleting a comment from an IPv6 ACL entry

The following shows the comment text for the ACL named "rtr" in a **show running-config** display.

```
Brocade#show running-config
ipv6 access-list rtr
  remark This entry permits ipv6 packets from 3002::2 to any destination
  permit ipv6 host 3000::2 any
  remark This entry denies udp packets from any source to any destination
  deny udp any any
  remark This entry denies IPv6 packets from any source to any destination
  deny ipv6 any any
```

Syntax: **show running-config**

Deleting a comment from an IPv6 ACL entry

To delete a comment from an IPv6 ACL entry, enter commands such as the following.

```
Brocade(config)#ipv6 access-list rtr
Brocade(config-ipv6-access-list rtr)#no remark This entry permits ipv6 packets
from 3002::2 to any destination
```

Syntax: **no remark** *<comment-text>*

For *<comment-text>*, enter the text exactly as you did when you created the comment.

Support for ACL logging

Brocade devices support ACL logging of inbound packets that are sent to the CPU for processing (denied packets). ACL logging is not supported for any packets that are processed in hardware (permitted packets).

You may want the software to log entries in the Syslog for packets that are denied by ACL filters. ACL logging is disabled by default; it must be explicitly enabled on a port. Refer to [“ACL logging”](#) on page 1729.

Displaying IPv6 ACLs

To display the IPv6 ACLs configured on a device, enter the **show ipv6 access-list** command. Here is an example.


```

Brocade#show ipv6 access-list
ipv6 access-list v6-ACL1: 1 entries
  deny ipv6 any any
ipv6 access-list v6-ACL2: 1 entries
  permit ipv6 any any
ipv6 access-list v6-ACL3: 2 entries
  deny ipv6 2001:aa:10::/64 any
  permit ipv6 any any
ipv6 access-list v6-ACL4: 2 entries
  deny ipv6 2002:aa::/64 any
  permit ipv6 any any
ipv6 access-list rate-ACL: 1 entries
  permit ipv6 any any traffic-policy rate800M
ipv6 access-list v6-ACL5: 8 entries
  permit tcp 2002:bb::/64 any
  permit ipv6 2002:bb::/64 any
  permit ipv6 2001:aa:101::/64 any
  permit ipv6 2001:aa:10::/64 2001:aa:102::/64
  permit ipv6 host 2001:aa:10::102 host 2001:aa:101::102
  permit ipv6 host 2001:aa:10::101 host 2001:aa:101::101 dscp-matching 0
  dscp-marking 63 dscp-cos-mapping
  permit ipv6 any any dscp-matching 63 dscp-cos-mapping
  permit ipv6 any any fragments

```

Syntax: show ipv6 access-list

To display a specific IPv6 ACL configured on a device, enter the **show ipv6 access-list** command followed by the ACL name. The following example shows the ACL named "rtr".

```

Brocade#show ipv6 access-list rtr
ipv6 access-list rtr: 3 entries
  remark This entry permits ipv6 packets from 3002::2 to any destination
  permit ipv6 host 3000::2 any
  remark This entry denies udp packets from any source to any destination
  deny udp any any
  remark This entry denies IPv6 packets from any source to any destination
  deny ipv6 any any

```

Syntax: show ipv6 access-list [<access-list-name>]

For the <access-list-name> parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

Displaying IPv6 ACLs

Traffic Policies

Table 288 lists the individual Brocade FastIron switches and the traffic policy features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 288 Supported traffic policy features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Traffic policies	Yes	Yes	Yes	Yes	Yes
ACL-based fixed rate limiting	Yes	Yes	Yes	Yes	Yes
ACL-based adaptive rate limiting	Yes	No	Yes	Yes	Yes
802.1p priority bit inspection in the ACL for adaptive rate limiting	Yes (IPv6 devices only)	No	Yes	Yes	Yes
ACL statistics	Yes	Yes	Yes	Yes	Yes. ICX 6430 does not support creating a policy that only counts packets.
CPU rate-limiting	Yes	Yes	Yes	Yes	Yes

Traffic policies overview

This chapter describes how traffic policies are implemented and configured on the FastIron devices.

Brocade devices use traffic policies for the following reasons:

- To rate limit inbound traffic
- To count the packets and bytes per packet to which ACL permit or deny clauses are applied

Traffic policies consist of policy names and policy definitions:

- Traffic policy name – A string of up to eight alphanumeric characters that identifies individual traffic policy definitions.
- Traffic policy definition (TPD) – The command filter associated with a traffic policy name. A TPD can define any one of the following:
 - Rate limiting policy

- ACL counting policy
- Combined rate limiting and ACL counting policy

The maximum number of supported active TPDs is a system-wide parameter and depends on the device you are configuring. The total number of active TPDs cannot exceed the system maximum. Refer to [“Maximum number of traffic policies supported on a device”](#) on page 1775.

When you apply a traffic policy to an interface, you do so by adding a reference to the traffic policy in an ACL entry, instead of applying the individual traffic policy to the interface. The traffic policy becomes an active traffic policy or active TPD when you bind its associated ACL to an interface.

To configure traffic policies for ACL-based rate limiting, refer to [“Configuring ACL-based fixed rate limiting”](#) on page 1777 and [“ACL-based adaptive rate limiting configuration”](#) on page 1778.

To configure traffic policies for ACL counting, refer to [“Enabling ACL statistics”](#) on page 1783.

Configuration notes and feature limitations for traffic policies

Note the following when configuring traffic policies:

- Traffic policies applies to IP ACLs only.
- Traffic policies are supported on FastIron X Series devices, but not on the 10 Gbps Ethernet interfaces of the SX-FI62XG and SX-FI42XG modules.
- The maximum number of supported active TPDs is a system-wide parameter and depends on the device you are configuring. The total number of active TPDs cannot exceed the system maximum. Refer to [“Maximum number of traffic policies supported on a device”](#) on page 1775.
- You can reference the same traffic policy in more than one ACL entry within an ACL. For example, two or more ACL statements in ACL 101 can reference a TPD named TPD1.
- You can reference the same traffic policy in more than one ACL. For example, ACLs 101 and 102 could both reference a TPD named TPD1.
- To modify or delete an active traffic policy, you must first unbind the ACL that references the traffic policy.
- When you define a TPD (when you enter the CLI command **traffic-policy**), explicit marking of CoS parameters, such as traffic class and 802.1p priority, are not available on the device. In the case of a TPD defining rate limiting, the device re-marks CoS parameters based on the DSCP value in the packet header and the determined conformance level of the rate limited traffic, as shown in [Table 289](#).

TABLE 289 CoS parameters for packets that use rate limiting traffic policies

Packet conformance level	Packet DSCP value	Traffic class and 802.1p priority
0 (Green) or 1 (Yellow)	0 – 7	0 (lowest priority queue)
	8 – 15	1
	16 – 23	2
	24 – 31	3
	32 – 39	4
	40 – 47	5
	48 – 55	6
	56 – 63	7 (highest priority queue)
2 (Red)	N/A	0 (lowest priority queue)

- When you define a TPD, reference the TPD in an ACL entry, and then apply the ACL to a VE in the Layer 3 router code, the rate limit policy is accumulative for all of the ports in the port region. If the VE or VLAN contains ports that are in different port regions, the rate limit policy is applied per port region.

For example, TPD1 has a rate limit policy of 600M and is referenced in ACL 101. ACL 101 is applied to VE 1, which contains ports e 1/11 to e 1/14. Because ports e 1/11 and 1/12 are in a different port region than e 1/13 and 1/14, the rate limit policy will be 600M for ports e 1/11 and 1/12, and 600M for ports e 1/13 and 1/14.

Maximum number of traffic policies supported on a device

The maximum number of supported active traffic policies is a system-wide parameter and depends on the device you are configuring, as follows:

- By default, up to 1024 active traffic policies are supported on Layer 2 switches. This value is fixed on Layer 2 switches and cannot be modified.
- For FastIron devices other than the FCX, the number of active traffic policies supported on Layer 3 switches varies depending on the configuration and the available system memory. The default value and also the maximum number of traffic policies supported on Layer 3 switches is 50.
- On FCX devices, up to 1024 active traffic policies are supported on Layer 3 switches. This is the default value as well as the maximum value.

NOTE

On FCX devices, by default 992 of the maximum of 1024 active traffic policies are applied. The other 32 are reserved and the **show traff** command returns zero references/bindings beyond the 992 traffic policies.

The **show default values** command displays the maximum number of traffic conditioners that can be applied, in the *hw-traffic-conditioner* section of the results. The configurable tables and their defaults and maximum values can be obtained using the **show default** command.

Setting the maximum number of traffic policies supported on a Layer 3 device

NOTE

This configuration is supported on FastIron devices with the exception of the FCX platforms. Setting the system-max for traffic policies is not required on FCX platforms as the default number of traffic policies is also the maximum number.

If desired, you can adjust the maximum number of active traffic policies that a Layer 3 device will support. To do so, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#system-max hw-traffic-conditioner 25
Brocade(config)#write memory
Brocade(config)#reload
```

NOTE

You must save the configuration and reload the software to place the change into effect.

Syntax: [no] **system-max hw-traffic-conditioner** <num>

The <num> variable is a value from 0 through *n*, where 0 disables hardware resources for traffic policies, and *n* is a number up to 50. The maximum number you can configure depends on the configuration and available memory on your device. If the configuration you enter causes the device to exceed the available memory, the device will reject the configuration and display a warning message on the console.

NOTE

Brocade does not recommend setting the system maximum for traffic policies to 0 (zero), because this renders traffic policies ineffective.

ACL-based rate limiting using traffic policies

ACL-based rate limiting provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs. This feature is available in the Layer 2 and Layer 3 code.

To configure ACL-based rate limiting, you create individual traffic policies, and then reference the traffic policies in one or more ACL entries (also called *clauses* or *statements*). The traffic policies become effective on ports to which the ACLs are bound.

When you configure a traffic policy for rate limiting, the device automatically enables rate limit counting, similar to the two-rate three-color marker (trTCM) mechanism described in RFC 2698 for adaptive rate limiting, and the single-rate three-color marker (srTCM) mechanism described in RFC 2697 for fixed rate limiting. This feature counts the number of bytes and trTCM or srTCM conformance level per packet to which rate limiting traffic policies are applied. Refer to [“ACL statistics and rate limit counting”](#) on page 1782.

You can configure ACL-based rate limiting on the following interface types:

- Physical Ethernet interfaces
- Virtual interfaces
- Trunk ports

- Specific VLAN members on a port (refer to [“Applying an IPv4 ACL to specific VLAN members on a port \(Layer 2 devices only\)”](#) on page 1734)
- A subset of ports on a virtual interface (refer to [“Applying an IPv4 ACL to a subset of ports on a virtual interface \(Layer 3 devices only\)”](#) on page 1735)

Support for fixed rate limiting and adaptive rate limiting

NOTE

ACL-based fixed rate limiting is supported on all FastIron devices. ACL-based adaptive rate limiting is supported on FastIron X Series and Brocade FCX Series devices only. It is not supported on FastIron WS devices.

FastIron devices support the following types of ACL-based rate limiting:

- Fixed rate limiting – Enforces a strict bandwidth limit. The device forwards traffic that is within the limit but either drops all traffic that exceeds the limit, or forwards all traffic that exceeds the limit at the lowest priority level, according to the action specified in the traffic policy.
- Adaptive rate limiting – Enforces a flexible bandwidth limit that allows for bursts above the limit. You can configure adaptive rate limiting to forward traffic, modify the IP precedence of and forward traffic, or drop traffic based on whether the traffic is within the limit or exceeds the limit.

Configuring ACL-based fixed rate limiting

Use the procedures in this section to configure ACL-based fixed rate limiting. Before configuring this feature, see what to consider in [“Configuration notes and feature limitations for traffic policies”](#) on page 1774.

Fixed rate limiting enforces a strict bandwidth limit. The port forwards traffic that is within the limit. If the port receives more than the specified number of fragments in a one-second interval, the device either drops or forwards subsequent fragments in hardware, depending on the action you specify.

To implement the ACL-based fixed rate limiting feature, first create a traffic policy, and then reference the policy in an extended ACL statement. Lastly, bind the ACL to an interface. Complete the following steps.

1. Create a traffic policy. Enter a command such as the following.

```
Brocade(config)#traffic-policy TPD1 rate-limit fixed 100 exceed-action drop
```

2. Create an extended ACL entry or modify an existing extended ACL entry that references the traffic policy. Enter a command such as the following.

```
Brocade(config)#access-list 101 permit ip host 210.10.12.2 any traffic-policy TPD1
```

3. Bind the ACL to an interface. Enter commands such as the following.

```
Brocade(config)#interface ethernet 5
Brocade(config-if-e5)#ip access-group 101 in
Brocade(config-if-e5)#exit
```

The previous commands configure a fixed rate limiting policy that allows port e5 to receive a maximum traffic rate of 100 kbps. If the port receives additional bits during a given one-second interval, the port drops the additional inbound packets that are received within that one-second interval.

Syntax: [no] **traffic-policy** <TPD name> **rate-limit fixed** <cir value> **exceed-action** <action> [count]

Syntax: **access-list** <num> **permit** | **deny**.... **traffic policy** <TPD name>

Syntax: [no] **ip access-group** <num> **in**

NOTE

For brevity, some parameters were omitted from the **access-list** syntax.

The software allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. The software does not issue a warning or error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

The **traffic-policy** <TPD name> parameter is the name of the traffic policy definition. This value can be eight or fewer alphanumeric characters.

The **rate-limit fixed** <cir value> parameter specifies that the traffic policy will enforce a strict bandwidth. The <cir value> variable is the committed information rate in kbps. This value can be from 64 through 1,000,000 Kbps.

The **exceed-action** <action> parameter specifies the action to be taken when packets exceed the configured committed information rate (CIR) value. Refer to [“Specifying the action to be taken for packets that are over the limit”](#) on page 1781.

The **count** parameter is optional and enables ACL counting. Refer to [“ACL statistics and rate limit counting”](#) on page 1782.

ACL-based adaptive rate limiting configuration

NOTE

ACL-based adaptive rate limiting is supported on FastIron X Series and Brocade FCX Series devices. It is not supported on FastIron WS devices.

Use the procedures in this section to configure ACL-based adaptive rate limiting. Before configuring this feature, see what to consider in [“Configuration notes and feature limitations for traffic policies”](#) on page 1774.

[Table 290](#) lists the configurable parameters for ACL-based adaptive rate limiting.

TABLE 290 ACL based adaptive rate limiting parameters

Parameter	Definition
Committed Information Rate (CIR)	The guaranteed kilobit rate of inbound traffic that is allowed on a port.
Committed Burst Size (CBS)	The number of bytes per second allowed in a burst before some packets will exceed the committed information rate. Larger bursts are more likely to exceed the rate limit. The CBS must be a value greater than zero (0). Brocade recommends that this value be equal to or greater than the size of the largest possible IP packet in a stream.
Peak Information Rate (PIR)	The maximum kilobit rate for inbound traffic on a port. The PIR must be equal to or greater than the CIR.
Peak Burst Size (PBS)	The number of bytes per second allowed in a burst before all packets will exceed the peak information rate. The PBS must be a value greater than zero (0). Brocade recommends that this value be equal to or greater than the size of the largest possible IP packet in the stream.

If a port receives more than the configured bit or byte rate in a one-second interval, the port will either drop or forward subsequent data in hardware, depending on the action you specify.

Configuring ACL-based adaptive rate limiting

To implement the ACL-based adaptive rate limiting feature, first create a traffic policy, and then reference the policy in an extended ACL statement. Lastly, bind the ACL to an interface. Complete the following steps.

1. Create a traffic policy. Enter a command such as the following.

```
Brocade(config)#traffic-policy TPDfour rate-limit adaptive cir 10000 cbs 1600
pir 20000 pbs 4000 exceed-action drop
```

2. Create a new extended ACL entry or modify an existing extended ACL entry that references the traffic policy. Enter a command such as the following.

```
Brocade(config)#access-list 104 permit ip host 210.10.12.2 any traffic-policy
TPDfour
```

3. Bind the ACL to an interface. Enter commands such as the following.

```
Brocade(config)#interface ethernet 7
Brocade(config-if-e7)#ip access-group 104 in
Brocade(config-if-e7)#exit
```

The previous commands configure an adaptive rate limiting policy that enforces a guaranteed committed rate of 10000 kbps on port e7 and allows bursts of up to 1600 bytes. It also enforces a peak rate of 20000 kbps and allows bursts of 4000 bytes above the PIR limit. If the port receives additional bits during a given one-second interval, the port drops all packets on the port until the next one-second interval starts.

Syntax: [no] **traffic-policy** <TPD name> **rate-limit adaptive cir** <cir value> **cbs** <cbs value> **pir** <pir value> **pbs** <pbs value> **exceed-action** <action> [count]

Syntax: **access-list** <num> **permit | deny.... traffic policy** <TPD name>

Syntax: [no] **ip access-group** <num> **in**

NOTE

For brevity, some parameters were omitted from the **access-list** syntax.

The software allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. The software does not issue a warning or error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

The **traffic-policy** *<TPD name>* parameter is the name of the traffic policy definition. This value can be eight or fewer alphanumeric characters.

The **rate-limit adaptive cir** *<cir value>* specifies that the policy will enforce a flexible bandwidth limit that allows for bursts above the limit. The *<cir value>* variable is the committed information rate in kbps. Refer to [Table 290](#).

The **cbs** *<cbs value>* parameter is the committed burst size in bytes. Refer to [Table 290](#).

The **pir** *<pir value>* parameter is the peak information rate in kbps. Refer to [Table 290](#).

The **pbs** *<pbs value>* parameter is the peak burst size in bytes. Refer to [Table 290](#).

The **exceed-action** *<action>* parameter specifies the action to be taken when packets exceed the configured values. Refer to [“Specifying the action to be taken for packets that are over the limit”](#) on page 1781.

The **count** parameter is optional and enables ACL counting. Refer to [“ACL statistics and rate limit counting”](#) on page 1782.

Inspecting the 802.1p bit in the ACL for adaptive rate limiting

NOTE

This feature is supported on FastIron X Series IPv6 devices and Brocade FCX Series devices only. It is *not* supported on FastIron WS Series devices.

You can configure the Brocade device to rate limit traffic for a specified 802.1p priority value. To do so, complete the following configuration steps.

1. Create an adaptive rate limiting traffic policy. Enter command such as the following:

```
Brocade(config)#traffic-policy adap rate-limit adaptive cir 1000 cbs 1000 pir
2000 pbs 10000 exceed-action drop
```

2. Create an IPv4 extended ACL or IPv6 ACL that includes the traffic policy and 802.1p priority matching value. Enter a command such as the following:

```
Brocade(config)#access-list 136 permit ip any any 802.1p-priority matching 3
traffic-policy adap
```

3. Bind the ACL to an interface. Enter commands such as the following,.

```
Brocade(config)#interface ethernet 7
Brocade(config-if-e7)#ip access-group 136 in
Brocade(config-if-e7)#exit
```

Use the **show access-list accounting** command to view accounting statistics. For more information, refer to [“Viewing ACL and rate limit counters”](#) on page 1784.

Specifying the action to be taken for packets that are over the limit

You can specify the action to be taken when packets exceed the configured CIR value for fixed rate limiting, or the CIR, CBS, PIR, and PBS values for adaptive rate limiting. You can specify one of the following actions:

- Drop packets that exceed the limit.
- Permit packets that exceed the limit and forward them at the lowest priority level.

Dropping packets that exceed the limit

This section shows some example configurations and provides the CLI syntax for configuring a port to drop packets that exceed the configured limits for rate limiting.

The following example shows a fixed rate limiting configuration.

```
Brocade(config)#traffic-policy TPD1 rate-limit fixed 10000 exceed-action drop
```

The command sets the fragment threshold at 10,000 packet fragments per second. If the port receives more than 10,000 packet fragments in a one-second interval, the device drops the excess fragments.

Syntax: [no] traffic-policy <TPD name> rate-limit fixed <cir value> exceed-action drop

The following example shows an adaptive rate limiting configuration.

```
Brocade(config)#traffic-policy TPDfour rate-limit adaptive cir 10000 cbs 1600
pir 20000 pbs 4000 exceed-action drop
```

The command configures an adaptive rate limiting policy that enforces a guaranteed committed rate of 10000 kbps and allows bursts of up to 1600 bytes. It also enforces a peak rate of 20000 kbps and allows bursts of 4000 bytes above the PIR limit. If the port receives additional bits during a given one-second interval, the port drops all packets on the port until the next one-second interval starts.

Syntax: **[no] traffic-policy** <TPD name> **rate-limit adaptive cir** <cir value> **cbs** <cbs value> **pir** <pir value> **pbs** <pbs value> **exceed-action drop**

Permitting packets that exceed the limit

This section shows some example configurations and provides the CLI syntax for configuring a port to permit packets that exceed the configured limit for rate limiting.

The following example shows a fixed rate limiting configuration.

```
Brocade(config)#traffic-policy TPD1 rate-limit fixed 10000 exceed-action
permit-at-low-pri
```

The command sets the fragment threshold at 10,000 packet fragments per second. If the port receives more than 10,000 packet fragments in a one-second interval, the device takes the specified action. The action specified with this command is to permit excess fragments and forward them at the lowest priority level.

Syntax: **[no] traffic-policy** <TPD name> **rate-limit fixed** <cir value> **exceed-action permit-at-low-pri**

The following example shows an adaptive rate limiting configuration.

```
Brocade(config)#traffic-policy TPDfour rate-limit adaptive cir 10000 cbs 1600
pir 20000 pbs 4000 exceed-action permit-at-low-pri
```

The command configures an adaptive rate limiting policy that enforces a guaranteed committed rate of 10000 kbps and allows bursts of up to 1600 bytes. It also enforces a peak rate of 20000 kbps and allows bursts of 4000 bytes above the PIR limit. If the port receives additional bits during a given one-second interval, the port permits all packets on the port and forwards the packets at the lowest priority level.

Syntax: **[no] traffic-policy** <TPD name> **rate-limit adaptive cir** <cir value> **cbs** <cbs value> **pir** <pir value> **pbs** <pbs value> **exceed-action permit-at-low-pri**

ACL statistics and rate limit counting

ACL statistics, also called *ACL counting*, enables the Brocade device to count the number of packets and the number of bytes per packet to which ACL filters are applied.

Rate limit counting counts the number of bytes and the conformance level per packet to which rate limiting traffic policies are applied. The device uses the counting method similar to the two-rate three-color marker (trTCM) mechanism described in RFC 2698 for adaptive rate limiting, and the single-rate three-color marker (srTCM) mechanism described in RFC 2697 for fixed rate limiting. Rate limit counting is automatically enabled when a traffic policy is enforced (active). You can view these counters using the **show** commands listed in [“Viewing traffic policies”](#) on page 1786.

Enabling ACL statistics

NOTE

ACL statistics and ACL counting are used interchangeably throughout this chapter and mean the same thing.

NOTE

The FastIron WS does not support the use of traffic policies for ACL statistics only. However, these models do support the use of traffic policies for ACL statistics together with rate limiting traffic policies. Refer to [“Enabling ACL statistics with rate limiting traffic policies”](#) on page 1784.

Use the procedures in this section to configure ACL statistics. Before configuring ACL statistics, see what to consider in [“Configuration notes and feature limitations for traffic policies”](#) on page 1774.

To enable ACL statistics on a FastIron X Series device, first create a traffic policy, and then reference the traffic policy in an extended ACL entry. Lastly, bind the ACL to an interface. The ACL counting policy becomes effective on ports to which the ACLs are bound.

You also can enable ACL statistics when you create a traffic policy for rate limiting. Refer to [“Enabling ACL statistics with rate limiting traffic policies”](#) on page 1784.

Complete the following steps to implement the ACL statistics feature.

1. Create a traffic policy. Enter a command such as the following.

```
Brocade(config)#traffic-policy TPD5 count
```

2. Create an extended ACL entry or modify an existing extended ACL entry that references the traffic policy definition. Enter a command such as the following.

```
Brocade(config)#access-list 101 permit ip host 210.10.12.2 any traffic-policy TPD5
```

3. Bind the ACL to an interface. Enter commands such as the following.

```
Brocade(config)#interface ethernet 4
Brocade(config-if-e4)#ip access-group 101 in
Brocade(config-if-e4)#exit
```

The previous commands configure an ACL counting policy and apply it to port e4. Port e4 counts the number of packets and the number of bytes on the port that were permitted or denied by ACL filters.

Syntax: [no] traffic-policy <TPD name> count

Syntax: access-list <num> permit | deny.... traffic policy <TPD name>

Syntax: [no] ip access-group <num> in

NOTE

For brevity, some parameters were omitted from the **access-list** syntax.

The software allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. The software does not issue a warning or error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

The `<TPD name>` variable is the name of the traffic policy definition. This value can be eight alphanumeric characters or less.

Enabling ACL statistics with rate limiting traffic policies

The configuration example in the section “[Enabling ACL statistics](#)” on page 1783 shows how to enable ACL counting without having to configure parameters for rate limiting. You also can enable ACL counting while defining a rate limiting traffic policy, as illustrated in the following configuration examples.

To enable ACL counting while defining traffic policies for fixed rate limiting, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#traffic-policy TPD1 rate-limit fixed 1000 count
Brocade(config)#traffic-policy TPD2 rate-limit fixed 10000 exceed-action drop
count
```

Syntax: `[no] traffic-policy <TPD name> rate-limit fixed <cir value> count`

Syntax: `[no] traffic-policy <TPD name> rate-limit fixed <cir value> exceed-action <action> count`

To enable ACL counting while defining traffic policies for adaptive rate limiting, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#traffic-policy TPDA4 rate-limit adaptive cir 10000 cbs 1600 pir
20000 pbs 4000 count
Brocade(config)#traffic-policy TPDA5 rate-limit adaptive cir 10000 cbs 1600 pir
20000 pbs 4000 exceed-action permit-at-low-pri count
```

Syntax: `[no] traffic-policy <TPD name> rate-limit adaptive cir <cir value> cbs <cbs value> pir <pir value> pbs <pbs value> count`

Syntax: `[no] traffic-policy <TPD name> rate-limit adaptive cir <cir value> cbs <cbs value> pir <pir value> pbs <pbs value> exceed-action <action> count`

Viewing ACL and rate limit counters

When ACL counting is enabled on the Brocade device, you can use **show** commands to display the total packet count and byte count of the traffic filtered by ACL statements. The output of the **show** commands also displays the rate limiting traffic counters, which are automatically enabled for active rate limiting traffic policies.

Use either the **show access-list accounting traffic-policy** command or the **show statistics traffic-policy** command to display ACL and traffic policy counters. The outputs of these commands is identical.

NOTE

In the SX-FI48GPP module only, the outputs of these commands are identical with one exception. When ACL counting is shown by **show statistics traffic-policy**, the Packet Count is not supported and displays “N/A”.

The following example shows the output from the **show access-list accounting** command.

```
Brocade#show access-list accounting traffic-policy g_voip
```

```
Traffic Policy - g_voip:
```

```
General Counters:
```

Port Region#	Byte Count	Packet Count
7 (4/1 - 4/12)	85367040	776064
All port regions	84367040	776064

```
Rate Limiting Counters:
```

Port Region#	Green Conformance	Yellow Conformance	Red Conformance
7 (4/1 - 4/12)	329114195612139520	37533986897781760	0
All port regions	329114195612139520	37533986897781760	0

Syntax: `show access-list accounting traffic-policy [<TPD name>]`

or

Syntax: `show statistics traffic-policy [<TPD name>]`

The `<TPD name>` variable is the name of the traffic policy definition for which you want to display ACL and traffic policy counters.

[Table 291](#) explains the output of the `show access-list accounting traffic-policy` and `show statistics traffic-policy` commands.

TABLE 291 ACL and rate limit counting statistics

Parameter	Description
Traffic Policy	The name of the traffic policy.
General Counters	
Port Region #	The port region to which the active traffic policy applies.
Byte Count	The number of bytes that were filtered (matched ACL clauses).
Packet Count	The number of packets that were filtered (matched ACL clauses).
Rate Limiting Counters	
Port Region#	The port region to which the active traffic policy applies.
Green Conformance	The number of bytes that did not exceed the CIR packet rate.
Yellow Conformance	The number of bytes that exceeded the CIR packet rate.
Red Conformance	The number of bytes that exceeded the PIR packet rate.

Clearing ACL and rate limit counters

The Brocade device keeps a running tally of the number of packets and the number of bytes per packet that are filtered by ACL statements and rate limiting traffic policies. You can clear these accumulated counters, essentially resetting them to zero. To do so, use either the `clear access-list accounting traffic-policy` command or the `clear statistics traffic-policy` command.

To clear the counters for ACL counting and rate limit counting, enter commands such as the following.

```
Brocade(config)#clear access-list accounting traffic-policy CountOne
```

```
Brocade(config)#clear statistics traffic-policy CountTwo
```

Syntax: `clear access-list accounting traffic-policy <TPD name>`

or

Syntax: `clear statistics traffic-policy <TPD name>`

The `<TPD name>` is the name of the traffic policy definition for which you want to clear traffic policy counters.

Viewing traffic policies

To view traffic policies that are currently defined on the Brocade device, enter the **show traffic-policy** command. The following example shows displayed output. [Table 292](#) explains the output of the **show traffic-policy** command.

```
Brocade#show traffic-policy t_voip
Traffic Policy - t_voip:
Metering Enabled, Parameters:
      Mode: Adaptive Rate-Limiting
      cir: 100 kbps,      cbs: 2000 bytes,      pir: 200 kbps,      pbs: 4000
bytes
Counting Not Enabled
Number of References/Bindings:1
```

Syntax: `show traffic-policy [<TPD name>]`

To display all traffic policies, enter the **show traffic-policy** command without entering a TPD name.

TABLE 292 Traffic policy information

Parameter	Description
Traffic Policy	The name of the traffic policy.
Metering	Shows whether or not rate limiting was configured as part of the traffic policy: <ul style="list-style-type: none"> • Enabled – The traffic policy includes a rate limiting configuration. • Disabled – The traffic policy does not include a rate limiting configuration.
Mode	If rate limiting is enabled, this field shows the type of metering enabled on the port: <ul style="list-style-type: none"> • Fixed Rate-Limiting • Adaptive Rate-Limiting
cir	The committed information rate, in kbps, for the adaptive rate limiting policy.
cbs	The committed burst size, in bytes per second, for the adaptive rate- limiting policy.
pir	The peak information rate, in kbps, for the adaptive rate limiting policy.
pbs	The peak burst size, in bytes per second, for the adaptive rate limiting policy.
Counting	Shows whether or not ACL counting was configured as part of the traffic policy: <ul style="list-style-type: none"> • Enabled – Traffic policy includes an ACL counting configuration. • Disabled – Traffic policy does not include an ACL traffic counting configuration.
Number of References/Bindings	NOTE: This field does not apply to FastIron X Series devices. The number of port regions to which this traffic policy applies. For example, if the traffic policy is applied to a trunk group that includes ports e 9/9, 9/10, 9/11, and 9/12, the value in this field would be 2, because these four trunk ports are in two different port regions.

CPU rate-limiting

Unnecessary traffic to the switch CPU lowers the efficiency of the CPU and delays handling of other traffic that requires processing. CPU rate limiting is a CPU protection scheme which limits certain traffic types.

CPU rate limiting identifies the traffic type and assigns a maximum rate limit to the traffic type. The traffic types which are subjected to rate limiting include broadcast ARP and other exceptions, such as TTL exceed, IP MTU failed, reverse path check failed, IP fragments, and unsupported tunneling. Each of these types is rate-limited individually.

[Table 293](#) shows the rate limits for each rate-limited packet type and shows which platforms on which each rate limit applies. These rates cannot be configured by users currently.

TABLE 293 CPU rate limits for packet type and applicable platforms

Packet type	Rate limit in packets per second	Applicable platforms
ARP	6000	All
IP TTL exceed, or Reverse path check failed	150	All
IP MTU exceed, IP tunnel-terminated packets which are fragmented or has options, or IP tunnel-terminated packets with unsupported GRE tunnel header	3000	All
IP Unicast packets mirrored to CPU due to ICMP redirect	100	All
Bridge packets forward to CPU	5000	FCX and ICX

All currently supported FastIron devices support the CPU rate-limiting feature. However, on the FSX devices, only the following modules support this feature:

- SX-FI-24GPP
- SX-FI-24HF
- SX-FI-2XG
- SX-FI-8XG
- SX-FI48GPP

CPU rate-limiting

802.1X Port Security

Table 294 lists individual Brocade switches and the 802.1X port security features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 294 Supported 802.1X port security features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
802.1X port security	Yes	Yes	Yes	Yes	Yes
Multiple host authentication	Yes	Yes	Yes	Yes	Yes
EAP pass-through support	Yes	Yes	Yes	Yes	Yes
802.1X accounting	Yes	Yes	Yes	Yes	Yes
802.1X dynamic assignment for ACL, MAC address filter, and VLAN	Yes	Yes	Yes	Yes	Yes
Automatic removal of Dynamic VLAN for 802.1X ports	Yes	Yes	Yes	Yes	Yes
RADIUS timeout action	Yes	Yes	Yes	Yes	Yes
802.1X and multi-device port authentication on the same port	Yes	Yes	Yes	Yes	Yes
802.1X and sFlow <ul style="list-style-type: none"> 802.1X username export support for encrypted and non-encrypted EAP types 	Yes	Yes	Yes	Yes	Yes

IETF RFC support

Brocade FastIron devices support the IEEE 802.1X standard for authenticating devices attached to LAN ports. Using 802.1X port security, you can configure a FastIron device to grant access to a port based on information supplied by a client to an authentication server.

When a user logs on to a network that uses 802.1X port security, the Brocade device grants (or does not grant) access to network services after the user is authenticated by an authentication server. The user-based authentication in 802.1X port security provides an alternative to granting network access based on a user IP address, MAC address, or subnetwork.

The Brocade implementation of 802.1X port security supports the following RFCs:

- RFC 2284 PPP Extensible Authentication Protocol (EAP)
- RFC 2865 Remote Authentication Dial In User Service (RADIUS)
- RFC 2869 RADIUS Extensions

How 802.1X port security works

This section explains the basic concepts behind 802.1X port security, including device roles, how the devices communicate, and the procedure used for authenticating clients.

NOTE

802.1X Port Security cannot be configured on MAC Port Security-enabled ports.

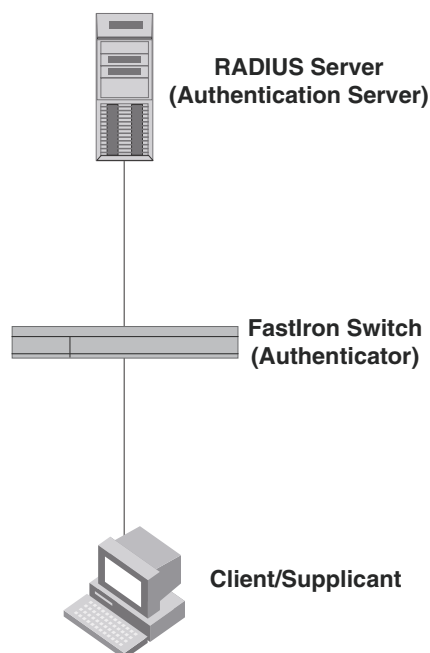
Device roles in an 802.1X configuration

The 802.1X standard defines the roles of *Client/Supplicant*, *Authenticator*, and *Authentication Server* in a network.

The Client (known as a **Supplicant** in the 802.1X standard) provides username/password information to the Authenticator. The Authenticator sends this information to the Authentication Server. Based on the Client's information, the Authentication Server determines whether the Client can use services provided by the Authenticator. The Authentication Server passes this information to the Authenticator, which then provides services to the Client, based on the authentication result.

[Figure 188](#) illustrates these roles.

FIGURE 188 Authenticator, client/supplicant, and authentication server in an 802.1X configuration



Authenticator – The device that controls access to the network. In an 802.1X configuration, the Brocade device serves as the Authenticator. The Authenticator passes messages between the Client and the Authentication Server. Based on the identity information supplied by the Client, and the authentication information supplied by the Authentication Server, the Authenticator either grants or does not grant network access to the Client.

Client/Supplicant – The device that seeks to gain access to the network. Clients must be running software that supports the 802.1X standard (for example, the Windows XP operating system). Clients can either be directly connected to a port on the Authenticator, or can be connected by way of a hub.

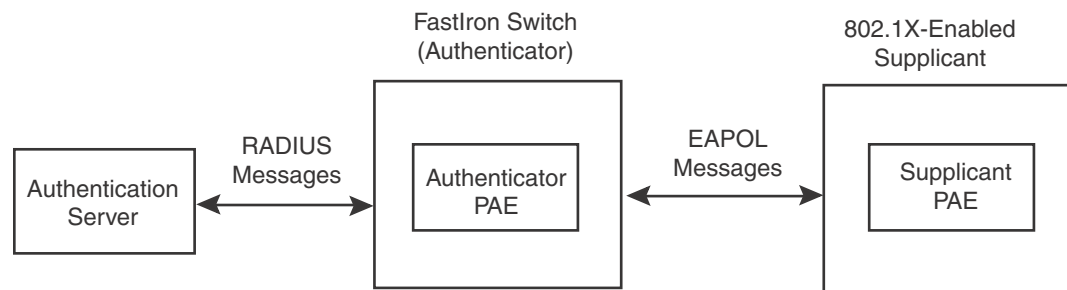
Authentication server – The device that validates the Client and specifies whether or not the Client may access services on the device. Brocade supports Authentication Servers running RADIUS.

Communication between the devices

For communication between the devices, 802.1X port security uses the **Extensible Authentication Protocol** (EAP), defined in RFC 2284. The 802.1X standard specifies a method for encapsulating EAP messages so that they can be carried over a LAN. This encapsulated form of EAP is known as EAP over LAN (**EAPOL**). The standard also specifies a means of transferring the EAPOL information between the Client/Supplicant, Authenticator, and Authentication Server.

EAPOL messages are passed between the **Port Access Entity (PAE)** on the Supplicant and the Authenticator. [Figure 189](#) shows the relationship between the Authenticator PAE and the Supplicant PAE.

FIGURE 189 Authenticator PAE and supplicant PAE



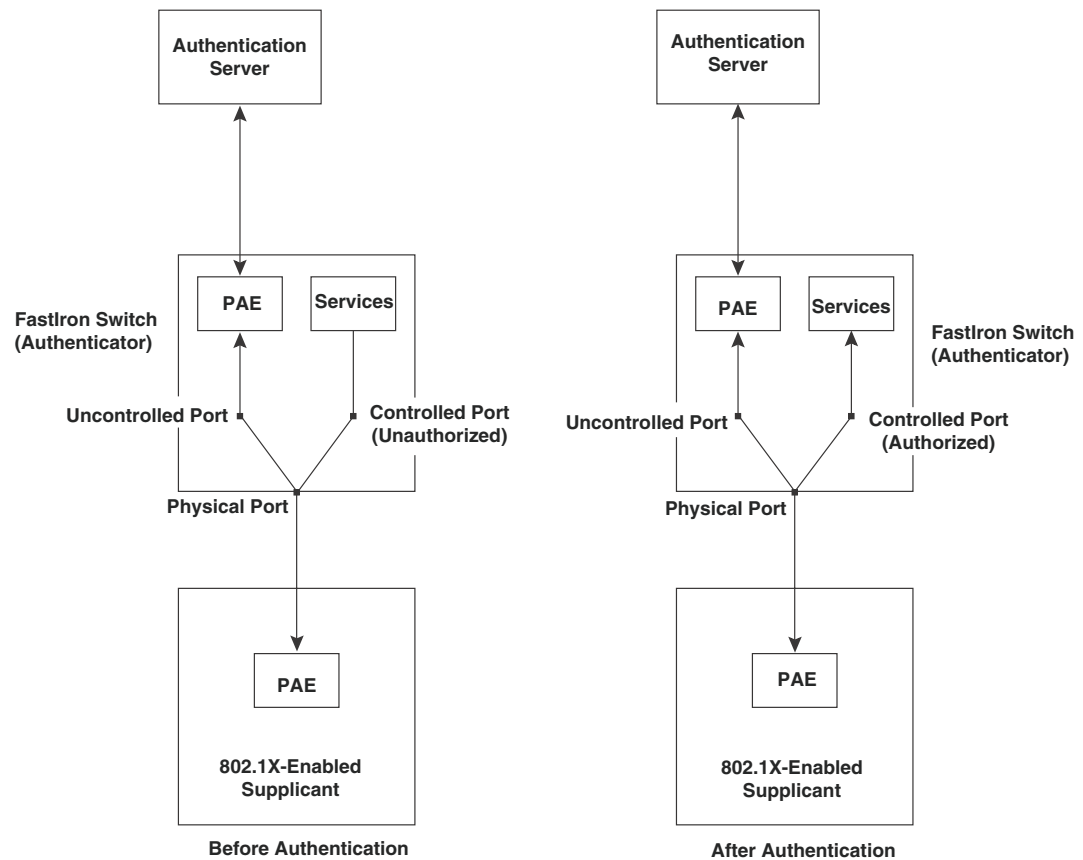
Authenticator PAE – The Authenticator PAE communicates with the Supplicant PAE, receiving identifying information from the Supplicant. Acting as a RADIUS client, the Authenticator PAE passes the Supplicant information to the Authentication Server, which decides whether the Supplicant can gain access to the port. If the Supplicant passes authentication, the Authenticator PAE grants it access to the port.

Supplicant PAE – The Supplicant PAE supplies information about the Client to the Authenticator PAE and responds to requests from the Authenticator PAE. The Supplicant PAE can also initiate the authentication procedure with the Authenticator PAE, as well as send log off messages.

Controlled and uncontrolled ports

A physical port on the device used with 802.1X port security has two virtual access points: a **controlled** port and an **uncontrolled** port. The controlled port provides full access to the network. The uncontrolled port provides access only for EAPOL traffic between the Client and the Authentication Server. When a Client is successfully authenticated, the controlled port is opened to the Client. [Figure 190](#) illustrates this concept.

FIGURE 190 Controlled and uncontrolled ports before and after client authentication



Before a Client is authenticated, only the uncontrolled port on the Authenticator is open. The uncontrolled port allows only EAPOL frames to be exchanged between the Client and the Authentication Server. The controlled port is in the unauthorized state and allows no traffic to pass through.

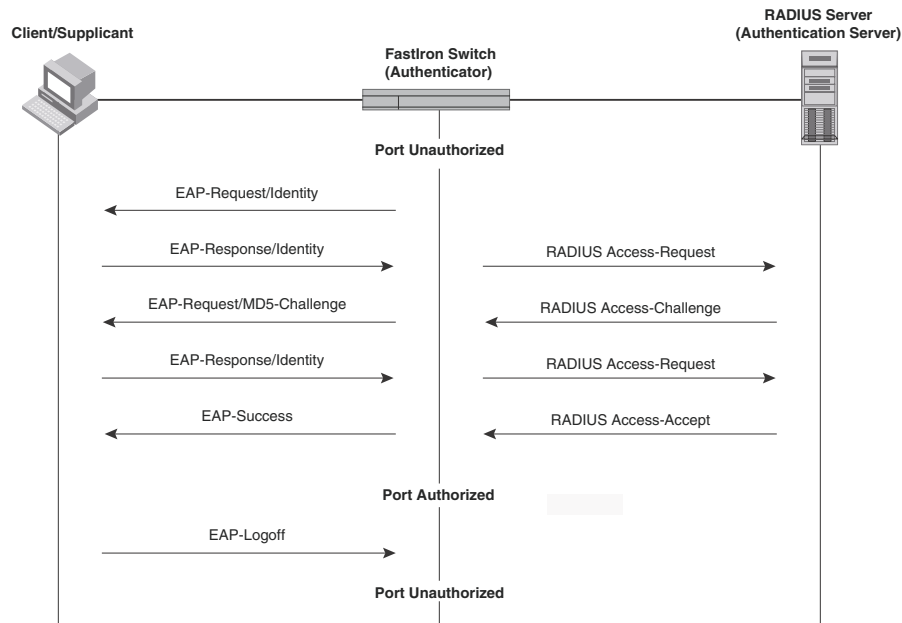
During authentication, EAPOL messages are exchanged between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are exchanged between the Authenticator PAE and the Authentication Server. Refer to [“Message exchange during authentication”](#) on page 1793 for an example of this process. If the Client is successfully authenticated, the controlled port becomes authorized, and traffic from the Client can flow through the port normally.

By default, all controlled ports on the Brocade device are placed in the authorized state, allowing all traffic. When authentication is activated on an 802.1X-enabled interface, the interface controlled port is placed initially in the unauthorized state. When a Client connected to the port is successfully authenticated, the controlled port is then placed in the authorized state until the Client logs off. Refer to [“Enabling 802.1X port security”](#) on page 1810 for more information.

Message exchange during authentication

Figure 191 illustrates a sample exchange of messages between an 802.1X-enabled Client, a FastIron switch acting as Authenticator, and a RADIUS server acting as an Authentication Server.

FIGURE 191 Message exchange between client/supplicant, authenticator, and authentication server



In this example, the Authenticator (the FastIron switch) initiates communication with an 802.1X-enabled Client. When the Client responds, it is prompted for a username (255 characters maximum) and password. The Authenticator passes this information to the Authentication Server, which determines whether the Client can access services provided by the Authenticator. When the Client is successfully authenticated by the RADIUS server, the port is authorized. When the Client logs off, the port becomes unauthorized again.

The Brocade 802.1X implementation supports dynamic VLAN assignment. If one of the attributes in the Access-Accept message sent by the RADIUS server specifies a VLAN identifier, and this VLAN is available on the Brocade device, the client port is moved from its default VLAN to the specified VLAN. When the client disconnects from the network, the port is placed back in its default VLAN. Refer to [“Dynamic VLAN assignment for 802.1X port configuration”](#) on page 1802 for more information.

If a Client does not support 802.1X, authentication cannot take place. The Brocade device sends EAP-Request/Identity frames to the Client, but the Client does not respond to them.

When a Client that supports 802.1X attempts to gain access through a non-802.1X-enabled port, it sends an EAP start frame to the Brocade device. When the device does not respond, the Client considers the port to be authorized, and starts sending normal traffic.

Brocade devices support Identity and MD5-challenge requests in EAP Request/Response messages as well as the following 802.1X authentication challenge types:

NOTE

Refer to also [“EAP pass-through support”](#) on page 1795.

- **EAP-TLS (RFC 2716)** – EAP Transport Level Security (TLS) provides strong security by requiring both client and authentication server to be identified and validated through the use of public key infrastructure (PKI) digital certificates. EAP-TLS establishes a tunnel between the client and the authentication server to protect messages from unauthorized users' eavesdropping activities. Since EAP-TLS requires PKI digital certificates on both the clients and the authentication servers, the roll out, maintenance, and scalability of this authentication method is much more complex than other methods. EAP-TLS is best for installations with existing PKI certificate infrastructures.
- **EAP-TTLS (Internet-Draft)** – The EAP Tunnelled Transport Level Security (TTLS) is an extension of EAP-TLS. Like TLS, EAP-TTLS provides strong authentication; however it requires only the authentication server to be validated by the client through a certificate exchange between the server and the client. Clients are authenticated by the authentication server using user names and passwords.

A TLS tunnel can be used to protect EAP messages and existing user credential services such as Active Directory, RADIUS, and LDAP. Backward compatibility for other authentication protocols such as PAP, CHAP, MS-CHAP, and MS-CHAP-V2 are also provided by EAP-TTLS. EAP-TTLS is not considered foolproof and can be fooled into sending identity credentials if TLS tunnels are not used. EAP-TTLS is suited for installations that require strong authentication without the use of mutual PKI digital certificates.

- **PEAP (Internet-Draft)** – Protected EAP Protocol (PEAP) is an Internet-Draft that is similar to EAP-TTLS. PEAP client authenticates directly with the backend authentication server. The authenticator acts as a pass-through device, which does not need to understand the specific EAP authentication protocols.

Unlike EAP-TTLS, PEAP does not natively support user name and password to authenticate clients against an existing user database such as LDAP. PEAP secures the transmission between the client and authentication server with a TLS encrypted tunnel. PEAP also allows other EAP authentication protocols to be used. It relies on the mature TLS keying method for its key creation and exchange. PEAP is best suited for installations that require strong authentication without the use of mutual certificates.

Configuration for these challenge types is the same as for the EAP-MD5 challenge type.

NOTE

If the 802.1X Client will be sending a packet that is larger than 1500 bytes, you must enable **jumbo** at the Global config level of the CLI. If the supplicant or the RADIUS server does not support jumbo frames and jumbo is enabled on the switch, you can set the CPU IP MTU size. Refer to [“Setting the IP MTU size”](#), next.

Setting the IP MTU size

When jumbo frames are enabled on a FastIron device and the certificate in use is larger than the standard packet size of 1500 bytes, 802.1X authentication will not work if the supplicant or the RADIUS server does not support jumbo frames. In this case, you can change the IP MTU setting so that the certificate will be fragmented before it is forwarded to the supplicant or server for processing. This feature is supported in the Layer 2 switch code only. It is not supported in the Layer 3 router code.

To enable this feature, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)# ip mtu 1500
```


Syntax: [no] ip mtu <num>

The <num> parameter specifies the MTU. Ethernet II packets can hold IP packets from 576 – 1500 bytes long. If jumbo mode is enabled, Ethernet II packets can hold IP packets from 576 – 10,222 bytes long. Ethernet SNAP packets can hold IP packets from 576 – 1492 bytes long. If jumbo mode is enabled, SNAP packets can hold IP packets from 576 to 10,214 bytes long. The default MTU is 1500 for Ethernet II packets and 1492 for SNAP packets.

EAP pass-through support

EAP pass-through is supported on FastIron devices that have 802.1X enabled. EAP pass-through support is fully compliant with RFC 3748, in which, by default, compliant pass-through authenticator implementations forward EAP challenge request packets of *any type*, including those listed in the previous section.

Configuration notes for setting the IP MTU size

If the 802.1X supplicant or authentication server will be sending packets that are greater than 1500 MTU, you should configure the device to accommodate a larger buffer size, in order to reduce problems during initial setup. Refer to [Chapter 26, “IP Configuration”](#).

Support for RADIUS user-name attribute in access-accept messages

Brocade 802.1X-enabled ports support the RADIUS user-name (type 1) attribute in the Access-Accept message returned during 802.1X authentication.

This feature is useful when the client/supplicant does not provide its user-name in the EAP-response/identity frame, and the username is key to providing useful information. For example, when the user-name attribute is sent in the Access-Accept message, it is then available for display in sFlow sample messages sent to a collector, and in the output of some show dot1x CLI commands, such as show dot1x mac-sessions.

This same information is sent as the “user-name” attribute of RADIUS accounting messages, and is sent to the RADIUS accounting servers.

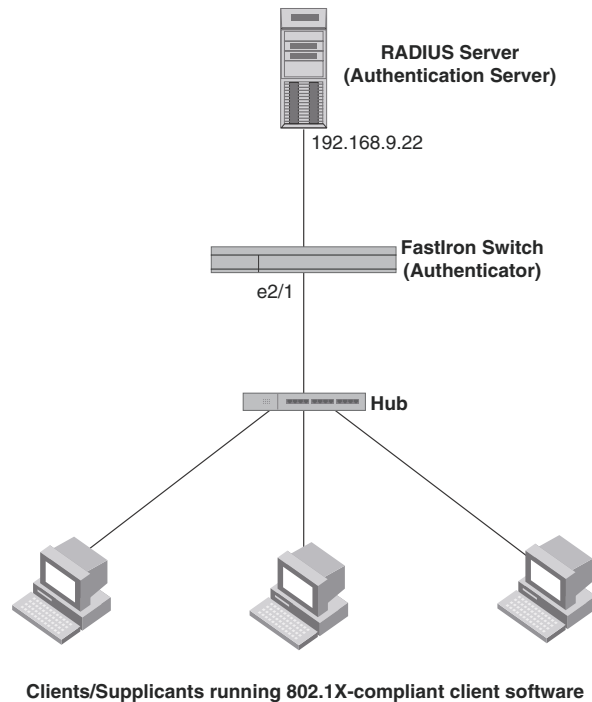
To enable this feature, add the following attribute on the RADIUS server.

Attribute name	Type	Value
user-name	1	<name> (string)

Authenticating multiple hosts connected to the same port

Brocade devices support 802.1X authentication for ports with more than one host connected to them. [Figure 192](#) illustrates a sample configuration where multiple hosts are connected to a single 802.1X port.

FIGURE 192 Multiple hosts connected to a single 802.1X-enabled port



If there are multiple hosts connected to a single 802.1X-enabled port, the Brocade device authenticates each of them individually. Each host authentication status is independent of the others, so that if one authenticated host disconnects from the network, it has no effect on the authentication status of any of the other authenticated hosts.

By default, traffic from hosts that cannot be authenticated by the RADIUS server is dropped in hardware. You can optionally configure the Brocade device to assign the port to a “restricted” VLAN if authentication of the Client is unsuccessful.

How 802.1X multiple-host authentication works

When multiple hosts are connected to a single 802.1X-enabled port on a Brocade device (as in [Figure 192](#)), 802.1X authentication is performed in the following way.

1. One of the 802.1X-enabled Clients attempts to log into a network in which a Brocade device serves as an Authenticator.
2. The Brocade device creates an internal session (called a ***dot1x-mac-session***) for the Client. A dot1x-mac-session serves to associate a Client MAC address and username with its authentication status.
3. The Brocade device performs 802.1X authentication for the Client. Messages are exchanged between the Brocade device and the Client, and between the device and the Authentication Server (RADIUS server). The result of this process is that the Client is either successfully authenticated or not authenticated, based on the username and password supplied by the client.
4. If the Client is successfully authenticated, the Client dot1x-mac-session is set to “access-is-allowed”. This means that traffic from the Client can be forwarded normally.

5. If authentication for the Client is unsuccessful the first time, multiple attempts to authenticate the client will be made as determined by the **attempts** variable in the **auth-fail-max-attempts** command.
 - Refer to [“Specifying the number of authentication attempts the device makes before dropping packets”](#) on page 1816 for information on how to do this.
6. If authentication for the Client is unsuccessful more than the number of times specified by the attempts variable in the **auth-fail-max-attempts** command, an **authentication-failure** action is taken. The authentication-failure action can be either to drop traffic from the Client, or to place the port in a “restricted” VLAN:
 - If the authentication-failure action is to drop traffic from the Client, then the Client dot1x-mac-session is set to “access-denied”, causing traffic from the Client to be dropped in hardware.
 - If the authentication-failure action is to place the port in a “restricted” VLAN, If the Client dot1x-mac-session is set to “access-restricted” then the port is moved to the specified restricted VLAN, and traffic from the Client is forwarded normally.
7. When the Client disconnects from the network, the Brocade device deletes the Client dot1x-mac-session. This does not affect the dot1x-mac-session or authentication status (if any) of the other hosts connected on the port.

Configuration notes for 802.1x multiple-host authentication

- The Client dot1x-mac-session establishes a relationship between the username and MAC address used for authentication. If a user attempts to gain access from different Clients (with different MAC addresses), he or she would need to be authenticated from each Client.
- If a Client has been denied access to the network (that is, the Client dot1x-mac-session is set to “access-denied”), then you can cause the Client to be re-authenticated by manually disconnecting the Client from the network, or by using the **clear dot1x mac-session** command. Refer to [“Clearing a dot1x-mac-session for a MAC address”](#) on page 1817 for information on this command.
- When a Client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the Client MAC address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. You can optionally change the software aging period for dot1x-mac-sessions or disable aging altogether. After the denied Client dot1x-mac-session is aged out, traffic from that Client is no longer blocked, and the Client can be re-authenticated.

In addition, you can configure disable aging for the dot1x-mac-session of Clients that have been granted either full access to the network, or have been placed in a restricted VLAN. After a Client dot1x-mac-session ages out, the Client must be re-authenticated. Refer to [“Disabling aging for dot1x-mac-sessions”](#) on page 1816 for more information.

- Dynamic IP ACL and MAC address filter assignment is supported in an 802.1X multiple-host configuration. Refer to [“Dynamically applying IP ACLs and MAC address filters to 802.1X ports”](#) on page 1806.
- 802.1X multiple-host authentication has the following additions:
 - Configurable hardware aging period for denied client dot1x-mac-sessions. Refer to [“Configurable hardware aging period for denied client dot1x-mac-sessions”](#) on page 1798.

- Dynamic ACL and MAC address filter assignment in 802.1X multiple-host configurations. Refer to [“Dynamically applying IP ACLs and MAC address filters to 802.1X ports”](#) on page 1806.
- Dynamic multiple VLAN assignment for 802.1X ports. Refer [“Dynamic multiple VLAN assignment for 802.1X ports”](#) on page 1803.
- Configure a restriction to forward authenticated and unauthenticated tagged and untagged clients to a restricted VLAN.
- Configure an override to send failed dot1x and non-dot1x clients to a restricted VLAN.
- Configure VLAN assignments for clients attempting to gain access through dual-mode ports.
- Enhancements to some **show** commands.
- Differences in command syntax for saving dynamic VLAN assignments to the startup-config file.

Configurable hardware aging period for denied client dot1x-mac-sessions

When one of the 802.1X-enabled Clients in a multiple-host configuration attempts to log into a network in which a Brocade device serves as an Authenticator, the device creates a dot1x-mac-session for the Client.

When a Client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the Client MAC address over a period of time. After a denied Client dot1x-mac-session ages out, the Client can be re-authenticated. Aging of a denied Client's dot1x-mac-session occurs in two phases, known as hardware aging and software aging.

The hardware aging period for a denied Client's dot1x-mac-session is not fixed at 70 seconds. The hardware aging period for a denied Client's dot1x-mac-session is equal to the length of time specified with the dot1x **timeout quiet-period** command. By default, the hardware aging time is 60 seconds. Once the hardware aging period ends, the software aging period begins. When the software aging period ends, the denied Client's dot1x-mac-session ages out, and the Client can be authenticated again.

802.1X port security and sFlow

sFlow is a standards-based protocol that allows network traffic to be sampled at a user-defined rate for the purpose of monitoring traffic flow patterns and identifying packet transfer rates on user-specified interfaces.

When you enable sFlow forwarding on an 802.1X-enabled interface, the samples taken from the interface include the user name string at the inbound or outbound port, or both, if that information is available.

For more information on sFlow, refer to [Chapter 15, “Network Monitoring”](#).

802.1X accounting

When 802.1X port security is enabled on the Brocade device, you can enable **802.1X accounting**. This feature enables the Brocade device to log information on the RADIUS server about authenticated 802.1X clients. The information logged on the RADIUS server includes the 802.1X client session ID, MAC address, and authenticating physical port number.

802.1X accounting works as follows.

1. A RADIUS server successfully authenticates an 802.1X client.
2. If 802.1X accounting is enabled, the Brocade device sends an 802.1X Accounting Start packet to the RADIUS server, indicating the start of a new session.
3. The RADIUS server acknowledges the Accounting Start packet.
4. The RADIUS server records information about the client.
5. When the session is concluded, the Brocade device sends an Accounting Stop packet to the RADIUS server, indicating the end of the session.
6. The RADIUS server acknowledges the Accounting Stop packet.

To enable 802.1X accounting, refer to [“802.1X accounting configuration”](#) on page 1818.

802.1X port security configuration

Configuring 802.1X port security on a Brocade device consists of the following tasks.

1. Configure the device interaction with the Authentication Server:
 - [“Configuring an authentication method list for 802.1X”](#) on page 1800
 - [“Setting RADIUS parameters”](#) on page 1800
 - [“Dynamic VLAN assignment for 802.1X port configuration”](#) on page 1802 (optional)
 - [“Dynamically applying IP ACLs and MAC address filters to 802.1X ports”](#) on page 1806
2. Configure the device role as the Authenticator:
 - [“Enabling 802.1X port security”](#) on page 1810
 - [“Initializing 802.1X on a port”](#) on page 1814 (optional)
3. Configure the device interaction with Clients:
 - [“Configuring periodic re-authentication”](#) on page 1811 (optional)
 - [“Re-authenticating a port manually”](#) on page 1812 (optional)
 - [“Setting the quiet period”](#) on page 1812 (optional)
 - [“Setting the wait interval for EAP frame retransmissions”](#) on page 1813 (optional)
 - [“Setting the maximum number of EAP frame retransmissions”](#) on page 1813 (optional)
 - [“Specifying a timeout for retransmission of messages to the authentication server”](#) on page 1814 (optional)
 - [“Allowing access to multiple hosts”](#) on page 1815 (optional)
 - [“MAC address filters for EAP frames”](#) on page 1817 (optional)

Configuring an authentication method list for 802.1X

To use 802.1X port security, you must specify an authentication method to be used to authenticate Clients. Brocade supports RADIUS authentication with 802.1X port security. To use RADIUS authentication with 802.1X port security, you create an authentication method list for 802.1X and specify RADIUS as an authentication method, then configure communication between the Brocade device and RADIUS server.

Example

```
Brocade(config)#aaa authentication dot1x default radius
```

Syntax: [no] **aaa authentication dot1x default** <method-list>

For the <method-list>, enter at least one of the following authentication methods

radius – Use the list of all RADIUS servers that support 802.1X for authentication.

none – Use no authentication. The Client is automatically authenticated by other means, without the device using information supplied by the Client.

NOTE

If you specify both **radius** and **none**, make sure **radius** comes before **none** in the method list.

Setting RADIUS parameters

To use a RADIUS server to authenticate access to a Brocade device, you must identify the server to the Brocade device.

Example

```
Brocade(config)#radius-server host 209.157.22.99 auth-port 1812 acct-port 1813
default key mirabeau dot1x
```

Syntax: **radius-server host** <ip-addr> | <ipv6-addr> | <server-name> [**auth-port** <num> | **acct-port** <num> | **default**] [**key** 0 | 1 <string>] [**dot1x**]

The **host** <ip-addr> | <ipv6-addr> | <server-name> parameter is either an IP address or an ASCII text string.

The **dot1x** parameter indicates that this RADIUS server supports the 802.1X standard. A RADIUS server that supports the 802.1X standard can also be used to authenticate non-802.1X authentication requests.

NOTE

To implement 802.1X port security, at least one of the RADIUS servers identified to the Brocade device must support the 802.1X standard.

Supported RADIUS attributes

Many IEEE 802.1X Authenticators will function as RADIUS clients. Some of the RADIUS attributes may be received as part of IEEE 802.1X authentication. Brocade devices support the following RADIUS attributes for IEEE 802.1X authentication:

- Username (1) – RFC 2865
- NAS-IP-Address (4) – RFC 2865
- NAS-Port (5) – RFC 2865

- Service-Type (6) – RFC 2865
- FilterId (11) – RFC 2865
- Framed-MTU (12) – RFC 2865
- State (24) – RFC 2865
- Vendor-Specific (26) – RFC 2865
- Session-Timeout (27) – RFC 2865
- Termination-Action (29) – RFC 2865
- Calling-Station-ID (31) – RFC 2865
- NAS-Port-Type (61) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Message-Authenticator (80) RFC 3579
- Tunnel-Private-Group-Id (81) – RFC 2868
- NAS-Port-id (87) – RFC 2869

Specifying the RADIUS timeout action

A RADIUS timeout occurs when the Brocade device does not receive a response from a RADIUS server within a specified time limit and after a certain number of retries. The time limit and number of retries can be manually configured using the CLI commands **radius-server timeout** and **radius-server retransmit**, respectively. If the parameters are not manually configured, the Brocade device applies the default value of three seconds time limit with a maximum of three retries.

You can better control port behavior when a RADIUS timeout occurs. That is, you can configure a port on the Brocade device to automatically pass or fail users being authenticated. A **pass** essentially bypasses the authentication process and permits user access to the network. A **fail** bypasses the authentication process and blocks user access to the network, unless restrict-vlan is configured, in which case, the user is placed into a VLAN with restricted or limited access. By default, the Brocade device will reset the authentication process and retry to authenticate the user.

Specify the RADIUS timeout action at the Interface level of the CLI.

Permit user access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and *permit* user access to the network, enter commands such as the following

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e100-3/1)#dot1x auth-timeout-action success
```

Syntax: [no] dot1x auth-timeout-action success

Once the *success* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

Re-authenticate a user

To configure RADIUS timeout behavior to bypass multi-device port authentication and *permit* user access to the network, enter commands similar to the following

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e100-3/1)#dot1x re-auth-timeout-success 60
```

Syntax: [no] **dot1x re-auth-timeout-success** <seconds>

The <seconds> parameter specifies the number of seconds the device will wait to re-authenticate a user after a timeout. The minimum value is 10 seconds. The maximum value is $2^{16}-1$ (maximum unsigned 16-bit value).

Deny user access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and *block* user access to the network, enter commands such as the following

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e100-3/1)#dot1x auth-timeout-action failure
```

Syntax: [no] **dot1x auth-timeout-action failure**

Once the failure timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to **retry**.

NOTE

If **restrict-vlan** is configured along with **auth-timeout-action failure**, the user will be placed into a VLAN with restricted or limited access. Refer to [“Allow user access to a restricted VLAN after a RADIUS timeout”](#) on page 1802.

Allow user access to a restricted VLAN after a RADIUS timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and place the user in a VLAN with restricted or limited access, enter commands such as the following

```
Brocade(config)#interface ethernet 3/1
Brocade(config-if-e100-3/1)#dot1x auth-timeout-action failure
```

Syntax: [no] **dot1x auth-timeout-action failure**

NOTE

The commands **auth-fail-action restrict-vlan** and **auth-fail-vlanid** are supported in the global dot1x mode and are not supported at the port-level. The failure action of **dot1x auth-timeout-action failure** will follow the **auth-fail-action** defined at the global dot1x level.

Dynamic VLAN assignment for 802.1X port configuration

When a client successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the Brocade device) a RADIUS Access-Accept message that grants the client access to the network. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies a VLAN identifier, and if this VLAN is available on the Brocade device, the client port is moved from its default VLAN to this specified VLAN.

NOTE

This feature is supported on port-based VLANs only. This feature cannot be used to place an 802.1X-enabled port into a Layer 3 protocol VLAN.

Automatic removal of dynamic VLAN assignments for 802.1X ports

For increased security, this feature removes any association between a port and a dynamically-assigned VLAN when all 802.1x sessions for that VLAN have expired on the port.

NOTE

When a **show run** command is issued during a session, the dynamically-assigned VLAN is not displayed.

Enable 802.1X VLAN ID support by adding the following attributes to a user profile on the RADIUS server.

Attribute name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	<vlan-name> (string) – either the name or the number of a VLAN configured on the Brocade device.

The device reads the attributes as follows:

- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do not have the values specified above, the Brocade device ignores the three Attribute-Value pairs. The client becomes authorized, but the client port is not dynamically placed in a VLAN.
- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do have the values specified above, but there is no value specified for the Tunnel-Private-Group-ID attribute, the client will not become authorized.
- When the Brocade device receives the value specified for the Tunnel-Private-Group-ID attribute, it checks whether the <vlan-name> string matches the name of a VLAN configured on the device. If there is a VLAN on the device whose name matches the <vlan-name> string, then the client port is placed in the VLAN whose ID corresponds to the VLAN name.
- If the <vlan-name> string does not match the name of a VLAN, the Brocade device checks whether the string, when converted to a number, matches the ID of a VLAN configured on the device. If it does, then the client port is placed in the VLAN with that ID.
- If the <vlan-name> string does not match either the name or the ID of a VLAN configured on the device, then the client will not become authorized.

The **show interface** command displays the VLAN to which an 802.1X-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port default VLAN). Refer to [“Displaying dynamically-assigned VLAN information”](#) on page 1825 for sample output indicating the port dynamically assigned VLAN.

Dynamic multiple VLAN assignment for 802.1X ports

When you add attributes to a user profile on the RADIUS server, the <vlan-name> value for the Tunnel-Private-Group-ID attribute can specify the name or number of one or more VLANs configured on the Brocade device.

For example, to specify one VLAN, configure the following for the `<vlan-name>` value in the Tunnel-Private-Group-ID attribute on the RADIUS server.

"10" or "marketing"

In this example, the port on which the Client is authenticated is assigned to VLAN 10 or the VLAN named "marketing". The VLAN to which the port is assigned must have previously been configured on the Brocade device.

Specifying an untagged VLAN

To specify an untagged VLAN, use the following.

"U:10" or "U:marketing"

When the RADIUS server specifies an untagged VLAN ID, the port default VLAN ID (or **PVID**) is changed from the system DEFAULT-VLAN (VLAN 1) to the specified VLAN ID. The port transmits only untagged traffic on its PVID. In this example, the port PVID is changed from VLAN 1 (the DEFAULT-VLAN) to VLAN 10 or the VLAN named "marketing".

The PVID for a port can be changed only once through RADIUS authentication. For example, if RADIUS authentication for a Client causes a port PVID to be changed from 1 to 10, and then RADIUS authentication for another Client on the same port specifies that the port PVID be moved to 20, then the second PVID assignment from the RADIUS server is ignored.

If the link goes down, or the dot1x-mac-session for the Client that caused the initial PVID assignment ages out, then the port reverts back to its original (non-RADIUS-specified) PVID, and subsequent RADIUS authentication can change the PVID assignment for the port.

If a port PVID is assigned through the multi-device port authentication feature, and 802.1X authentication subsequently specifies a different PVID, then the PVID specified through 802.1X authentication overrides the PVID specified through multi-device port authentication.

Specifying a tagged VLAN

To specify a tagged VLAN, use the following.

"T:12;T:20" or "T:12;T:marketing"

In this example, the port is added to VLANs 12 and 20 or VLANs 12 and the VLAN named "marketing". When a tagged packet is authenticated, and a list of VLANs is specified on the RADIUS server for the MAC address, then the packet tag must match one of the VLANs in the list in order for the Client to be successfully authenticated. If authentication is successful, then the port is added to all of the VLANs specified in the list.

Unlike with a RADIUS-specified untagged VLAN, if the dot1x-mac-session for the Client ages out, the port membership in RADIUS-specified tagged VLANs is not changed. In addition, if multi-device port authentication specifies a different list of tagged VLANs, then the port is added to the specified list of VLANs. Membership in the VLANs specified through 802.1X authentication is not changed.

Specifying an untagged VLAN and multiple tagged VLANs

To specify an untagged VLAN and multiple tagged VLANs, use the following.

"U:10;T:12;T:marketing"

When the RADIUS server returns a value specifying both untagged and tagged VLAN IDs, the port becomes a dual-mode port, accepting and transmitting both tagged traffic and untagged traffic at the same time. A dual-mode port transmits only untagged traffic on its default VLAN (PVID) and only tagged traffic on all other VLANs.

In this example, the port VLAN configuration is changed so that it transmits untagged traffic on VLAN 10, and transmits tagged traffic on VLAN 12 and the VLAN named "marketing".

For a configuration example, refer to [“802.1X Authentication with dynamic VLAN assignment”](#) on page 1835.

Saving dynamic VLAN assignments to the running-config file

You can configure the Brocade device to save the RADIUS-specified VLAN assignments to the device's running-config file. Enter commands such as the following.

```
Brocade(config)#dot1x-enable
Brocade(config-dot1x)#save-dynamicvlan-to-config
```

Syntax: save-dynamicvlan-to-config

By default, the dynamic VLAN assignments are not saved to the running-config file. Entering the **show running-config** command does not display dynamic VLAN assignments, although they can be displayed with the **show vlan** and **show authenticated-mac-address detail** commands.

NOTE

When this feature is enabled, issuing the command **write mem** will save any dynamic VLAN assignments to the startup configuration file.

Considerations for dynamic VLAN assignment in an 802.1X multiple-host configuration

The following considerations apply when a Client in a 802.1X multiple-host configuration is successfully authenticated, and the RADIUS Access-Accept message specifies a VLAN for the port:

- If the port is not already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the Brocade device, then the port is placed in that VLAN.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a different VLAN, then it is considered an authentication failure. The port VLAN membership is not changed.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of that same VLAN, then traffic from the Client is forwarded normally.
- If the RADIUS Access-Accept message specifies the name or ID of a VLAN that does not exist on the Brocade device, then it is considered an authentication failure.
- If the port is a tagged or dual-mode port, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the Brocade device, then the port is placed in that VLAN. If the port is already a member of the RADIUS-specified VLAN, no further action is taken.
- If the RADIUS Access-Accept message does not contain any VLAN information, the Client dot1x-mac-session is set to “access-is-allowed”. If the port is already in a RADIUS-specified VLAN, it remains in that VLAN.

Dynamically applying IP ACLs and MAC address filters to 802.1X ports

The Brocade 802.1X implementation supports dynamically applying an IP ACL or MAC address filter to a port, based on information received from an Authentication Server.

When a client/supplicant successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the Brocade device) a RADIUS Access-Accept message that grants the client access to the network. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server.

If the Access-Accept message contains Filter-ID (type 11) or Vendor-Specific (type 26), or both attributes, the Brocade device can use information in these attributes to apply an IP ACL or MAC address filter to the authenticated port. This IP ACL or MAC address filter applies to the port for as long as the client is connected to the network. When the client disconnects from the network, the IP ACL or MAC address filter is no longer applied to the port. If an IP ACL or MAC address filter had been applied to the port prior to 802.1X authentication, it is then re-applied to the port.

The Brocade device uses information in the Filter ID and Vendor-Specific attributes as follows:

- The Filter-ID attribute can specify the number of an existing IP ACL or MAC address filter configured on the Brocade device. In this case, the IP ACL or MAC address filter with the specified number is applied to the port.
- The Vendor-Specific attribute can specify actual syntax for a Brocade IP ACL or MAC address filter, which is then applied to the authenticated port. Configuring a Vendor-Specific attribute in this way allows you to create IP ACLs and MAC address filters that apply to individual users; that is, *per-user* IP ACLs or MAC address filters.

Configuration considerations for applying IP ACLs and MAC address filters to 802.1x ports

The following restrictions apply to dynamic IP ACLs or MAC address filters:

- Inbound dynamic IP ACLs are supported. Outbound dynamic ACLs are not supported.
- Inbound Vendor-Specific attributes are supported. Outbound Vendor-Specific attributes are not supported.
- A maximum of one IP ACL can be configured in the inbound direction on an interface.
- 802.1X with dynamic MAC filter will work for one client at a time on a port. If a second client tries to authenticate with 802.1X and dynamic MAC filter, the second client will be rejected.
- MAC address filters cannot be configured in the outbound direction on an interface.
- Concurrent operation of MAC address filters and IP ACLs is not supported.
- A dynamic IP ACL will take precedence over an IP ACL that is bound to a port (port ACL). When a client authenticates with a dynamic IP ACL, the port ACL will not be applied. Also, future clients on the same port will authenticate with a dynamic IP ACL or no IP ACL. If no clients on the port use dynamic ACL, then the port ACL will be applied to all traffic.

Disabling and enabling strict security mode for dynamic filter assignment

By default, 802.1X dynamic filter assignment operates in **strict security mode**. When strict security mode is enabled, 802.1X authentication for a port fails if the Filter-ID attribute contains invalid information, or if insufficient system resources are available to implement the per-user IP ACLs or MAC address filters specified in the Vendor-Specific attribute.

When strict security mode is enabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the port will not be authenticated, regardless of any other information in the message (for example, if the Tunnel-Private-Group-ID attribute specifies a VLAN on which to assign the port).
- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port will not be authenticated.
- If the device does not have the system resources available to dynamically apply a filter to a port, then the port will not be authenticated.

NOTE

If the Access-Accept message contains values for both the Filter-ID and Vendor-Specific attributes, then the value in the Vendor-Specific attribute (the per-user filter) takes precedence.

Also, if authentication for a port fails because the Filter-ID attribute referred to a non-existent filter, or there were insufficient system resources to implement the filter, then a Syslog message is generated.

Disabled strict security mode

When strict security mode is disabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the port is still authenticated, but no filter is dynamically applied to it.
- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port is still authenticated, but the filter specified in the Vendor-Specific attribute is not applied to the port.

By default, strict security mode is enabled for all 802.1X-enabled interfaces, but you can manually disable or enable it, either globally or for specific interfaces.

Disabling strict security mode globally

To disable strict security mode globally, enter the following commands.

```
Brocade(config)#dot1x-enable
Brocade(config-dot1x)#no global-filter-strict-security
```

After you globally disable strict security mode, you can re-enable it by entering the following command.

```
Brocade(config-dot1x)#global-filter-strict-security
```

Syntax: [no] global-filter-strict-security

To disable strict security mode for a specific interface, enter commands such as the following.

```
Brocade(config)#interface e 1
Brocade(config-if-e1000-1)#dot1x disable-filter-strict-security
```

To re-enable strict security mode for an interface, enter the following command.

```
Brocade(config-if-e1000-1)#no dot1x disable-filter-strict-security
```

Syntax: [no] dot1x disable-filter-strict-security

The output of the **show dot1x** and **show dot1x config** commands has been enhanced to indicate whether strict security mode is enabled or disabled globally and on an interface. Refer to [“Displaying the status of strict security mode”](#) on page 1828.

Dynamically applying existing ACLs or MAC address filters

When a port is authenticated using 802.1X security, an IP ACL or MAC address filter that exists in the running-config on the Brocade device can be dynamically applied to the port. To do this, you configure the Filter-ID (type 11) attribute on the RADIUS server. The Filter-ID attribute specifies the name or number of the Brocade IP ACL or MAC address filter.

The following is the syntax for configuring the Filter-ID attribute to refer to a Brocade IP ACL or MAC address filter.

Value	Description
ip.<number>.in	Applies the specified numbered ACL to the 802.1X authenticated port in the inbound direction.
ip.<name>.in	Applies the specified named ACL to the 802.1X authenticated port in the inbound direction.
mac.<number>.in	Applies the specified numbered MAC address filter to the 802.1X authenticated port in the inbound direction.

The following table lists examples of values you can assign to the Filter-ID attribute on the RADIUS server to refer to IP ACLs and MAC address filters configured on a Brocade device.

Possible values for the filter ID attribute on the RADIUS server	ACL or MAC address filter configured on the Brocade device
ip.2.in	access-list 2 permit host 36.48.0.3 access-list 2 permit 36.0.0.0 0.255.255.255
ip.102.in	access-list 102 permit ip 36.0.0.0 0.255.255.255 any
ip.fdry_filter.in	ip access-list standard fdry_filter permit host 36.48.0.3
mac.2.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800
mac.2.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800
mac.3.in	mac filter 3 permit 2222.2222.2222 ffff.ffff.ffff any etype eq 0800

Notes for dynamically applying ACLs or MAC address filters

- The <name> in the Filter ID attribute is case-sensitive.
- You can specify only numbered MAC address filters in the Filter ID attribute. Named MAC address filters are not supported.
- Dynamic ACL filters are supported only for the inbound direction. Dynamic outbound ACL filters are not supported.
- MAC address filters are supported only for the inbound direction. Outbound MAC address filters are not supported.
- Dynamically assigned IP ACLs and MAC address filters are subject to the same configuration restrictions as non-dynamically assigned IP ACLs and MAC address filters.

Configuring per-user IP ACLs or MAC address filters

Per-user IP ACLs and MAC address filters make use of the Vendor-Specific (type 26) attribute to dynamically apply filters to ports. Defined in the Vendor-Specific attribute are Brocade ACL or MAC address filter statements. When the RADIUS server returns the Access-Accept message granting a client access to the network, the Brocade device reads the statements in the Vendor-Specific attribute and applies these IP ACLs or MAC address filters to the client port. When the client disconnects from the network, the dynamically applied filters are no longer applied to the port. If any filters had been applied to the port previous to the client connecting, then those filters are reapplied to the port.

NOTE

Dynamic IP ACL filters and MAC address filters are not supported on the same port at the same time.

The following table shows the syntax for configuring the Brocade Vendor-Specific attributes with ACL or MAC address filter statements.

Value	Description
ipACL.e.in=<extended-ACL-entries>	Applies the specified extended ACL entries to the 802.1X authenticated port in the inbound direction.
macfilter.in=<mac-filter-entries>	Applies the specified MAC address filter entries to the 802.1X authenticated port in the inbound direction.

The following table shows examples of IP ACLs and MAC address filters configured in the Brocade Vendor-Specific attribute on a RADIUS server. These IP ACLs and MAC address filters follow the same syntax as other Brocade ACLs and MAC address filters. Refer to the related chapters in this book for information on syntax.

ACL or MAC address filter	Vendor-specific attribute on RADIUS server
MAC address filter with one entry	macfilter.in= deny any any
MAC address filter with two entries	macfilter.in= permit 0000.0000.3333 ffff.ffff.0000 any, macfilter.in= permit 0000.0000.4444 ffff.ffff.0000 any

The RADIUS server allows one instance of the Vendor-Specific attribute to be sent in an Access-Accept message.

Enabling 802.1X port security

By default, 802.1X port security is disabled on Brocade devices. To enable the feature on the device and enter the dot1x configuration level, enter the following command.

```
Brocade(config)#dot1x-enable
Brocade(config-dot1x)#
```

Syntax: [no] dot1x-enable

At the dot1x configuration level, you can enable 802.1X port security on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to enable 802.1X port security on all interfaces on the device, enter the following command.

```
Brocade(config-dot1x)#enable all
```

Syntax: [no] enable all

To enable 802.1X port security on interface 3/11, enter the following command.

```
Brocade(config-dot1x)#enable ethernet 3/11
```

Syntax: [no] enable ethernet <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To enable 802.1X port security on interfaces 3/11 through 3/16, enter the following command.

```
Brocade(config-dot1x)#enable ethernet 3/11 to 3/16
```

Syntax: [no] enable ethernet <port> to <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Setting the port control

To activate authentication on an 802.1X-enabled interface, you specify the kind of **port control** to be used on the interface. An interface used with 802.1X port security has two virtual access points: a controlled port and an uncontrolled port:

- The controlled port can be either the authorized or unauthorized state. In the authorized state, it allows normal traffic to pass between the Client and the Authenticator. In the unauthorized state, no traffic is allowed to pass.
- The uncontrolled port allows only EAPOL traffic between the Client and the Authentication Server.

Refer to [Figure 190](#) for an illustration of this concept.

By default, all controlled ports on the device are in the authorized state, allowing all traffic. When you activate authentication on an 802.1X-enabled interface, its controlled port is placed in the unauthorized state. When a Client connected to the interface is successfully authenticated, the controlled port is then placed in the authorized state. The controlled port remains in the authorized state until the Client logs off.

To activate authentication on an 802.1X-enabled interface, you configure the interface to place its controlled port in the authorized state when a Client is authenticated by an Authentication Server. To do this, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-3/1)#dot1x port-control auto
```

Syntax: [no] dot1x port-control [force-authorized | force-unauthorized | auto]

When an interface control type is set to **auto**, the controlled port is initially set to unauthorized, but is changed to authorized when the connecting Client is successfully authenticated by an Authentication Server.

The port control type can be one of the following

force-authorized – The controlled port is placed unconditionally in the authorized state, allowing all traffic. This is the default state for ports on the Brocade device.

force-unauthorized – The controlled port is placed unconditionally in the unauthorized state.

auto – The controlled port is unauthorized until authentication takes place between the Client and Authentication Server. Once the Client passes authentication, the port becomes authorized. This activates authentication on an 802.1X-enabled interface.

NOTE

You cannot enable 802.1X port security on ports that have any of the following features enabled:

- Link aggregation
- Metro Ring Protocol (MRP)
- Mirror port
- Trunk port

Configuring periodic re-authentication

You can configure the device to periodically re-authenticate Clients connected to 802.1X-enabled interfaces. When you enable periodic re-authentication, the device re-authenticates Clients every 3,600 seconds by default. You can optionally specify a different re-authentication interval of between 1 – 4294967295 seconds.

To configure periodic re-authentication using the default interval of 3,600 seconds, enter the following command.

```
Brocade(config-dot1x)#re-authentication
```

Syntax: [no] re-authentication

To configure periodic re-authentication with an interval of 2,000 seconds, enter the following commands.

```
Brocade(config-dot1x)#re-authentication
Brocade(config-dot1x)#timeout re-authperiod 2000
```

Syntax: `[no] timeout re-authperiod <seconds>`

The re-authentication interval is a global setting, applicable to all 802.1X-enabled interfaces. To re-authenticate Clients connected to a specific port manually, use the **dot1x re-authenticate** command. Refer to [“Re-authenticating a port manually”](#), below.

Re-authenticating a port manually

When periodic re-authentication is enabled, by default the Brocade device re-authenticates Clients connected to an 802.1X-enabled interface every 3,600 seconds (or the time specified by the **dot1x timeout re-authperiod** command). You can also manually re-authenticate Clients connected to a specific port.

For example, to re-authenticate Clients connected to interface 3/1, enter the following command.

```
Brocade#dot1x re-authenticate e 3/1
```

Syntax: `dot1x re-authenticate ethernet <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Setting the quiet period

If the Brocade device is unable to authenticate the Client, the Brocade device waits a specified amount of time before trying again. The amount of time the Brocade device waits is specified with the **quiet-period** parameter. The **quiet-period** parameter can be from 1 – 4294967295 seconds. The default is 60 seconds.

For example, to set the quiet period to 30 seconds, enter the following command.

```
Brocade(config-dot1x)#timeout quiet-period 30
```

Syntax: `[no] timeout quiet-period <seconds>`

Specifying the wait interval and number of EAP-request/identity frame retransmissions from the Brocade device

When the Brocade device sends an EAP-request/identity frame to a Client, it expects to receive an EAP-response/identity frame from the Client. By default, if the Brocade device does not receive an EAP-response/identity frame from a Client, the device waits 30 seconds, then retransmits the EAP-request/identity frame. Also by default, the Brocade device retransmits the EAP-request/identity frame a maximum of two times. You can optionally configure the amount of time the device will wait before retransmitting an EAP-request/identity frame, and the number of times the EAP-request/identity frame will be transmitted. This section provides the command syntax for these features.

Setting the wait interval for EAP frame retransmissions

By default, if the Brocade device does not receive an EAP-response/identity frame from a Client, the device waits 30 seconds, then retransmits the EAP-request/identity frame. You can optionally change the amount of time the Brocade device waits before retransmitting the EAP-request/identity frame to the Client.

For example, to cause the Brocade device to wait 60 seconds before retransmitting an EAP-request/identity frame to a Client, enter the following command.

```
Brocade(config-dot1x)#timeout tx-period 60
```

If the Client does not send back an EAP-response/identity frame within 60 seconds, the device will transmit another EAP-request/identity frame.

Syntax: [no] timeout tx-period <seconds>

where <seconds> is a value from 1 – 4294967295. The default is 30 seconds.

Setting the maximum number of EAP frame retransmissions

The Brocade device retransmits the EAP-request/identity frame a maximum of two times. If no EAP-response/identity frame is received from the Client after two EAP-request/identity frame retransmissions (or the amount of time specified with the **auth-max** command), the device restarts the authentication process with the Client.

You can optionally change the number of times the Brocade device should retransmit the EAP-request/identity frame. You can specify between 1 – 10 frame retransmissions. For example, to configure the device to retransmit an EAP-request/identity frame to a Client a maximum of three times, enter the following command:

```
Brocade(config-dot1x)#auth-max 3
```

Syntax: auth-max <value>

<value> is a number from 1 – 10. The default is 2.

Wait interval and number of EAP-request/identity frame retransmissions from the RADIUS server

Acting as an intermediary between the RADIUS Authentication Server and the Client, the Brocade device receives RADIUS messages from the RADIUS server, encapsulates them as EAPOL frames, and sends them to the Client. By default, when the Brocade device relays an EAP-Request frame from the RADIUS server to the Client, it expects to receive a response from the Client within 30 seconds. If the Client does not respond within the allotted time, the device retransmits the EAP-Request frame to the Client. Also by default, the Brocade device retransmits the EAP-request frame twice. If no EAP-response frame is received from the Client after two EAP-request frame retransmissions, the device restarts the authentication process with the Client.

You can optionally configure the amount of time the device will wait before retransmitting an EAP-request/identity frame, and the number of times the EAP-request/identity frame will be transmitted. This section provides the command syntax for these features.

Setting the wait interval for EAP frame retransmissions

By default, when the Brocade device relays an EAP-Request frame from the RADIUS server to the Client, it expects to receive a response from the Client within 30 seconds. You can optionally specify the wait interval using the **supptimeout** command.

For example, to configure the device to retransmit an EAP-Request frame if the Client does not respond within 45 seconds, enter the following command.

```
Brocade(config-dot1x)#supptimeout 45
```

Syntax: **supptimeout** <seconds>

<seconds> is a number from 1 – 4294967295 seconds. The default is 30 seconds.

Setting the maximum number of EAP frame retransmissions

You can optionally specify the number of times the Brocade device will retransmit the EAP-request frame. You can specify between 1 – 10 frame retransmissions. For example, to configure the device to retransmit an EAP-request frame to a Client a maximum of three times, enter the following command.

```
Brocade(config-dot1x)#maxreq 3
```

Syntax: **maxreq** <value>

<value> is a number from 1 – 10. The default is 2.

Specifying a timeout for retransmission of messages to the authentication server

When performing authentication, the Brocade device receives EAPOL frames from the Client and passes the messages on to the RADIUS server. The device expects a response from the RADIUS server within 30 seconds. If the RADIUS server does not send a response within 30 seconds, the Brocade device retransmits the message to the RADIUS server. The time constraint for retransmission of messages to the Authentication Server can be between 0 – 4294967295 seconds.

For example, to configure the device to retransmit a message if the Authentication Server does not respond within 45 seconds, enter the following command.

```
Brocade(config-dot1x)#servertimeout 45
```

Syntax: **servertimeout** <seconds>

Initializing 802.1X on a port

To initialize 802.1X port security on a port, enter a command such as the following.

```
Brocade#dot1x initialize e 3/1
```

Syntax: **dot1x initialize ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*

- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Allowing access to multiple hosts

Brocade devices support 802.1X authentication for ports with more than one host connected to them. If there are multiple hosts connected to a single 802.1X-enabled port, the Brocade device authenticates each of them individually. Refer to [“Configuring 802.1X multiple-host authentication”](#) on page 1815.

Configuring 802.1X multiple-host authentication

When multiple hosts are connected to the same 802.1X-enabled port, the functionality described in [“How 802.1X multiple-host authentication works”](#) on page 1796 is enabled by default. You can optionally do the following:

- Specify the authentication-failure action
- Specify the number of authentication attempts the device makes before dropping packets
- Disabling aging for dot1x-mac-sessions
- Configure aging time for blocked clients
- Moving native VLAN mac-sessions to restrict VLAN
- Clear the dot1x-mac-session for a MAC address

Specifying the authentication-failure action

In an 802.1X multiple-host configuration, if RADIUS authentication for a client is unsuccessful, either traffic from that client is dropped in hardware (the default), or the client port is placed in a “restricted” VLAN. You can specify which of these authentication-failure actions to use. When you enable 802.1X, the default authentication-failure action is to drop client traffic.

If you configure the authentication-failure action to place the client port in a restricted VLAN, you can specify the ID of the restricted VLAN. If you do not specify a VLAN ID, the default VLAN is used.

You can configure the authentication-failure action using one of the following methods:

- Configure the same authentication-failure action for all ports on the device (globally).
- Configure an authentication-failure action on individual ports.

NOTE

You cannot configure the authentication-failure action globally and per-port at the same time.

To configure the authentication-failure action for all ports on the device to place the client port in a restricted VLAN, enter the following commands.

```
Brocade(config)# dot1x-enable
Brocade(config-dot1x)#auth-fail-action restricted-vlan
```

Syntax: [no] auth-fail-action restricted-vlan

To specify VLAN 300 as the restricted VLAN for all ports on the device, enter the **auth-fail-vlanid** *<num>* command.

```
Brocade(config-dot1x)# auth-fail-vlanid 300
```

Syntax: [no] auth-fail-vlanid *<vlan-id>*

To specify on an individual port that the authentication-failure action is to place the client port in restricted VLAN 300, enter the following command at the interface configuration level.

```
Brocade(config-if-e1000-1/1/1)# dot1x auth-fail-action restrict-vlan 300
```

Syntax: [no] dot1x auth-fail-action restrict-vlan <vlan-id>

Specifying the number of authentication attempts the device makes before dropping packets

When the authentication-failure action is to drop traffic from the Client, and the initial authentication attempt made by the device to authenticate the Client is unsuccessful, the Brocade device immediately retries to authenticate the Client. After three unsuccessful authentication attempts, the Client dot1x-mac-session is set to “access-denied”, causing traffic from the Client to be dropped in hardware.

Optionally, you can configure the number of authentication attempts the device makes before dropping traffic from the Client. To do so, enter a command such as the following.

```
Brocade(config-dot1x)# auth-fail-max-attempts 2
```

Syntax: [no] auth-fail-max-attempts <attempts>

By default, the device makes three attempts to authenticate a Client before dropping packets from the Client. You can specify from 1 through 10 authentication attempts.

Disabling aging for dot1x-mac-sessions

The dot1x-mac-sessions for Clients authenticated or denied by a RADIUS server are aged out if no traffic is received from the Client MAC address for a certain period of time. After a Client dot1x-mac-session is aged out, the Client must be re-authenticated:

- **Permitted** dot1x-mac-sessions, which are the dot1x-mac-sessions for authenticated Clients, as well as for non-authenticated Clients whose ports have been placed in the restricted VLAN, are aged out if no traffic is received from the Client MAC address over the normal MAC aging interval on the Brocade device.
- **Denied** dot1x-mac-sessions, which are the dot1x-mac-sessions for non-authenticated Clients that are blocked by the Brocade device are aged out over a configurable software aging period. (Refer to the next section for more information on configuring the software aging period).

You can optionally disable aging of the permitted or denied dot1x-mac-sessions, or both, on the Brocade device.

To disable aging of the permitted dot1x-mac-sessions, enter the following command.

```
Brocade(config-dot1x)#mac-session-aging no-aging permitted-mac-only
```

Syntax: [no] mac-session-aging no-aging permitted-mac-only

To disable aging of the denied dot1x-mac-sessions, enter the following command.

```
Brocade(config-dot1x)#mac-session-aging no-aging denied-mac-only
```

Syntax: [no] mac-session-aging no-aging denied-mac-only

NOTE

This command enables aging of permitted sessions.

As a shortcut, use the command [no] **mac-session-aging** to enable or disable aging for permitted and denied sessions.

Specifying the aging time for blocked clients

When the Brocade device is configured to drop traffic from non-authenticated Clients, traffic from the blocked Clients is dropped in hardware, without being sent to the CPU. A Layer 2 CAM entry is created that drops traffic from the blocked Client MAC address in hardware. If no traffic is received from the blocked Client MAC address for a certain amount of time, this Layer 2 CAM entry is aged out. If traffic is subsequently received from the Client MAC address, then an attempt can be made to authenticate the Client again.

Aging of the Layer 2 CAM entry for a blocked Client MAC address occurs in two phases, known as **hardware aging** and **software aging**. The hardware aging period is fixed at 70 seconds and is non-configurable. The software aging time is configurable through the CLI.

Once the Brocade device stops receiving traffic from a blocked Client MAC address, the hardware aging begins and lasts for a fixed period of time. After the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (by default 120 seconds). After the software aging period ends, the blocked Client MAC address ages out, and can be authenticated again if the Brocade device receives traffic from the Client MAC address.

Change the length of the software aging period for a blocked Client MAC address by entering the **mac-age-time** *<num>* command.

```
Brocade(config)#mac-age-time 180
```

Syntax: [no] **mac-age-time** *<seconds>*

You can specify from 1 – 65535 seconds. The default is 120 seconds.

Moving native VLAN mac-sessions to restrict VLAN

You can move the native VLAN mac-sessions to restrict VLAN on authentication failure. You can configure the option of overriding the dual-mode port native untagged VLAN with restricted VLAN in case 802.1x authentication fails and there is no RADIUS assigned VLAN. Use this command when you configure multi-device port authentication and 802.1X authentication configuration with dynamic VLAN assignment from RADIUS Server on the same port.

Example

```
Brocade(config-dot1x)# auth-fail-force-restrict
```

Syntax: [no] **auth-fail-force-restrict**

Clearing a dot1x-mac-session for a MAC address

You can clear the dot1x-mac-session for a specified MAC address, so that the Client with that MAC address can be re-authenticated by the RADIUS server.

Example

```
Brocade#clear dot1x mac-session 00e0.1234.abd4
```

Syntax: **clear dot1x mac-session** *<mac-address>*

MAC address filters for EAP frames

You can create MAC address filters to permit or deny EAP frames. To do this, you specify the Brocade device 802.1X group MAC address as the destination address in a MAC address filter, then apply the filter to an interface.

Creating MAC address filters for EAPS on most devices

For example, the following command creates a MAC address filter that denies frames with the destination MAC address of 0180.c200.0003, which is the 802.1X group MAC address on the Brocade device.

```
Brocade(config)#mac filter 1 deny any 0180.c200.0003 ffff.ffff.ffff
```

The following commands apply this filter to interface e 3/1.

```
Brocade(config)#interface e 3/11
Brocade(config-if-3/1)#mac filter-group 1
```

Refer to [“Defining MAC address filters”](#) on page 1857 for more information.

Configuring VLAN access for non-EAP-capable clients

You can configure the Brocade device to grant "guest" or restricted VLAN access to clients that do not support Extensible EAP. The restricted VLAN limits access to the network or applications, instead of blocking access to these services altogether.

When the Brocade device receives the first packet (non-EAP packet) from a client, the device waits for 10 seconds or the amount of time specified with the **timeout restrict-fwd-period** command. If the Brocade device does not receive subsequent packets after the timeout period, the device places the client on the restricted VLAN.

This feature is disabled by default. To enable this feature and change the timeout period, enter commands such as the following.

```
Brocade(config)#dot1x-enable
Brocade(config-dot1x)#restrict-forward-non-dot1x
Brocade(config-dot1x)#timeout restrict-fwd-period 15
```

Once the *success* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

Syntax: **timeout restrict-fwd-period** <num>

The <num> parameter is a value from 0 to 4294967295. The default value is 10.

802.1X accounting configuration

802.1X accounting enables the recording of information about 802.1X clients who were successfully authenticated and allowed access to the network. When 802.1X accounting is enabled on the Brocade device, it sends the following information to a RADIUS server whenever an authenticated 802.1X client (user) logs into or out of the Brocade device:

- The user name
- The session ID
- The user MAC address
- The authenticating physical port number

An Accounting Start packet is sent to the RADIUS server when a user is successfully authenticated. The Start packet indicates the start of a new session and contains the user MAC address and physical port number. The 802.1X session state will change to Authenticated and Permit after receiving a response from the accounting server for the accounting Start packet. If the Accounting service is not available, the 802.1X session status will change to Authenticated and Permit after a RADIUS timeout. The device will retry authentication requests three times (the default), or the number of times configured on the device.

An **Accounting Stop packet** is sent to the RADIUS server when one of the following events occur:

- The user logs off
- The port goes down
- The port is disabled
- The user fails to re-authenticate after a RADIUS timeout
- The 802.1X port control-auto configuration changes
- The MAC session clears (through use of the **clear dot1x mac-session** CLI command)

The Accounting Stop packet indicates the end of the session and the time the user logged out.

802.1X Accounting attributes for RADIUS

Brocade devices support the following RADIUS attributes for 802.1X accounting.

TABLE 295 802.1X accounting attributes for RADIUS

Attribute name	Attribute ID	Data Type	Description
Acct-Session-ID	44	Integer	The account session ID, which is a number from 1 to 4294967295.
Acct-Status-Type	40	integer	Indicates whether the accounting request marks the beginning (start) or end (stop) of the user service. 1 – Start 2 – Stop
Calling-Station-Id	31	string	The supplicant MAC address in ASCII format (upper case only), with octet values separated by a dash (-). For example 00-10-A4-23-19-C0
NAS-Port	5	integer	The physical port number.
NAS-Port-Type	61	integer	The physical port type.
user-name	1	string	The user name.

Enabling 802.1X accounting

To enable 802.1X accounting, enter the following command.

```
Brocade(config)#aaa accounting dot1x default start-stop radius none
```

Syntax: **aaa accounting dot1x default start-stop radius | none**

radius – Use the list of all RADIUS servers that support 802.1X for authentication.

none – Use no authentication. The client is automatically authenticated without the device using information supplied by the client.

NOTE

If you specify both **radius** and **none**, make sure **radius** comes before **none**.

Displaying 802.1X information

You can display the following 802.1X-related information:

- The 802.1X configuration on the device and on individual ports
- Statistics about the EAPOL frames passing through the device
- 802.1X-enabled ports dynamically assigned to a VLAN
- User-defined and dynamically applied MAC address filters and IP ACLs currently active on the device
- The 802.1X multiple-host configuration

Displaying 802.1X configuration information

To display information about the 802.1X configuration on the Brocade device, enter the **show dot1x** command.

```
Brocade#show dot1x
PAE Capability:    Authenticator Only
system-auth-control: Enable
re-authentication: Disable
global-filter-strict-security: Enable
quiet-period:     60 Seconds
tx-period:        30 Seconds
supertimeout:     30 Seconds
servertimeout:    30 Seconds
maxreq:           2
re-authperiod:    3600 Seconds
Protocol Version: 1
```

Syntax: show dot1x

The following table describes the information displayed by the **show dot1x** command.

TABLE 296 Output from the **show dot1x** command

Field	Description
PAE Capability	The Port Access Entity (PAE) role for the Brocade device. This is always "Authenticator Only".
system-auth-control	Whether system authentication control is enabled on the device. The dot1x-enable command enables system authentication control on the device.
re-authentication	Whether periodic re-authentication is enabled on the device. Refer to "Configuring periodic re-authentication" on page 1811. When periodic re-authentication is enabled, the device automatically re-authenticates Clients every 3,600 seconds by default.
global-filter-strict-security	Whether strict security mode is enabled or disabled globally. Refer to "Disabling and enabling strict security mode for dynamic filter assignment" on page 1807.
quiet-period	When the Brocade device is unable to authenticate a Client, the amount of time the Brocade device waits before trying again (default 60 seconds). Refer to "Setting the quiet period" on page 1812 for information on how to change this setting.

TABLE 296 Output from the **show dot1x** command (Continued)

Field	Description
tx-period	When a Client does not send back an EAP-response/identity frame, the amount of time the Brocade device waits before retransmitting the EAP-request/identity frame to a Client (default 30 seconds). Refer to “Setting the wait interval for EAP frame retransmissions” on page 1813 for information on how to change this setting.
supp-timeout	When a Client does not respond to an EAP-request frame, the amount of time before the Brocade device retransmits the frame. Refer to “Setting the wait interval for EAP frame retransmissions” on page 1814 for information on how to change this setting.
server-timeout	When the Authentication Server does not respond to a message sent from the Client, the amount of time before the Brocade device retransmits the message. Refer to “Specifying a timeout for retransmission of messages to the authentication server” on page 1814 for information on how to change this setting.
maxreq	The number of times the Brocade device retransmits an EAP-request/identity frame if it does not receive an EAP-response/identity frame from a Client (default 2 times). Refer to “Setting the maximum number of EAP frame retransmissions” on page 1813 for information on how to change this setting.
re-authperiod	How often the device automatically re-authenticates Clients when periodic re-authentication is enabled (default 3,600 seconds). Refer to “Configuring periodic re-authentication” on page 1811 for information on how to change this setting.
Protocol Version	The version of the 802.1X protocol in use on the device.

To display information about the 802.1X configuration on an individual port, enter the **show dot1x configuration ethernet** command.

```
Brocade#show dot1x configuration ethernet 1/3
Port-Control                : control-auto
filter strict security      : Enable
Action on RADIUS timeout   : Treat as a failed authentication
  re-authenticate          : 150 seconds
PVID State                  : Normal (101)
Original PVID               : 101
PVID mac total              : 1
PVID mac authorized         : 1
num mac sessions            : 1
num mac authorized          : 1
Number of Auth filter       : 0
```

Syntax: **show dot1x config ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The following additional information is displayed in the **show dot1x config** command for an interface.

TABLE 297 Output from the **show dot1x config** command for an interface

Field	Description
Authenticator PAE state	<p>The current status of the Authenticator PAE state machine. This can be INITIALIZE, DISCONNECTED, CONNECTING, AUTHENTICATING, AUTHENTICATED, ABORTING, HELD, FORCE_AUTH, or FORCE_UNAUTH.</p> <p>NOTE: When the Authenticator PAE state machine is in the AUTHENTICATING state, if the reAuthenticate, eapStart, eapLogoff, or authTimeout parameters are set to TRUE, it may place the Authenticator PAE state machine indefinitely in the ABORTING state. If this should happen, use the dot1x initialize command to initialize 802.1X port security on the port, or unplug the Client or hub connected to the port, then reconnect it.</p>
Backend Authentication state	The current status of the Backend Authentication state machine. This can be REQUEST, RESPONSE, SUCCESS, FAIL, TIMEOUT, IDLE, or INITIALIZE.
AdminControlledDirections	Indicates whether an unauthorized controlled port exerts control over communication in both directions (disabling both reception of incoming frames and transmission of outgoing frames), or just in the incoming direction (disabling only reception of incoming frames). On Brocade devices, this parameter is set to BOTH.
OperControlledDirections	The setting for the OperControlledDirections parameter, as defined in the 802.1X standard. According to the 802.1X standard, if the AdminControlledDirections parameter is set to BOTH, the OperControlledDirections parameter is unconditionally set to BOTH. Since the AdminControlledDirections parameter on Brocade devices is always set to BOTH, the OperControlledDirections parameter is also set to BOTH.
AuthControlledPortControl	The port control type configured for the interface. If set to auto, authentication is activated on the 802.1X-enabled interface.
AuthControlledPortStatus	The current status of the interface controlled port either authorized or unauthorized.
multiple-hosts	Whether the port is configured to allow multiple Supplicants accessing the interface on the Brocade device through a hub. Refer to “Allowing access to multiple hosts” on page 1815 for information on how to change this setting.

Displaying 802.1X statistics

To display 802.1X statistics for an individual port, enter the **show dot1x statistics** command.

```
Brocade#show dot1x statistics e 3/3
```

```
Port 3/3 Statistics:
RX EAPOL Start:      0
RX EAPOL Logoff:     0
RX EAPOL Invalid:    0
RX EAPOL Total:      0
RX EAP Resp/Id:      0
RX EAP Resp other than Resp/Id:  0
RX EAP Length Error:  0
Last EAPOL Version:  0
Last EAPOL Source:   0007.9550.0B83
TX EAPOL Total:      217
TX EAP Req/Id:       163
TX EAP Req other than Req/Id:    0
```

Syntax: **show dot1x statistics ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The following table describes the information displayed by the **show dot1x statistics** command for an interface.

TABLE 298 Output from the **show dot1x statistics** command

Field	Statistics
RX EAPOL Start	The number of EAPOL-Start frames received on the port.
RX EAPOL Logoff	The number of EAPOL-Logoff frames received on the port.
RX EAPOL Invalid	The number of invalid EAPOL frames received on the port.
RX EAPOL Total	The total number of EAPOL frames received on the port.
RX EAP Resp/Id	The number of EAP-Response/Identity frames received on the port
RX EAP Resp other than Resp/Id	The total number of EAPOL-Response frames received on the port that were not EAP-Response/Identity frames.
RX EAP Length Error	The number of EAPOL frames received on the port that have an invalid packet body length.
Last EAPOL Version	The version number of the last EAPOL frame received on the port.
Last EAPOL Source	The source MAC address in the last EAPOL frame received on the port.
TX EAPOL Total	The total number of EAPOL frames transmitted on the port.
TX EAP Req/Id	The number of EAP-Request/Identity frames transmitted on the port.
TX EAP Req other than Req/Id	The number of EAP-Request frames transmitted on the port that were not EAP-Request/Identity frames.

Clearing 802.1X statistics

You can clear the 802.1X statistics counters on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to clear the 802.1X statistics counters on all interfaces on the device, enter the **clear dot1x statistics all** command.

```
Brocade#clear dot1x statistics all
```

Syntax: **clear dot1x statistics all**

To clear the 802.1X statistics counters on interface e 3/11, enter the following command.

```
Brocade#clear dot1x statistics e 3/11
```

Syntax: **clear dot1x statistics ethernet <port>**

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying dynamically-assigned VLAN information

The **show interface** command displays the VLAN to which an 802.1X-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port default VLAN).

The following example of the **show interface** command indicates the port dynamically assigned VLAN. Information about the dynamically assigned VLAN is shown in bold type.

```
Brocade#show interface e 12/2
FastEthernet12/2 is up, line protocol is up
  Hardware is FastEthernet, address is 0204.80a0.4681 (bia 0204.80a0.4681)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Member of L2 VLAN ID 2 (dot1x-RADIUS assigned), original L2 VLAN ID is 1,
  port is untagged, port state is FORWARDING
  STP configured to ON, priority is level0, flow control enabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1518 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 256 bits/sec, 0 packets/sec, 0.00% utilization
  3 packets input, 192 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 3 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 3 packets
  919 packets output, 58816 bytes, 0 underruns
  Transmitted 1 broadcasts, 916 multicasts, 2 unicasts
  0 output errors, 0 collisions, DMA transmitted 919 packets
```

In this example, the 802.1X-enabled port has been moved from VLAN 1 to VLAN 2. When the client disconnects, the port will be moved back to VLAN 1.

The **show run** command also indicates the VLAN to which the port has been dynamically assigned. The output can differ depending on whether GARP VLAN Registration Protocol (GVRP) is enabled on the device:

- **Without GVRP** – When you enter the **show run** command, the output indicates that the port is a member of the VLAN to which it was dynamically assigned through 802.1X. If you then enter the **write memory** command, the VLAN to which the port is currently assigned becomes the port default VLAN in the device configuration.
- **With GVRP** – When you enter the **show run** command, if the VLAN name supplied by the RADIUS server corresponds to a VLAN learned through GVRP, then the output indicates that the port is a member of the VLAN to which it was originally assigned (not the VLAN to which it was dynamically assigned).

If the VLAN name supplied by the RADIUS server corresponds to a statically configured VLAN, the output indicates that the port is a member of the VLAN to which it was dynamically assigned through 802.1X. If you then enter the **write memory** command, the VLAN to which the port is currently assigned becomes the port default VLAN in the device configuration.

Displaying information about dynamically applied MAC address filters and IP ACLs

You can display information about currently active user-defined and dynamically applied MAC address filters and IP ACLs.

Displaying user-defined MAC address filters and IP ACLs

To display the user-defined MAC address filters active on the device, enter the following command.

```
Brocade#show dot1x mac-address filter
```

```
Port 1/3 (User defined MAC Address Filter) :  
    mac filter 1 permit any any
```

Syntax: show dot1x mac-address-filter

To display the user-defined IP ACLs active on the device, enter the **show dot1x ip-ACL** command.

```
Brocade#show dot1x ip-ACL
```

```
Port 1/3 (User defined IP ACLs):
```

```
Extended IP access list Port_1/3_E_IN  
permit udp any any
```

```
Extended IP access list Port_1/3_E_OUT  
permit udp any any
```

Syntax: show dot1x ip-ACL

Displaying dynamically applied MAC address filters and IP ACLs

To display the dynamically applied MAC address filters active on an interface, enter a command such as the following.

```
Brocade#show dot1x mac-address-filter e 1/3
```

```
Port 1/3 MAC Address Filter information:
  802.1X Dynamic MAC Address Filter :
    mac filter-group 2
  Port default MAC Address Filter:
    No mac address filter is set
```

Syntax: `show dot1x mac-address-filter all | ethernet <port>`

The **all** keyword displays all dynamically applied MAC address filters active on the device.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To display the dynamically applied IP ACLs active on an interface, enter a command such as the following.

```
Brocade#show dot1x ip-ACL e 1/3
```

```
Port 1/3 IP ACL information:
  802.1X dynamic IP ACL (user defined) in:
    ip access-list extended Port_1/3_E_IN in
  Port default IP ACL in:
    No inbound ip access-list is set
  802.1X dynamic IP ACL (user defined) out:
    ip access-list extended Port_1/3_E_OUT out
  Port default IP ACL out:
    No outbound ip access-list is set
```

Syntax: `show dot1x ip-ACL all | ethernet <port>`

The **all** keyword displays all dynamically applied IP ACLs active on the device.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying the status of strict security mode

The output of the **show dot1x** and **show dot1x config** commands indicate whether strict security mode is enabled or disabled globally and on an interface.

Displaying the status of strict security mode globally on the device

To display the status of strict security mode globally on the device, enter the **show dot1x** command.

```
Brocade#show dot1x
PAE Capability:      Authenticator Only
system-auth-control: Enable
re-authentication: Disable
global-filter-strict-security: Enable
quiet-period:       60 Seconds
tx-period:          30 Seconds
supertimeout:       30 Seconds
servertimeout:      30 Seconds
maxreq:             2
re-authperiod:      3600 Seconds
security-hold-time: 60 Seconds
Protocol Version:   1
```

Syntax: show dot1x

Displaying the status of strict security mode on an interface

To display the status of strict security mode on an interface, enter a command such as the following

```
Brocade#show dot1x config e 1/3

Port 1/3 Configuration:
Authenticator PAE state:      AUTHENTICATED
Backend Authentication state: IDLE
AdminControlledDirections:    BOTH
OperControlledDirections:     BOTH
AuthControlledPortControl:    Auto
AuthControlledPortStatus:     authorized
quiet-period:                 60 Seconds
tx-period:                     30 Seconds
supertimeout:                 30 Seconds
servertimeout:                30 Seconds
maxreq:                        2
re-authperiod:                3600 Seconds
security-hold-time:           60 Seconds
re-authentication:            Disable
multiple-hosts:               Disable
filter-strict-security: Enable
Protocol Version:             1
```

Syntax: show dot1x config ethernet <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*

- FESX compact switches – *portnum*

Displaying 802.1X multiple-host authentication information

You can display the following information about 802.1X multiple-host authentication:

- Information about the 802.1X multiple-host configuration
- The dot1x-mac-sessions on each port
- The number of users connected on each port in a 802.1X multiple-host configuration

Displaying 802.1X multiple-host configuration information

The output of the **show dot1x** and **show dot1x config** commands displays information related to 802.1X multiple-host authentication.

The following is an example of the output of the **show dot1x** command. The information related to multiple-host authentication is highlighted in bold.

```
Brocade#show dot1x
```

```

Number of Ports enabled      : 2
Re-Authentication           : Enabled
Authentication-fail-action : Restricted VLAN
Authentication Failure VLAN : 111
Mac Session Aging          : Disabled for permitted MAC sessions
Mac Session max-age        : 60 seconds
Protocol Version            : 1
quiet-period                 : 5 Seconds
tx-period                    : 30 Seconds
supptimeout                  : 30 Seconds
servertimeout                : 30 Seconds
maxreq                       : 2
re-authperiod                : 3600 Seconds
security-hold-time           : 60 Seconds
re-authentication            : Enable
Flow based multi-user policy : Disable
```

Syntax: **show dot1x**

[Table 299](#) describes the bold fields in the display.

TABLE 299 Output from the **show dot1x** command for multiple host authentication

Field	Description
Authentication-fail-action	The configured authentication-failure action. This can be Restricted VLAN or Block Traffic.
Authentication Failure VLAN	If the authentication-failure action is Restricted VLAN, the ID of the VLAN to which unsuccessfully authenticated Client ports are assigned.
Mac Session Aging	Whether aging for dot1x-mac-sessions has been enabled or disabled for permitted or denied dot1x-mac-sessions.

TABLE 299 Output from the **show dot1x** command for multiple host authentication (Continued)

Field	Description
Mac Session max-age	The configured software aging time for dot1x-mac-sessions.
Flow based multi-user policy	The dynamically assigned IP ACLs and MAC address filters used in the 802.1X multiple-host configuration.

The output of the **show dot1x config** command for an interface displays the configured port control for the interface. This command also displays information related to 802.1X multiple host-authentication.

The following is an example of the output of the **show dot1x config** command for an interface.

```
Brocade#show dot1x config e 3/1

Port-Control                : control-auto
filter strict security      : Enable
PVID State                  : Restricted (10)
Original PVID               : 10
PVID mac total              : 1
PVID mac authorized         : 0
num mac sessions            : 1
num mac authorized          : 0
```

Syntax: **show dot1x config ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The following table lists the fields in the display.

TABLE 300 Output from the **show dot1x config** command

Field	Description
Port-Control	The configured port control type for the interface. This can be one of the following: force-authorized – The controlled port is placed unconditionally in the authorized state, allowing all traffic. This is the default state for ports on the Brocade device. force-unauthorized – The controlled port is placed unconditionally in the unauthorized state. No authentication takes place for any connected 802.1X Clients. auto – The authentication status for each 802.1X Client depends on the authentication status returned from the RADIUS server.
filter strict security	Whether strict security mode is enabled or disabled on the interface.
PVID State	The port default VLAN ID (PVID) and the state of the port PVID. The PVID state can be one of the following Normal – The port PVID is not set by a RADIUS server, nor is it the restricted VLAN. RADIUS – The port PVID was dynamically assigned by a RADIUS server. RESTRICTED – The port PVID is the restricted VLAN.
Original PVID	The originally configured (not dynamically assigned) PVID for the port.
PVID mac total	The number of devices transmitting untagged traffic on the port PVID.

TABLE 300 Output from the **show dot1x config** command (Continued)

Field	Description
PVID mac authorized	The number of devices transmitting untagged traffic on the port PVID as a result of dynamic VLAN assignment.
num mac sessions	The number of dot1x-mac-sessions on the port.
num mac authorized	The number of authorized dot1x-mac-sessions on the port.

Displaying information about the dot1x MAC sessions on each port

The **show dot1x mac-session** command displays information about the dot1x-mac-sessions on each port on the device. The output also shows the authenticator PAE state.

Example

```
Brocade#show dot1x mac-session
```

Port	MAC/(username)	Vlan	Auth State	ACL	Age	PAE State
1	0010.a498.24f7 :User	10	permit	none	S20	AUTHENTICATED

Syntax: **show dot1x mac-session**

[Table 301](#) lists the new fields in the display.

TABLE 301 Output from the **show dot1x mac-session** command

Field	Description
Port	The port on which the dot1x-mac-session exists.
MAC/ (username)	The MAC address of the Client and the username used for RADIUS authentication.
Vlan	The VLAN to which the port is currently assigned.
Auth-State	The authentication state of the dot1x-mac-session. This can be one of the following permit – The Client has been successfully authenticated, and traffic from the Client is being forwarded normally. blocked – Authentication failed for the Client, and traffic from the Client is being dropped in hardware. restricted – Authentication failed for the Client, but traffic from the Client is allowed in the restricted VLAN only. init - The Client is in is in the process of 802.1X authentication, or has not started the authentication process.
Age	The software age of the dot1x-mac-session.
PAE State	The current status of the Authenticator PAE state machine. This can be INITIALIZE, DISCONNECTED, CONNECTING, AUTHENTICATING, AUTHENTICATED, ABORTING, HELD, FORCE_AUTH, or FORCE_UNAUTH. NOTE: When the Authenticator PAE state machine is in the AUTHENTICATING state, if the reAuthenticate, eapStart, eapLogoff, or authTimeout parameters are set to TRUE, it may place the Authenticator PAE state machine indefinitely in the ABORTING state. If this should happen, use the dot1x initialize command to initialize 802.1X port security on the port, or unplug the Client or hub connected to the port, then reconnect it.

Displaying information about the ports in an 802.1X multiple-host configuration

To display information about the ports in an 802.1X multiple-host configuration, enter the **sho do mac-s br** command.

```
Brocade(config-dot1x)#sh do mac-s br
```

Port	Number of users	Number of Authorized users	Dynamic VLAN	Dynamic ACL	Dynamic MAC-Filt
1/1/1	0	0	no	no	no
1/1/2	0	0	no	no	no
1/1/3	0	0	no	no	no
1/1/4	0	0	no	no	no
1/1/5	0	0	no	no	no
1/1/6	0	0	no	no	no
1/1/7	0	0	no	no	no
1/1/8	0	0	no	no	no
1/1/9	0	0	no	no	no
1/1/10	0	0	no	no	no
1/1/11	0	0	no	no	no
1/1/12	0	0	no	no	no
1/1/13	0	0	no	no	no
1/1/14	0	0	no	no	no
1/1/15	0	0	no	no	no
1/1/16	0	0	no	no	no

Syntax: show dot1x mac-session brief

The following table describes the information displayed by the **show dot1x mac-session brief** command.

TABLE 302 Output from the **show dot1x mac-session brief** command

Field	Description
Port	Information about the users connected to each port.
Number of users	The number of users connected to the port.
Number of Authorized users	The number of users connected to the port that have been successfully authenticated.
Dynamic VLAN	Whether the port is a member of a RADIUS-specified VLAN.
Dynamic Filters	Whether RADIUS-specified IP ACLs or MAC address filters have been applied to the port.

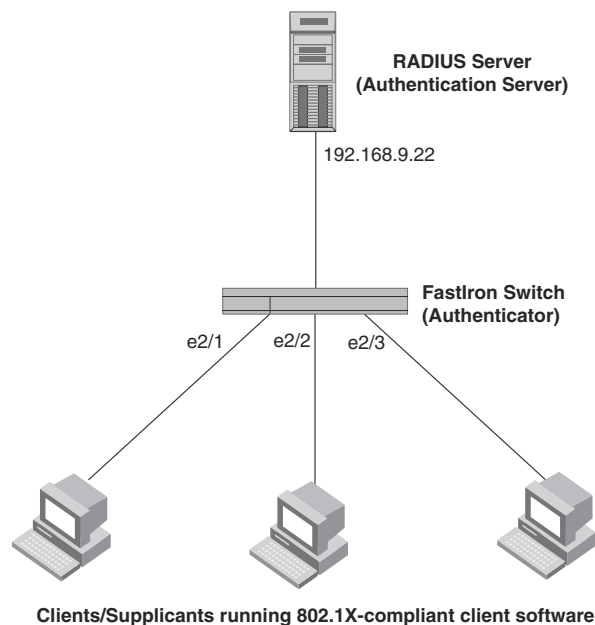
Sample 802.1X configurations

This section illustrates a sample point-to-point configuration and a sample hub configuration that use 802.1X port security.

Point-to-point configuration

[Figure 193](#) illustrates a sample 802.1X configuration with Clients connected to three ports on the Brocade device. In a point-to-point configuration, only one 802.1X Client can be connected to each port.

FIGURE 193 Sample point-to-point 802.1X configuration



Same point-to-point 802.1x configuration

The following commands configure the Brocade device in [Figure 193](#)

```

Brocade(config)#aaa authentication dot1x default radius
Brocade(config)#radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
Brocade(config)#dot1x-enable e 1 to 3
Brocade(config-dot1x)#re-authentication
Brocade(config-dot1x)#timeout re-authperiod 2000
Brocade(config-dot1x)#timeout quiet-period 30
Brocade(config-dot1x)#timeout tx-period 60
Brocade(config-dot1x)#maxreq 6
Brocade(config-dot1x)#exit
Brocade(config)#interface e 1
Brocade(config-if-e1000-1)#dot1x port-control auto
Brocade(config-if-e1000-1)#exit
Brocade(config)#interface e 2
Brocade(config-if-e1000-2)#dot1x port-control auto
Brocade(config-if-e1000-2)#exit

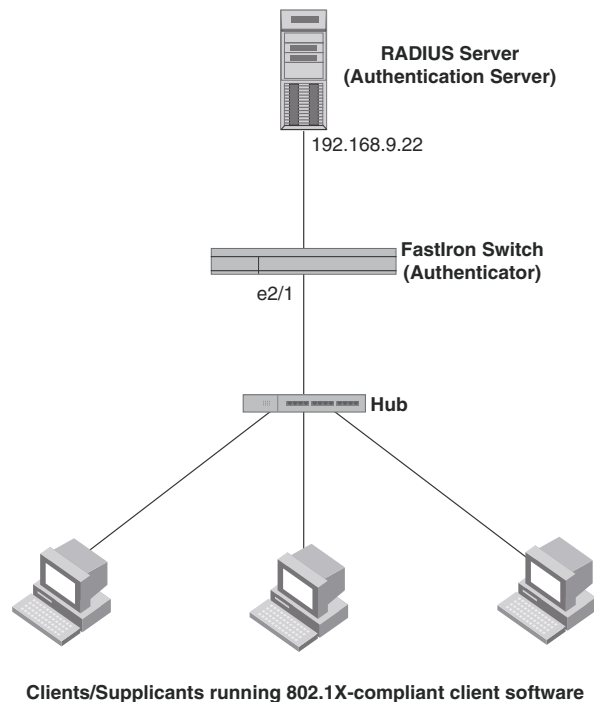
```

```
Brocade(config)#interface e 3
Brocade(config-if-e1000-3)#dot1x port-control auto
Brocade(config-if-e1000-3)#exit
```

Hub configuration

Figure 194 illustrates a configuration where three 802.1X-enabled Clients are connected to a hub, which is connected to a port on the Brocade device. The configuration is similar to that in Figure 193, except that 802.1X port security is enabled on only one port, and the **multiple-hosts** command is used to allow multiple Clients on the port.

FIGURE 194 Sample 802.1X configuration using a hub



Sample 802.1x configuration using a hub

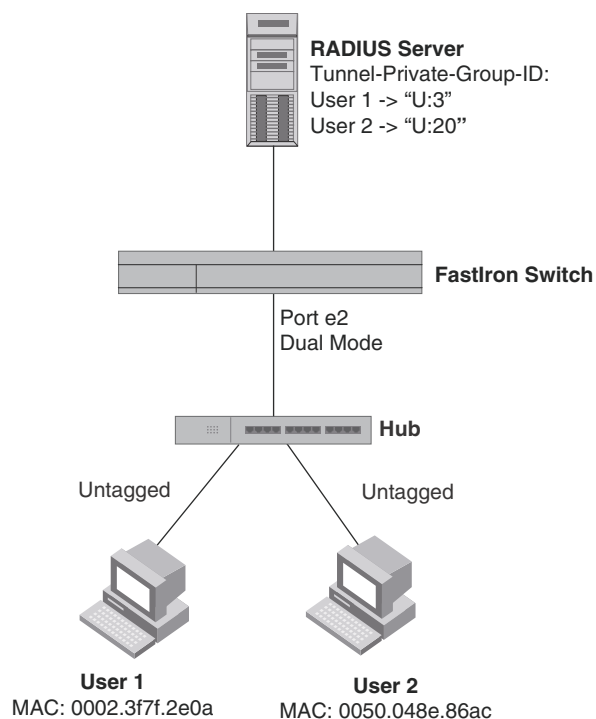
The following commands configure the Brocade device in Figure 194

```
Brocade(config)#aaa authentication dot1x default radius
Brocade(config)#radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x
Brocade(config)#dot1x-enable e 1
Brocade(config-dot1x)#re-authentication
Brocade(config-dot1x)#timeout re-authperiod 2000
Brocade(config-dot1x)#timeout quiet-period 30
Brocade(config-dot1x)#timeout tx-period 60
Brocade(config-dot1x)#maxreq 6
Brocade(config-dot1x)#exit
Brocade(config)#interface e 1
Brocade(config-if-e1000-1)#dot1x port-control auto
Brocade(config-if-e1000-1)#exit
```


802.1X Authentication with dynamic VLAN assignment

Figure 195 illustrates 802.1X authentication with dynamic VLAN assignment. In this configuration, two user PCs are connected to a hub, which is connected to port e2. Port e2 is configured as a dual-mode port. Both PCs transmit untagged traffic. The profile for User 1 on the RADIUS server specifies that User 1 PC should be dynamically assigned to VLAN 3. The RADIUS profile for User 2 on the RADIUS server specifies that User 2 PC should be dynamically assigned to VLAN 20.

FIGURE 195 Sample configuration using 802.1X authentication with dynamic VLAN assignment



In this example, the PVID for port e2 would be changed based on the first host to be successfully authenticated. If User 1 is authenticated first, then the PVID for port e2 is changed to VLAN 3. If User 2 is authenticated first, then the PVID for port e2 is changed to VLAN 20. Since a PVID cannot be changed by RADIUS authentication after it has been dynamically assigned, if User 2 is authenticated after the port PVID was changed to VLAN 3, then User 2 would not be able to gain access to the network.

If there were only one device connected to the port, and authentication failed for that device, it could be placed into the restricted VLAN, where it could gain access to the network.

The portion of the running-config related to 802.1X authentication is as follows.

```

dot1x-enable
 re-authentication
 servertimeout 10
 timeout re-authperiod 10
 auth-fail-action restricted-vlan
 auth-fail-vlanid 1023
 mac-session-aging no-aging permitted-mac-only
 enable ethe 2 to 4
!
!

```

```
!  
interface ethernet 2  
  dot1x port-control auto  
  dual-mode
```

If User 1 is successfully authenticated before User 2, the PVID for port e2 would be changed from the default VLAN to VLAN 3.

Had User 2 been the first to be successfully authenticated, the PVID would be changed to 20, and User 1 would not be able to gain access to the network. If there were only one device connected to the port that was sending untagged traffic, and 802.1X authentication failed for that device, it would be placed in the restricted VLAN 1023, and would be able to gain access to the network.

Multi-device port authentication and 802.1X security on the same port

You can configure the Brocade device to use multi-device port authentication and 802.1X security on the same port:

- The multi-device port authentication feature allows you to configure a Brocade device to forward or block traffic from a MAC address based on information received from a RADIUS server. Incoming traffic originating from a given MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication. A connecting user does not need to provide a specific username and password to gain access to the network.
- The IEEE 802.1X standard is a means for authenticating devices attached to LAN ports. Using 802.1X port security, you can configure a Brocade device to grant access to a port based on information supplied by a client to an authentication server.

When both of these features are enabled on the same port, multi-device port authentication is performed prior to 802.1X authentication. If multi-device port authentication is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

For more information, including configuration examples, see [“Multi-device port authentication and 802.1X security on the same port”](#) on page 1853.

MAC Port Security

Table 303 lists the individual Brocade FastIron switches and the Media Access Control (MAC) port security features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 303 Supported MAC port security features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
MAC port security	Yes	Yes	Yes	Yes	Yes
Setting the maximum number of secure MAC addresses on an interface	Yes	Yes	Yes	Yes	Yes
Setting the port security age timer	Yes	Yes	Yes	Yes	Yes
Specifying secure MAC addresses	Yes	Yes	Yes	Yes	Yes
Autosaving secure MAC addresses to the startup-config file	Yes	Yes	Yes	Yes	Yes
Specifying the action taken when a security violation occurs	Yes	Yes	Yes	Yes	Yes
Clearing port security statistics	Yes	Yes	Yes	Yes	Yes

This chapter describes how to configure Brocade devices to learn “secure” MAC addresses on an interface so that the interface will forward only packets that match the secure addresses.

MAC port security overview

You can configure the Brocade device to learn “secure” MAC addresses on an interface. The interface will forward only packets with source MAC addresses that match these learned secure addresses. The secure MAC addresses can be specified manually, or the Brocade device can learn them automatically. After the device reaches the limit for the number of secure MAC addresses it can learn on the interface, if the interface then receives a packet with a source MAC address that does not match the learned addresses, it is considered a security violation.

When a security violation occurs, a Syslog entry and an SNMP trap are generated. In addition, the device takes one of two actions: it either drops packets from the violating address (and allows packets from the secure addresses), or disables the port for a specified amount of time. You specify which of these actions takes place.

The secure MAC addresses are not flushed when an interface is disabled and re-enabled on FastIron X Series devices. The secure MAC addresses are flushed when an interface is disabled and re-enabled on FCX and ICX devices.

The secure addresses can be kept secure permanently (the default), or can be configured to age out, at which time they are no longer secure. You can configure the device to automatically save the secure MAC address list to the startup-config file at specified intervals, allowing addresses to be kept secure across system restarts.

Local and global resources used for MAC port security

The MAC port security feature uses a concept of local and global “resources” to determine how many MAC addresses can be secured on each interface. In this context, a “resource” is the ability to store one secure MAC address entry. Each interface is allocated 64 local resources. Additional global resources are shared among all interfaces on the device.

When the MAC port security feature is enabled on an interface, the interface can store one secure MAC address. You can increase the number of MAC addresses that can be secured using local resources to a maximum of 64.

Besides the maximum of 64 local resources available to an interface, there are additional global resources. Depending on flash memory size, a device can have 1024, 2048, or 4096 global resources available. When an interface has secured enough MAC addresses to reach its limit for local resources, it can secure additional MAC addresses by using global resources. Global resources are shared among all the interfaces on a first-come, first-served basis.

The maximum number of MAC addresses any single interface can secure is 64 (the maximum number of local resources available to the interface), plus the number of global resources not allocated to other interfaces.

Configuration notes and feature limitations for MAC port security

The following limitations apply to this feature:

- MAC port security applies only to Ethernet interfaces.
- MAC port security is not supported on static trunk group members or ports that are configured for link aggregation.
- MAC port security is not supported on 802.1X port security-enabled ports.

- Brocade devices do not support the **reserved-vlan-id** <num> command, which changes the default VLAN ID for the MAC port security feature.
- The SNMP trap generated for restricted MAC addresses indicates the VLAN ID associated with the MAC address, as well as the port number and MAC address.
- MAC port security is not supported on ports that have multi-device port authentication enabled.
- The first packet from each new secure MAC address is dropped if secure MAC addresses are learned dynamically.

MAC port security configuration

To configure the MAC port security feature, perform the following tasks:

- Enable the MAC port security feature
- Set the maximum number of secure MAC addresses for an interface
- Set the port security age timer
- Specify secure MAC addresses
- Configure the device to automatically save secure MAC addresses to the startup-config file
- Specify the action taken when a security violation occurs

Enabling the MAC port security feature

By default, the MAC port security feature is disabled on all interfaces. You can enable or disable the feature on all interfaces at once, or on individual interfaces.

To enable the feature on all interfaces at once, enter the following commands.

```
Brocade(config)#port security
Brocade(config-port-security)#enable
```

To disable the feature on all interfaces at once, enter the following commands.

```
Brocade(config)#port security
Brocade(config-port-security)#no enable
```

To enable the feature on a specific interface, enter the following commands.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#enable
```

Syntax: port security

Syntax: [no] enable

Setting the maximum number of secure MAC addresses for an interface

When MAC port security is enabled, an interface can store one secure MAC address. You can increase the number of MAC addresses that can be stored to a maximum of 64, plus the total number of global resources available.

For example, to configure interface 7/11 to have a maximum of 10 secure MAC addresses, enter the following commands.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#maximum 10
```

Syntax: `maximum <number-of-addresses>`

The `<number-of-addresses>` parameter can be set to a number from 0 through 64 plus (the total number of global resources available). The total number of global resources is 2048 or 4096, depending on flash memory size. Setting the parameter to 0 prevents any addresses from being learned. The default is 1.

Setting the port security age timer

By default, learned MAC addresses stay secure indefinitely. You can optionally configure the device to age out secure MAC addresses after a specified amount of time.

To set the port security age timer to 10 minutes on all interfaces, enter the following commands.

```
Brocade(config)#port security
Brocade(config-port-security)#age 10
```

To set the port security age timer to 10 minutes on a specific interface, enter the following commands.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#age 10
```

Syntax: `[no] age <minutes>`

The `<minutes>` variable specifies a range from 0 through 1440 minutes. The default is 0 (never age out secure MAC addresses).

NOTE

Even though you can set age time to specific ports independent of the device-level setting, the actual age timer will take the greater of the two values. Thus, if you set the age timer to 3 minutes for the port, and 10 minutes for the device, the port MAC aging happens in 10 minutes (the device-level setting), which is greater than the port setting that you have configured.

Specifying secure MAC addresses

You can configure secure MAC addresses on tagged and untagged interfaces.

On an untagged interface

To specify a secure MAC address on an untagged interface, enter commands such as the following.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#secure-mac-address 0050.DA18.747C
```

Syntax: [no] **secure-mac-address** <mac-address>

On a tagged interface

When specifying a secure MAC address on a tagged interface, you must also specify the VLAN ID. To do so, enter commands such as the following.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#secure-mac-address 0050.DA18.747C 2
```

Syntax: [no] **secure-mac-address** <mac-address> <vlan-ID>

NOTE

If MAC port security is enabled on a port and you change the VLAN membership of the port, make sure that you also change the VLAN ID specified in the **secure-mac-address** configuration statement for the port.

When a secure MAC address is applied to a tagged port, the VLAN ID is generated for both tagged and untagged ports. When you display the configuration, you will see an entry for the secure MAC addresses. For example, you might see an entry similar to the following line.

```
secure-mac-address 0000.1111.2222 10
```

This line means that MAC address 0000.1111.2222 on VLAN 10 is a secure MAC address.

Autosaving secure MAC addresses to the startup configuration

Learned MAC addresses can automatically be saved to the startup configuration at specified intervals. The autosave feature saves learned MAC addresses by copying the running configuration to the startup configuration.

For example, to automatically save learned secure MAC addresses every 20 minutes, enter the following commands.

```
Brocade(config)#port security
Brocade(config-port-security)#autosave 20
```

Syntax: [no] **autosave** <minutes>

The <minutes> variable can be from 15 through 1440 minutes. By default, secure MAC addresses are not autosaved to the startup-config file.

If you change the autosave interval, the next save happens according to the old interval, then the new interval takes effect. To change the interval immediately, disable autosave by entering the **no autosave** command, then configure the new autosave interval using the **autosave** command.

Specifying the action taken when a security violation occurs

A security violation can occur when a user tries to connect to a port where a MAC address is already locked, or the maximum number of secure MAC addresses has been exceeded. When a security violation occurs, an SNMP trap and Syslog message are generated.

You can configure the device to take one of two actions when a security violation occurs; either drop packets from the violating address (and allow packets from secure addresses), or disable the port for a specified time.

Dropping packets from a violating address

To configure the device to drop packets from a violating address and allow packets from secure addresses, enter the following commands.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#violation restrict
```

Syntax: violation [restrict]

NOTE

When the **restrict** option is used, the maximum number of MAC addresses that can be restricted is 128. If the number of violating MAC addresses exceeds this number, the port is shut down. An SNMP trap and the following Syslog message are generated: "Port Security violation restrict limit 128 exceeded on interface ethernet <port_id>". This is followed by a port shutdown Syslog message and trap.

Specifying the period of time to drop packets from a violating address

To specify the number of minutes that the device drops packets from a violating address, use commands similar to the following.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#violation restrict 5
```

Syntax: violation restrict <age>

The <age> variable can be from 0 through 1440 minutes. The default is 5 minutes. Specifying 0 drops packets from the violating address permanently.

Aging for restricted MAC addresses is done in software. There can be a worst case inaccuracy of one minute from the specified time.

The restricted MAC addresses are denied in hardware.

Disabling the port for a specified amount of time

You can configure the device to disable the port for a specified amount of time when a security violation occurs.

To shut down the port for 5 minutes when a security violation occurs, enter the following commands.

```
Brocade(config)#interface ethernet 7/11
Brocade(config-if-e1000-7/11)#port security
Brocade(config-port-security-e1000-7/11)#violation shutdown 5
```

Syntax: `violation shutdown <minutes>`

The minutes can be from 0 through 1440 minutes. Specifying 0 shuts down the port permanently when a security violation occurs.

Clearing port security statistics

You can clear restricted MAC addresses and violation statistics from ports on all ports or on individual ports.

Clearing restricted MAC addresses

To clear all restricted MAC addresses globally, enter the **clear port security restricted-macs all** command.

```
Brocade#clear port security restricted-macs all
```

To clear restricted MAC addresses on a specific port, enter a command such as the following.

```
Brocade#clear port security restricted-macs ethernet 5
```

Syntax: `clear port security restricted-macs all | ethernet <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Clearing violation statistics

To clear violation statistics globally, enter the **clear port security statistics all** command.

```
Brocade#clear port security statistics all
```

To clear violation statistics on a specific port, enter a command such as the following.

```
Brocade#clear port security statistics ethernet 1/5
```

Syntax: `clear port security statistics all | ethernet <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying port security information

You can display the following information about the MAC port security feature:

- The port security settings for an individual port or for all the ports on a specified module
- The secure MAC addresses configured on the device
- Port security statistics for an interface or for a module

Displaying port security settings

You can display the port security settings for an individual port or for all the ports on a specified module. For example, to display the port security settings for port 7/11, enter the following command.

```
Brocade#show port security ethernet 7/11
Port  Security Violation Shutdown-Time Age-Time  Max-MAC
-----
7/11 disabled shutdown 10 10 1
```

Syntax: `show port security ethernet <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

TABLE 304 Output from the `show port security ethernet` command

Field	Description
Port	The slot and port number of the interface.
Security	Whether the port security feature has been enabled on the interface.
Violation	The action to be undertaken when a security violation occurs, either “shutdown” or “restrict”.
Shutdown-Time	The number of seconds a port is shut down following a security violation, if the port is set to “shutdown” when a violation occurs.
Age-Time	The amount of time, in minutes, MAC addresses learned on the port will remain secure.
Max-MAC	The maximum number of secure MAC addresses that can be learned on the interface.

Displaying the secure MAC addresses

To list the secure MAC addresses configured on the device, enter the following command.

```
Brocade#show port security mac
Port Num-Addr Secure-Src-Addr Resource Age-Left Shutdown/Time-Left
-----
7/11 1 0050.da18.747c Local 10 no
```

Syntax: `show port security mac`

[Table 305](#) describes the output from the `show port security mac` command.

TABLE 305 Output from the `show port security mac` command

Field	Description
Port	The slot and port number of the interface.
Num-Addr	The number of MAC addresses secured on this interface.
Secure-Src-Addr	The secure MAC address.
Resource	Whether the address was secured using a local or global resource. Refer to “Local and global resources used for MAC port security” on page 1838 for more information.
Age-Left	The number of minutes the MAC address will remain secure.
Shutdown/Time-Left	Whether the interface has been shut down due to a security violation and the number of seconds before it is enabled again.

NOTE

For FCX and ICX switches, after every switchover or failover, the MAC “Age-Left” timer is reset to start since it is not synchronized between the master and the standby stack unit. This behavior is different on the FastIron SX devices where the “Age-Left” timer is not reset.

Displaying port security statistics

You can display port security statistics for an interface or for a module.

For example, to display port security statistics for interface 7/11, enter the following command.

```
Brocade#show port security statistics e 7/11
Port   Total-Addrs Maximum-Addrs Violation Shutdown/Time-Left
-----
7/11           1             1           0           no
```

Syntax: `show port security statistics <port>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

TABLE 306 Output from the `show port security statistics <port>` command

Field	Description
Port	The slot and port number of the interface.
Total-Addrs	The total number of secure MAC addresses on the interface.
Maximum-Addrs	The maximum number of secure MAC addresses on the interface.

TABLE 306 Output from the **show port security statistics <port>** command (Continued)

Field	Description
Violation	The number of security violations on the port.
Shutdown/Time-Left	Whether the port has been shut down due to a security violation and the number of seconds before it is enabled again.

For example, to display port security statistics for interface module 7, enter the **show port security statistics** command.

```
Brocade#show port security statistics 7
Module 7:
  Total ports: 0
  Total MAC address(es): 0
  Total violations: 0
  Total shutdown ports 0
```

Syntax: **show port security statistics <module>**

Table 307 describes the output from the **show port security statistics** *<module>* command.

TABLE 307 Output from the **show port security statistics** *<module>* command

Field	Description
Total ports	The number of ports on the module.
Total MAC address(es)	The total number of secure MAC addresses on the module.
Total violations	The number of security violations encountered on the module.
Total shutdown ports	The number of times that ports on the module shut down as a result of security violations.

Displaying restricted MAC addresses on a port

To display a list of restricted MAC addresses on a port, enter a command such as the following.

```
Brocade#show port security ethernet 1/5 restricted-macs
```

Syntax: **show port security ethernet** *<port>* **restricted-macs**

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying port security information

Multi-Device Port Authentication

Table 308 lists individual Brocade switches and the multi-device port authentication features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 308 Supported Multi-device port authentication (MDPA) features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Multi-Device Port Authentication	Yes	Yes	Yes	Yes	Yes
Support for Multi-Device Port Authentication together with:					
• Dynamic VLAN assignment	Yes	Yes	Yes	Yes	Yes
• Dynamic ACLs	Yes	Yes	Yes	Yes	Yes
• 802.1X	Yes	Yes	Yes	Yes	Yes
• Dynamic ARP inspection with dynamic ACLs	Yes	No	No	No	Yes
• DHCP snooping with dynamic ACLs	Yes	No	No	No	Yes
• Denial of Service (DoS) attack protection	Yes	No	Yes	Yes	Yes
• Source guard protection	Yes	Yes	Yes	Yes	Yes
• ACL-per-port-per-VLAN	Yes	Yes	Yes	Yes	Yes
Automatic removal of Dynamic VLAN for MAC authenticated ports	Yes	No	No	No	Yes
Authenticating multiple MAC addresses on an interface	Yes	Yes	Yes	Yes	Yes
Authenticating clients that send tagged packets on non-member ports	Yes	No	No	No	No
Specifying the format of the MAC addresses sent to the RADIUS server	Yes	Yes	Yes	Yes	Yes
Specifying the authentication-failure action	Yes	Yes	Yes	Yes	Yes
Password override	Yes	Yes	Yes	Yes	Yes
Specifying the RADIUS timeout action	Yes	Yes	Yes	Yes	Yes
SNMP Traps	Yes	Yes	Yes	Yes	Yes
MAC Address Filters	Yes	Yes	Yes	Yes	Yes
Aging time for blocked MAC Addresses	Yes	Yes	Yes	Yes	Yes

NOTE

FCX devices do **not** support:

- multi-device authentication on dynamic (LACP) and static trunk ports
 - multi-device authentication and port security configured on the same port
 - multi-device authentication and lock-address configured on the same port
-

Multi-device port authentication is a way to configure a Brocade device to forward or block traffic from a MAC address based on information received from a RADIUS server.

How multi-device port authentication works

Multi-device port authentication is a way to configure a Brocade device to forward or block traffic from a MAC address based on information received from a RADIUS server.

The multi-device port authentication feature is a mechanism by which incoming traffic originating from a specific MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication; the user does not need to provide a specific username and password to gain access to the network. If RADIUS authentication for the MAC address is successful, traffic from the MAC address is forwarded in hardware.

If the RADIUS server cannot validate the user's MAC address, then it is considered an authentication failure, and a specified authentication-failure action can be taken. The default authentication-failure action is to drop traffic from the non-authenticated MAC address in hardware. You can also configure the device to move the port on which the non-authenticated MAC address was learned into a restricted or “guest” VLAN, which may have limited access to the network.

RADIUS authentication

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The Brocade device supports multiple RADIUS servers; if communication with one of the RADIUS servers times out, the others are tried in sequential order. If a response from a RADIUS server is not received within a specified time (by default, 3 seconds) the RADIUS session times out, and the device retries the request up to three times. If no response is received, the next RADIUS server is chosen, and the request is sent for authentication.

The RADIUS server is configured with the usernames and passwords of authenticated users. For multi-device port authentication, the username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server. For example, given a MAC address of 0007e90feaa1, the users file on the RADIUS server would be configured with a username and password both set to 0007e90feaa1. When traffic from this MAC address is encountered on a MAC-authentication-enabled interface, the device sends the RADIUS server an Access-Request message with 0007e90feaa1 as both the username and password. The format of the MAC address sent to the RADIUS server is configurable through the CLI.

The request for authentication from the RADIUS server is successful only if the username and password provided in the request matches an entry in the users database on the RADIUS server. When this happens, the RADIUS server returns an Access-Accept message back to the Brocade device. When the RADIUS server returns an Access-Accept message for a MAC address, that MAC address is considered authenticated, and traffic from the MAC address is forwarded normally by the Brocade device.

Authentication-failure actions

If the MAC address does not match the username and password of an entry in the users database on the RADIUS server, then the RADIUS server returns an Access-Reject message. When this happens, it is considered an authentication failure for the MAC address. When an authentication failure occurs, the Brocade device can either drop traffic from the MAC address in hardware (the default), or move the port on which the traffic was received to a restricted VLAN.

Supported RADIUS attributes

Brocade devices support the following RADIUS attributes for multi-device port authentication:

- Username (1) – RFC 2865
- NAS-IP-Address (4) – RFC 2865
- NAS-Port (5) – RFC 2865
- Service-Type (6) – RFC 2865
- FilterId (11) – RFC 2865
- Framed-MTU (12) – RFC 2865
- State (24) – RFC 2865
- Vendor-Specific (26) – RFC 2865
- Session-Timeout (27) – RFC 2865
- Termination-Action (29) – RFC 2865
- Calling-Station-ID (31) – RFC 2865
- NAS-Port-Type (61) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Message-Authenticator (80) RFC 3579
- Tunnel-Private-Group-Id (81) – RFC 2868
- NAS-Port-id (87) – RFC 2869

Support for dynamic VLAN assignment

The Brocade multi-device port authentication feature supports **dynamic VLAN assignment**, where a port can be placed in one or more VLANs based on the MAC address learned on that interface. For details about this feature, refer to [“Configuring the RADIUS server to support dynamic VLAN assignment”](#) on page 1859.

Support for dynamic ACLs

The multi-device port authentication feature supports the assignment of a MAC address to a specific ACL, based on the MAC address learned on the interface. For details about this feature, refer to [“Dynamically applying IP ACLs to authenticated MAC addresses”](#) on page 1861.

Support for authenticating multiple MAC addresses on an interface

The multi-device port authentication feature allows multiple MAC addresses to be authenticated or denied authentication on each interface. The maximum number of MAC addresses that can be authenticated on each interface is limited only by the amount of system resources available on the Brocade device.

Support for dynamic ARP inspection with dynamic ACLs

NOTE

This feature is not supported on FWS and FCX devices.

Multi-device port authentication and Dynamic ARP Inspection (DAI) are supported in conjunction with dynamic ACLs. Support is available in the Layer 3 software images only.

DAI is supported together with multi-device port authentication as long as **ACL-per-port-per-vlan** is enabled. Otherwise, you do not need to perform any extra configuration steps to enable support with dynamic ACLs. When these features are enabled on the same port/VLAN, support is automatically enabled.

Support for DHCP snooping with dynamic ACLs

NOTE

This feature is not supported on FWS and FCX devices.

Multi-device port authentication and DHCP snooping are supported in conjunction with dynamic ACLs. Support is available in the Layer 3 software images only.

DHCP Snooping is supported together with multi-device port authentication as long as **ACL-per-port-per-vlan** is enabled. Otherwise, you do not need to perform any extra configuration steps to enable support with dynamic ACLs. When these features are enabled on the same port/VLAN, support is automatically enabled.

Support for source guard protection

The Brocade proprietary **Source Guard Protection** feature, a form of IP Source Guard, can be used in conjunction with multi-device port authentication. For details, refer to [“Enabling source guard protection”](#) on page 1864.

Multi-device port authentication and 802.1X security on the same port

On some Brocade devices, multi-device port authentication and 802.1X security can be configured on the same port, as long as the port is not a trunk port or an LACP port. When both of these features are enabled on the same port, multi-device port authentication is performed prior to 802.1X authentication. If multi-device port authentication is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

NOTE

When multi-device port authentication and 802.1X security are configured together on the same port, Brocade recommends that dynamic VLANs and dynamic ACLs are done at the multi-device port authentication level, and not at the 802.1X level.

When both features are configured on a port, a device connected to the port is authenticated as follows.

1. Multi-device port authentication is performed on the device to authenticate the device MAC address.
2. If multi-device port authentication is successful for the device, then the device checks whether the RADIUS server included the Foundry-802_1x-enable VSA (described in [Table 309](#)) in the Access-Accept message that authenticated the device.
3. If the Foundry-802_1x-enable VSA is not present in the Access-Accept message, or is present and set to 1, then 802.1X authentication is performed for the device.
4. If the Foundry-802_1x-enable VSA is present in the Access-Accept message, and is set to 0, then 802.1X authentication is skipped. The device is authenticated, and any dynamic VLANs specified in the Access-Accept message returned during multi-device port authentication are applied to the port.
5. If 802.1X authentication is performed on the device, and is successful, then dynamic VLANs or ACLs specified in the Access-Accept message returned during 802.1X authentication are applied to the port.

If multi-device port authentication fails for a device, then by default traffic from the device is either blocked in hardware, or the device is placed in a restricted VLAN. You can optionally configure the Brocade device to perform 802.1X authentication on a device when it fails multi-device port authentication. Refer to [“Example 2 – Creating a profile on the RADIUS server for each MAC address”](#) on page 1884 for a sample configuration where this is used.

Configuring Brocade-specific attributes on the RADIUS server

If the RADIUS authentication process is successful, the RADIUS server sends an Access-Accept message to the Brocade device, authenticating the device. The Access-Accept message can include Vendor-Specific Attributes (VSAs) that specify additional information about the device. If you are configuring multi-device port authentication and 802.1X authentication on the same port, then you can configure the Brocade VSAs listed in [Table 309](#) on the RADIUS server.

You add these Brocade vendor-specific attributes to your RADIUS server configuration, and configure the attributes in the individual or group profiles of the devices that will be authenticated. The Brocade Vendor-ID is 1991, with Vendor-Type 1.

TABLE 309 Brocade vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Description
Foundry-802_1x-enable	6	integer	Specifies whether 802.1X authentication is performed when multi-device port authentication is successful for a device. This attribute can be set to one of the following: 0 - Do not perform 802.1X authentication on a device that passes multi-device port authentication. Set the attribute to zero for devices that do not support 802.1X authentication. 1 - Perform 802.1X authentication when a device passes multi-device port authentication. Set the attribute to one for devices that support 802.1X authentication.
Foundry-802_1x-valid	7	integer	Specifies whether the RADIUS record is valid only for multi-device port authentication, or for both multi-device port authentication and 802.1X authentication. This attribute can be set to one of the following: 0 - The RADIUS record is valid only for multi-device port authentication. Set this attribute to zero to prevent a user from using their MAC address as username and password for 802.1X authentication 1 - The RADIUS record is valid for both multi-device port authentication and 802.1X authentication.

If neither of these VSAs exist in a device profile on the RADIUS server, then by default the device is subject to multi-device port authentication (if configured), then 802.1X authentication (if configured). The RADIUS record can be used for both multi-device port authentication and 802.1X authentication.

Configuration examples are shown in [“Examples of multi-device port authentication and 802.1X authentication configuration on the same port”](#) on page 1882.

Multi-device port authentication configuration

Configuring multi-device port authentication on the Brocade device consists of the following tasks:

- Enabling multi-device port authentication globally and on individual interfaces
- Specifying the format of the MAC addresses sent to the RADIUS server (optional)
- Specifying the authentication-failure action (optional)
- Enabling and disabling SNMP traps for multi-device port authentication
- Defining MAC address filters (optional)
- Configuring dynamic VLAN assignment (optional)
- Dynamically Applying IP ACLs to authenticated MAC addresses
- Enabling denial of service attack protection (optional)
- Clearing authenticated MAC addresses (optional)
- Disabling aging for authenticated MAC addresses (optional)
- Configuring the hardware aging period for blocked MAC addresses
- Specifying the aging time for blocked MAC addresses (optional)

Enabling multi-device port authentication

To enable multi-device port authentication, you first enable the feature globally on the device. On some Brocade devices, you can then enable the feature on individual interfaces.

Globally enabling multi-device port authentication

To globally enable multi-device port authentication on the device, enter the following command.

```
Brocade(config)#mac-authentication enable
```

Syntax: [no] mac-authentication enable

Enabling multi-device port authentication on an interface

To enable multi-device port authentication on an individual interface, enter a command such as the following.

```
Brocade(config)#mac-authentication enable ethernet 3/1
```

Syntax: [no] mac-authentication enable <port> | all

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **all** option enables the feature on all interfaces at once.

You can enable the feature on an interface at the interface CONFIG level.

Example of enabling multi-device port authentication on an interface

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication enable
```

Syntax: [no] mac-authentication enable

You can also configure multi-device port authentication commands on a range of interfaces.

Example of enabling multi-device port authentication on a range of interfaces

```
Brocade(config)#int e 3/1 to 3/12
Brocade(config-mif-3/1-3/12)#mac-authentication enable
```

Specifying the format of the MAC addresses sent to the RADIUS server

When multi-device port authentication is configured, the Brocade device authenticates MAC addresses by sending username and password information to a RADIUS server. The username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server.

By default, the MAC address is sent to the RADIUS server in the format xxxxxxxxxxxx. You can optionally configure the device to send the MAC address to the RADIUS server in the format xx-xx-xx-xx-xx-xx, or the format xxxx.xxxx.xxxx. To do this, enter a command such as the following.

```
Brocade(config)#mac-authentication auth-passwd-format xxxx.xxxx.xxxx
```

Syntax: [no] mac-authentication auth-passwd-format xxxx.xxxx.xxxx | xx-xx-xx-xx-xx-xx | xxxxxxxxxxxx

Specifying the authentication-failure action

When RADIUS authentication for a MAC address fails, you can configure the device to perform one of two actions:

- Drop traffic from the MAC address in hardware (the default)
- Move the port on which the traffic was received to a restricted VLAN

To configure the device to move the port to a restricted VLAN when multi-device port authentication fails, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication auth-fail-action restrict-vlan 100
```

Syntax: [no] mac-authentication auth-fail-action restrict-vlan [<vlan-id>]

If the ID for the restricted VLAN is not specified at the interface level, the global restricted VLAN ID applies for the interface.

To specify the VLAN ID of the restricted VLAN globally, enter the following command.

```
Brocade(config)#mac-authentication auth-fail-vlan-id 200
```

Syntax: [no] mac-authentication auth-fail-vlan-id <vlan-id>

The command above applies globally to all MAC-authentication-enabled interfaces.

Note that the restricted VLAN must already exist on the device. You cannot configure the restricted VLAN to be a non-existent VLAN. If the port is a tagged or dual-mode port, you cannot use a restricted VLAN as the authentication-failure action.

To configure the device to drop traffic from non-authenticated MAC addresses in hardware, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication auth-fail-action block-traffic
```

Syntax: [no] **mac-authentication auth-fail-action block-traffic**

Dropping traffic from non-authenticated MAC addresses is the default behavior when multi-device port authentication is enabled.

Generating traps for multi-device port authentication

You can enable and disable SNMP traps for multi-device port authentication. SNMP traps are enabled by default.

To enable SNMP traps for multi-device port authentication after they have been disabled, enter the following command.

```
Brocade(config)#snmp-server enable traps mac-authentication
```

Syntax: [no] **snmp-server enable traps mac-authentication**

Use the **no** form of the command to disable SNMP traps for multi-device port authentication.

Defining MAC address filters

You can specify MAC addresses that do not have to go through multi-device port authentication. These MAC addresses are considered pre-authenticated, and are not subject to RADIUS authentication. To do this, you can define MAC address filters that specify the MAC addresses to exclude from multi-device port authentication.

You should use a MAC address filter when the RADIUS server itself is connected to an interface where multi-device port authentication is enabled. If a MAC address filter is not defined for the MAC address of the RADIUS server and applied on the interface, the RADIUS authentication process would fail since the device would drop all packets from the RADIUS server itself.

For example, the following command defines a MAC address filter for address 0010.dc58.aca4.

```
Brocade(config)#mac-authentication mac-filter 1 0010.dc58.aca4
```

Syntax: [no] **mac-authentication mac-filter <filter>**

The following commands apply the MAC address filter on an interface so that address 0010.dc58.aca4 is excluded from multi-device port authentication.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication apply-mac-auth-filter 1
```

Syntax: [no] **mac-authentication apply-mac-auth-filter <filter-id>**

Configuring dynamic VLAN assignment

An interface can be dynamically assigned to one or more VLANs based on the MAC address learned on that interface. When a MAC address is successfully authenticated, the RADIUS server sends the Brocade device a RADIUS Access-Accept message that allows the Brocade device to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain attributes set for the MAC address in its access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies one or more VLAN identifiers, and the VLAN is available on the Brocade device, the port is moved from its default VLAN to the specified VLAN.

To enable dynamic VLAN assignment for authenticated MAC addresses, you must add attributes to the profile for the MAC address on the RADIUS server, then enable dynamic VLAN assignment on multi-device port authentication-enabled interfaces. Refer to [“Configuring the RADIUS server to support dynamic VLAN assignment”](#) on page 1859 for a list of the attributes that must be set on the RADIUS server.

To enable dynamic VLAN assignment on a multi-device port authentication-enabled interface, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication enable-dynamic-vlan
```

Syntax: [no] mac-authentication enable-dynamic-vlan

Configuring a port to remain in the restricted VLAN after a successful authentication attempt

If a previous authentication attempt for a MAC address failed, and as a result the port was placed in the restricted VLAN, but a subsequent authentication attempt was successful, the RADIUS Access-Accept message may specify a VLAN for the port. By default, the Brocade device moves the port out of the restricted VLAN and into the RADIUS-specified VLAN. You can optionally configure the device to leave the port in the restricted VLAN. To do this, enter the following command.

```
Brocade(config-if-e1000-3/1)#mac-authentication no-override-restrict-vlan
```

When the above command is applied, if the RADIUS-specified VLAN configuration is tagged (e.g., T:1024) and the VLAN is valid, then the port is placed in the RADIUS-specified VLAN as a tagged port and left in the restricted VLAN. If the RADIUS-specified VLAN configuration is untagged (e.g., U:1024), the configuration from the RADIUS server is ignored, and the port is left in the restricted VLAN.

Syntax: [no] mac-authentication no-override-restrict-vlan

Configuration notes for configuring a port to remain in the restricted VLAN

- If you configure dynamic VLAN assignment on a multi-device port authentication enabled interface, and the Access-Accept message returned by the RADIUS server contains a Tunnel-Type and Tunnel-Medium-Type, but does not contain a Tunnel-Private-Group-ID attribute, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.

- If the `<vlan-name>` string does not match either the name or the ID of a VLAN configured on the device, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- For tagged or dual-mode ports, if the VLAN ID provided by the RADIUS server does not match the VLAN ID in the tagged packet that contains the authenticated MAC address as its source address, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- If an untagged port had previously been assigned to a VLAN through dynamic VLAN assignment, and then another MAC address is authenticated on the same port, but the RADIUS Access-Accept message for the second MAC address specifies a different VLAN, then it is considered an authentication failure for the second MAC address, and the configured authentication failure action is performed. Note that this applies only if the first MAC address has not yet aged out. If the first MAC address has aged out, then dynamic VLAN assignment would work as expected for the second MAC address.
- For dual mode ports, if the RADIUS server returns T:<vlan-name>, the traffic will still be forwarded in the statically assigned PVID. If the RADIUS server returns U:<vlan-name>, the traffic will not be forwarded in the statically assigned PVID.

Configuring the RADIUS server to support dynamic VLAN assignment

To specify VLAN identifiers on the RADIUS server, add the following attributes to the profile for the MAC address on the RADIUS server, then enable dynamic VLAN assignment on multi-device port authentication-enabled interfaces.

Attribute name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	<vlan-name> (string) The <vlan-name> value can specify either the name or the number of one or more VLANs configured on the Brocade device.

For information about the attributes, refer to [“Dynamic multiple VLAN assignment for 802.1X ports”](#) on page 1803.

Also, refer to the example configuration of [“Multi-device port authentication with dynamic VLAN assignment”](#) on page 1879.

Enabling dynamic VLAN support for tagged packets on non-member VLAN ports

NOTE

This feature is not supported on FWS and FCX devices.

By default, the Brocade device drops tagged packets that are received on non-member VLAN ports. This process is called **ingress filtering**. Since the MAC address of the packets are not learned, authentication does not take place.

The Brocade device can authenticate clients that send tagged packets on non-member VLAN ports. This enables the Brocade device to add the VLAN dynamically. To enable support, enter the following command at the Interface level of the CLI.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication disable-ingress-filtering
```

If the client MAC address is successfully authenticated and the correct VLAN attribute is sent by the RADIUS server, the MAC address will be successfully authenticated on the VLAN.

Syntax: `mac-authentication disable-ingress-filtering`

Configuration notes and limitations:

- This feature works in conjunction with multi-device port authentication with dynamic VLAN assignment only. If this feature is not enabled, authentication works as in [“Example 2 – multi-device port authentication with dynamic VLAN assignment”](#) on page 1881 .
- The port on which ingress filtering is disabled must be tagged to a VLAN.
- If a host sends both tagged and untagged traffic, and ingress filtering is disabled on the port, the port must be configured as a dual-mode port.

Specifying to which VLAN a port is moved after its RADIUS-specified VLAN assignment expires

When a port is dynamically assigned to a VLAN through the authentication of a MAC address, and the MAC session for that address is deleted on the Brocade device, then by default the port is removed from its RADIUS-assigned VLAN and placed back in the VLAN where it was originally assigned.

A port can be removed from its RADIUS-assigned VLAN when any of the following occur:

- The link goes down for the port
- The MAC session is manually deleted with the `mac-authentication clear-mac-session` command
- The MAC address that caused the port to be dynamically assigned to a VLAN ages out

For example, say port 1/1 is currently in VLAN 100, to which it was assigned when MAC address 0007.eaa1.e90f was authenticated by a RADIUS server. The port was originally configured to be in VLAN 111. If the MAC session for address 0007.eaa1.e90f is deleted, then port 1/1 is moved from VLAN 100 back into VLAN 111.

You can optionally specify an alternate VLAN to which to move the port when the MAC session for the address is deleted. For example, to place the port in the restricted VLAN, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-auth move-back-to-old-vlan port-restrict-vlan
```

Syntax: `[no] mac-authentication move-back-to-old-vlan port-restrict-vlan | port-configured-vlan | system-default-vlan`

The **port-configured-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it back in the VLAN where it was originally assigned. This is the default.

The **port-restrict-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the restricted VLAN.

The **system-default-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the DEFAULT-VLAN.

NOTE

When a MAC session is deleted, if the port is moved back to a VLAN that is different than the running-config file, the system will update the running-config file to reflect the changes. This will occur even if **mac-authentication save-dynamicvlan-to-config** is not configured.

Automatic removal of dynamic VLAN assignments for MAC authenticated ports

NOTE

This feature is not supported on FWS and FCX devices.

By default, the Brocade device removes any association between a port and a dynamically-assigned VLAN when all authenticated MAC sessions for that tagged or untagged VLAN have expired on the port. Thus, RADIUS-specified VLAN assignments are not saved to the device running-config file. When the **show run** command is issued during a session, dynamically-assigned VLANs are not displayed, although they can be displayed with the **show vlan**, **show auth-mac-addresses detail**, and **show auth-mac-addresses authorized-mac** commands.

You can optionally configure the Brocade device to save the RADIUS-specified VLAN assignments to the device's running-config file. Refer to [“Saving dynamic VLAN assignments to the running-config file”](#), next.

Saving dynamic VLAN assignments to the running-config file

By default, dynamic VLAN assignments are not saved to the running-config file of the Brocade device. However, you can configure the device to do so by entering the following command.

```
Brocade(config)#mac-authentication save-dynamicvlan-to-config
```

When the above command is applied, dynamic VLAN assignments are saved to the running-config file and are displayed when the **show run** command is issued. Dynamic VLAN assignments can also be displayed with the **show vlan**, **show auth-mac-addresses detail**, and **show auth-mac-addresses authorized-mac** commands.

Syntax: [no] **mac-authentication save-dynamicvlan-to-config**

Dynamically applying IP ACLs to authenticated MAC addresses

The Brocade multi-device port authentication implementation supports the assignment of a MAC address to a specific ACL, based on the MAC address learned on the interface.

When a MAC address is successfully authenticated, the RADIUS server sends the Brocade device a RADIUS Access-Accept message that allows the Brocade device to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain, among other attributes, the Filter-ID (type 11) attribute for the MAC address. When the Access-Accept message containing the Filter-ID (type 11) attribute is received by the Brocade device, it will use the information in these attributes to apply an IP ACL on a per-MAC (per user) basis.

The dynamic IP ACL is active as long as the client is connected to the network. When the client disconnects from the network, the IP ACL is no longer applied to the port. If an IP ACL had been applied to the port prior to multi-device port authentication; it will be re-applied to the port.

NOTE

A dynamic IP ACL will take precedence over an IP ACL that is bound to a port (port ACL). When a client authenticates with a dynamic IP ACL, the port ACL will not be applied. Also, future clients on the same port will authenticate with a dynamic IP ACL or no IP ACL. If no clients on the port use dynamic ACL, then the port ACL will be applied to all traffic.

The Brocade device uses information in the Filter ID to apply an IP ACL on a per-user basis. The Filter-ID attribute can specify the number of an existing IP ACL configured on the Brocade device. If the Filter-ID is an ACL number, the specified IP ACL is applied on a per-user basis.

Multi-device port authentication with dynamic IP ACLs and ACL-per-port-per-VLAN

The following features are supported:

- FastIron X Series devices support multi-device port authentication and dynamic ACLs together with **ACL-per-port-per-vlan** (ACL filtering based on VLAN membership or VE port membership).
- Multi-device port authentication and dynamic ACLs are supported on tagged, dual-mode, and untagged ports, with or without virtual Interfaces.

Support is automatically enabled when all of the required conditions are met.

The following describes the conditions and feature limitations:

- On Layer 3 router code, dynamic IP ACLs are allowed on physical ports when **ACL-per-port-per-vlan** is enabled.
- On Layer 3 router code, dynamic IP ACLs are allowed on tagged and dual-mode ports when **ACL-per-port-per-vlan** is enabled. If **ACL-per-port-per-vlan** is not enabled, dynamic IP ACLs are not allowed on tagged or dual-mode ports.
- Dynamic IP ACLs can be added to tagged/untagged ports in a VLAN with or without a VE, as long as the tagged/untagged ports do not have configured ACLs assigned to them. The following shows some example scenarios where dynamic IP ACLs would not apply:
 - A port is a tagged/untagged member of VLAN 20, VLAN 20 includes VE 20, and an ACL is bound to VE 20.
 - A port is a tagged/untagged member of VLAN 20, VLAN 20 includes VE 20, and a per-port-per-vlan ACL is bound to VE 20 and to a subset of ports in VE 20

In the above scenarios, dynamic IP ACL assignment would not apply in either instance, because a configured ACL is bound to VE 20 on the port. Consequently, the MAC session would fail.

Configuration considerations and guidelines for multi-device port authentication

- On FastIron X Series devices, dynamic ARP inspection (DAI) and DHCP Snooping are supported together with dynamic ACLs.
- Dynamic IP ACLs with multi-device port authentication are supported. Dynamic MAC address filters with multi-device port authentication are not supported.
- In the Layer 2 switch code, dynamic IP ACLs are not supported when **ACL-per-port-per-vlan** is enabled on a global-basis.

- The RADIUS Filter ID (type 11) attribute is supported. The Vendor-Specific (type 26) attribute is not supported.
- The dynamic ACL must be an extended ACL. Standard ACLs are not supported.
- Multi-device port authentication and 802.1x can be used together on the same port. However, Brocade does not recommend the use of multi-device port authentication and 802.1X with dynamic ACLs together on the same port. If a single supplicant requires both 802.1x and multi-device port authentication, and if both 802.1x and multi-device port authentication try to install different dynamic ACLs for the same supplicant, the supplicant will fail authentication.
- Dynamically assigned IP ACLs are subject to the same configuration restrictions as non-dynamically assigned IP ACLs. One caveat is that ports with VE interfaces cannot have assigned user-defined ACLs. For example, a user-defined ACL bound to a VE or a port on a VE is not allowed. There are no restrictions on ports that do not have VE interfaces.
- Dynamic ACL filters are supported only for the inbound direction. Dynamic outbound ACL filters are not supported.
- Dynamic ACL assignment with multi-device port authentication is not supported in conjunction with any of the following features:
 - IP source guard
 - Rate limiting
 - Protection against ICMP or TCP Denial-of-Service (DoS) attacks
 - Policy-based routing
 - 802.1X dynamic filter

Configuring the RADIUS server to support dynamic IP ACLs

When a port is authenticated using multi-device port authentication, an IP ACL filter that exists in the running-config file on the Brocade device can be dynamically applied to the port. To do this, you configure the Filter-ID (type 11) attribute on the RADIUS server. The Filter-ID attribute specifies the name or number of the Brocade IP ACL.

The following is the syntax for configuring the Filter-ID attribute on the RADIUS server to refer to a Brocade IP ACL.

Value	Description
ip.<number>.in ¹	Applies the specified numbered ACL to the authenticated port in the inbound direction.
ip.<name>.in ^{1,2}	Applies the specified named ACL to the authenticated port in the inbound direction.

1. The ACL must be an extended ACL. Standard ACLs are not supported.
2. The <name> in the Filter ID attribute is case-sensitive

The following table lists examples of values you can assign to the Filter-ID attribute on the RADIUS server to refer to IP ACLs configured on a Brocade device.

Possible values for the filter ID attribute on the RADIUS server	ACLs configured on the Brocade device
ip.102.in	access-list 102 permit ip 36.0.0.0 0.255.255.255 any
ip.fdry_filter.in	ip access-list standard fdry_filter permit host 36.48.0.3

Enabling denial of service attack protection

NOTE

This feature is not supported on FWS devices.

The Brocade device does not start forwarding traffic from an authenticated MAC address in hardware until the RADIUS server authenticates the MAC address; traffic from the non-authenticated MAC addresses is sent to the CPU. A denial of service (DoS) attack could be launched against the device where a high volume of new source MAC addresses is sent to the device, causing the CPU to be overwhelmed with performing RADIUS authentication for these MAC addresses. In addition, the high CPU usage in such an attack could prevent the RADIUS response from reaching the CPU in time, causing the device to make additional authentication attempts.

To limit the susceptibility of the Brocade device to such attacks, you can configure the device to use multiple RADIUS servers, which can share the load when there are a large number of MAC addresses that need to be authenticated. The Brocade device can run a maximum of 10 RADIUS clients per server and will attempt to authenticate with a new RADIUS server if current one times out.

In addition, you can configure the Brocade device to limit the rate of authentication attempts sent to the RADIUS server. When the multi-device port authentication feature is enabled, it keeps track of the number of RADIUS authentication attempts made per second. When you also enable the DoS protection feature, if the number of RADIUS authentication attempts for MAC addresses learned on an interface per second exceeds a configurable rate (by default 512 authentication attempts per second), the device considers this a possible DoS attack and disables the port. You must then manually re-enable the port.

The DoS protection feature is disabled by default. To enable it on an interface, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication dos-protection enable
```

To specify a maximum rate for RADIUS authentication attempts, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication dos-protection mac-limit 256
```

Syntax: [no] **mac-authentication dos-protection mac-limit** <number>

You can specify a rate from 1 – 65535 authentication attempts per second. The default is a rate of 512 authentication attempts per second.

Enabling source guard protection

Source Guard Protection is a form of IP Source Guard used in conjunction with multi-device port authentication. When Source Guard Protection is enabled, IP traffic is blocked until the system learns the IP address. Once the IP address is validated, traffic with that source address is permitted.

NOTE

Source Guard Protection is supported together with multi-device port authentication as long as **ACL-per-port-per-vlan** is enabled.

When a new MAC session begins on a port that has Source Guard Protection enabled, the session will either apply a dynamically created Source Guard ACL entry, or it will use the dynamic IP ACL assigned by the RADIUS server. If a dynamic IP ACL is not assigned, the session will use the Source Guard ACL entry. The Source Guard ACL entry is **permit ip <secure-ip> any**, where <secure-ip> is obtained from the ARP Inspection table or from the DHCP Secure table. The DHCP Secure table is comprised of DHCP Snooping and Static ARP Inspection entries.

The Source Guard ACL permit entry is added to the hardware table after all of the following events occur:

- The MAC address is authenticated
- The IP address is learned
- The MAC-to-IP mapping is checked against the Static ARP Inspection table or the DHCP Secure table.

The Source Guard ACL entry is not written to the running configuration file. However, you can view the configuration using the **show auth-mac-addresses authorized-mac ip-addr**. Refer to [“Viewing the assigned ACL for ports on which source guard protection is enabled”](#) in the following section.

NOTE

The secure MAC-to-IP mapping is assigned at the time of authentication and remains in effect as long as the MAC session is active. If the DHCP Secure table is updated after the session is authenticated and while the session is still active, it does not affect the existing MAC session.

The Source Guard ACL permit entry is removed when the MAC session expires or is cleared.

To enable Source Guard Protection on a port on which multi-device port authentication is enabled, enter the following command at the Interface level of the CLI.

```
FastIron(config)#int e 1/4
FastIron(config-if-e1000-1/4)#mac-authentication source-guard-protection enable
```

Syntax: [no] **mac-authentication source-guard-protection enable**

Enter the **no** form of the command to disable SG protection.

Viewing the assigned ACL for ports on which source guard protection is enabled

Use the following command to view whether a Source Guard ACL or dynamic ACL is applied to ports on which Source Guard Protection is enabled.

```
Brocade(config)#show auth-mac-addresses authorized-mac ip-addr
```

MAC Address	SourceIp	Port	Vlan	Auth	Age	ACL	dot1x
00A1.0010.2000	200.1.17.5	6/12	171	Yes	Dis	SG	Ena
00A1.0010.2001	200.1.17.6	6/13	171	Yes	Dis	103	Ena

In the above output, for port 6/12, Source Guard Protection is enabled and the Source Guard ACL is applied to the MAC session, as indicated by **SG** in the **ACL** column. For port 6/13, Source Guard Protection is also enabled, but in this instance, a dynamic ACL (103) is applied to the MAC session.

Clearing authenticated MAC addresses

The Brocade device maintains an internal table of the authenticated MAC addresses (viewable with the **show authenticated-mac-address** command). You can clear the contents of the authenticated MAC address table either entirely, or just for the entries learned on a specified interface. In addition, you can clear the MAC session for an address learned on a specific interface.

To clear the entire contents of the authenticated MAC address table, enter the **clear auth-mac-table** command.

```
Brocade#clear auth-mac-table
```

Syntax: clear auth-mac-table

To clear the authenticated MAC address table of entries learned on a specified interface, enter a command such as the following.

```
Brocade#clear auth-mac-table e 3/1
```

Syntax: clear auth-mac-table ethernet <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

To clear the MAC session for an address learned on a specific interface, enter commands such as the following.

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication clear-mac-session 00e0.1234.abd4
```

Syntax: mac-authentication clear-mac-session <mac-address>

This command removes the Layer 2 CAM entry created for the specified MAC address. If the Brocade device receives traffic from the MAC address again, the MAC address is authenticated again.

NOTE

In a configuration with multi-device port authentication and 802.1X authentication on the same port, the **mac-authentication clear-mac-session** command will clear the MAC session, as well as its respective 802.1X session, if it exists.

Disabling aging for authenticated MAC addresses

MAC addresses that have been authenticated or denied by a RADIUS server are aged out if no traffic is received from the MAC address for a certain period of time:

- Authenticated MAC addresses or non-authenticated MAC addresses that have been placed in the restricted VLAN are aged out if no traffic is received from the MAC address over the device normal MAC aging interval.
- Non-authenticated MAC addresses that are blocked by the device are aged out if no traffic is received from the address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. (Refer to the next section for more information on configuring the software aging period).

You can optionally disable aging for MAC addresses subject to authentication, either for all MAC addresses or for those learned on a specified interface.

Globally disabling aging of MAC addresses

On most devices, you can disable aging for all MAC addresses on all interfaces where multi-device port authentication has been enabled by entering the **mac-authentication disable-aging** command.

```
Brocade(config)#mac-authentication disable-aging
```

Syntax: mac-authentication disable-aging

Enter the command at the global or interface configuration level.

The **denied-only** parameter prevents denied sessions from being aged out, but ages out permitted sessions.

The **permitted-only** parameter prevents permitted (authenticated and restricted) sessions from being aged out and ages denied sessions.

Disabling the aging of MAC addresses on interfaces

To disable aging for all MAC addresses subject to authentication on a specific interface where multi-device port authentication has been enabled, enter the command at the interface level.

Example

```
Brocade(config)#interface e 3/1
Brocade(config-if-e1000-3/1)#mac-authentication disable-aging
```

Syntax: [no] mac-authentication disable-aging

Changing the hardware aging period for blocked MAC addresses

When the Brocade device is configured to drop traffic from non-authenticated MAC addresses, traffic from the blocked MAC addresses is dropped in hardware, without being sent to the CPU. A Layer 2 hardware entry is created that drops traffic from the MAC address in hardware. If no traffic is received from the MAC address for a certain amount of time, this Layer 2 hardware entry is aged out. If traffic is subsequently received from the MAC address, then an attempt can be made to authenticate the MAC address again.

Aging of the Layer 2 hardware entry for a blocked MAC address occurs in two phases, known as hardware aging and software aging.

On FastIron devices, the hardware aging period for blocked MAC addresses is fixed at 70 seconds and is non-configurable. (The hardware aging time for non-blocked MAC addresses is the length of time specified with the **mac-age** command.) The software aging period for blocked MAC addresses is configurable through the CLI, with the **mac-authentication max-age** command. Once the hardware aging period ends, the software aging period begins. When the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the Brocade device receives traffic from the MAC address.

On FastIron X Series devices, the hardware aging period for blocked MAC addresses is not fixed at 70 seconds. The hardware aging period for blocked MAC addresses is equal to the length of time specified with the **mac-age** command. As on FastIron devices, once the hardware aging period ends, the software aging period begins. When the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the device receives traffic from the MAC address.

To change the hardware aging period for blocked MAC addresses, enter a command such as the following.

```
Brocade(config)#mac-authentication hw-deny-age 10
```

Syntax: [no] **mac-authentication hw-deny-age** <num>

The <num> parameter is a value from 1 to 65535 seconds. The default is 70 seconds.

Specifying the aging time for blocked MAC addresses

When the Brocade device is configured to drop traffic from non-authenticated MAC addresses, traffic from the blocked MAC addresses is dropped in hardware, without being sent to the CPU. A Layer 2 CAM entry is created that drops traffic from the blocked MAC address in hardware. If no traffic is received from the blocked MAC address for a certain amount of time, this Layer 2 CAM entry is aged out. If traffic is subsequently received from the MAC address, then an attempt can be made to authenticate the MAC address again.

Aging of the Layer 2 CAM entry for a blocked MAC address occurs in two phases, known as **hardware aging** and **software aging**. The hardware aging period is fixed at 70 seconds and is non-configurable. The software aging time is configurable through the CLI.

Once the Brocade device stops receiving traffic from a blocked MAC address, the hardware aging begins and lasts for a fixed period of time. After the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (by default 120 seconds). After the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the Brocade device receives traffic from the MAC address.

To change the length of the software aging period for blocked MAC addresses, enter a command such as the following.

```
Brocade(config)#mac-authentication max-age 180
```

Syntax: [no] **mac-authentication max-age** <seconds>

You can specify from 1 – 65535 seconds. The default is 120 seconds.

Specifying the RADIUS timeout action

A RADIUS timeout occurs when the Brocade device does not receive a response from a RADIUS server within a specified time limit and after a certain number of retries. The time limit and number of retries can be manually configured using the CLI commands **radius-server timeout** and **radius-server retransmit**, respectively. If the parameters are not manually configured, the Brocade device applies the default value of three seconds with a maximum of three retries.

You can better control port behavior when a RADIUS timeout occurs by configuring a port on the Brocade device to automatically pass or fail user authentication. A **pass** essentially bypasses the authentication process and permits user access to the network. A **fail** bypasses the authentication process and blocks user access to the network, unless restrict-vlan is configured, in which case, the user is placed into a VLAN with restricted or limited access. By default, the Brocade device will reset the authentication process and retry to authenticate the user.

Specify the RADIUS timeout action at the Interface level of the CLI.

Permit User access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and *permit* user access to the network, enter commands such as the following.

```
Brocade(config)#interface ethernet 1/3
Brocade(config-if-e100-1/3)#mac-authentication auth-timeout-action success
```

Syntax: [no] mac-authentication auth-timeout-action success

Once the *success* timeout action is enabled, use the *no* form of the command to reset the RADIUS timeout behavior to *retry*.

Deny User access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and block user access to the network, enter commands such as the following.

```
Brocade(config)#interface ethernet 1/3
Brocade(config-if-e100-1/3)#mac-authentication auth-timeout-action failure
```

Syntax: [no] mac-authentication auth-timeout-action failure

Once the *failure* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

NOTE

If **restrict-vlan** is configured along with **auth-timeout-action failure**, the user will be placed into a VLAN with restricted or limited access. Refer to [“Allow user access to a restricted VLAN after a RADIUS timeout”](#) on page 1869.

Allow user access to a restricted VLAN after a RADIUS timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and place the user in a VLAN with restricted or limited access, enter commands such as the following.

```
Brocade(config)#interface ethernet 1/3
Brocade(config-if-e100-1/3)#mac-authentication auth-fail-action restrict-vlan 100
Brocade(config-if-e100-1/3)#mac-authentication auth-timeout-action failure
```

Syntax: [no] mac-authentication auth-fail-action restrict-vlan [<vlan-id>]

Syntax: [no] mac-authentication auth-timeout-action failure

Multi-device port authentication password override

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The RADIUS server is configured with the usernames and passwords of authenticated users. For multi-device port authentication, the username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server. For example, given a MAC address of 0007e90feaa1, the users file on the RADIUS server would be configured with a username and password both set to 0007e90feaa1. When traffic from this MAC address is encountered on a MAC-authentication-enabled interface, the device sends the RADIUS server an Access-Request message with 0007e90feaa1 as both the username and password.

The MAC address is the *default* password for multi-device port authentication, and you can optionally configure the device to use a different password. Note that the MAC address is still the username and cannot be changed.

To change the password for multi-device port authentication, enter a command such as the following at the GLOBAL Config Level of the CLI.

```
Brocade(config)#mac-authentication password-override
```

Syntax: [no] **mac-authentication password-override** <password>

where <password> can have up to 32 alphanumeric characters, but cannot include blank spaces.

Limiting the number of authenticated MAC addresses

You cannot enable MAC port security on the same port that has multi-device port authentication enabled. To simulate the function of MAC port security, you can enter a command such as the following.

```
Brocade(config-if-e1000-2)#mac-authentication max-accepted-session 5
```

Syntax: [no] **mac-authentication max-accepted-session** <session-number>

This command limits the number of successfully authenticated MAC addresses. Enter a value from 1 - 250 for session-number

Displaying multi-device port authentication information

You can display the following information about the multi-device port authentication configuration:

- Information about authenticated MAC addresses
- Information about the multi-device port authentication configuration
- Authentication Information for a specific MAC address or port
- Multi-device port authentication settings and authenticated MAC addresses for each port where the multi-device port authentication feature is enabled
- The MAC addresses that have been successfully authenticated
- The MAC addresses for which authentication was not successful

Displaying authenticated MAC address information

To display information about authenticated MAC addresses on the ports where the multi-device port authentication feature is enabled, enter the **show auth-mac address** command.

```
Brocade#show auth-mac-address
```

Port	Vlan	Accepted MACs	Rejected MACs	Attempted-MACs
1/18	100	1	100	0
1/20	40	0	0	0
1/22	100	0	0	0
4/5	30	0	0	0

Syntax: **show auth-mac-address**

The following table describes the information displayed by the **show auth-mac-address** command.

TABLE 310 Output from the **show authenticated-mac-address** command

Field	Description
Port	The port number where the multi-device port authentication feature is enabled.
Vlan	The VLAN to which the port has been assigned.
Accepted MACs	The number of MAC addresses that have been successfully authenticated
Rejected MACs	The number of MAC addresses for which authentication has failed.
Attempted-MACs	The rate at which authentication attempts are made for MAC addresses.

Displaying multi-device port authentication configuration information

To display information about the multi-device port authentication configuration, enter the **show auth-mac-address configuration** command.

```
Brocade#show auth-mac-address configuration
```

```
Feature enabled           : Yes
Number of Ports enabled   : 4
```

Port	Fail-Action	Fail-vlan	Dyn-vlan	MAC-filter
1/18	Block Traffic	1	No	No
1/20	Block Traffic	1	No	No
1/22	Block Traffic	1	No	Yes
4/5	Block Traffic	1	No	No

Syntax: **show auth-mac-address configuration**

The following table describes the output from the **show auth-mac-address configuration** command.

TABLE 311 Output from the **show authenticated-mac-address** configuration command

Field	Description
Feature enabled	Whether multi-device port authentication is enabled on the Brocade device.
Number of Ports enabled	The number of ports on which the multi-device port authentication feature is enabled.
Port	Information for each multi-device port authentication-enabled port.
Fail-Action	What happens to traffic from a MAC address for which RADIUS authentication has failed either block the traffic or assign the MAC address to a restricted VLAN.
Fail-vlan	The restricted VLAN to which non-authenticated MAC addresses are assigned, if the Fail-Action is to assign the MAC address to a restricted VLAN.
Dyn-vlan	Whether RADIUS dynamic VLAN assignment is enabled for the port.
MAC-filter	Whether a MAC address filter has been applied to specify pre-authenticated MAC addresses.

Displaying multi-device port authentication information for a specific MAC address or port

To display authentication information for a specific MAC address or port, enter a command such as the following.

```
Brocade#show auth-mac-address 0007.e90f.eaa1
-----
MAC/IP Address                Port      Vlan  Authenticated Time   Age  CAM
                                Index
-----
0007.e90f.eaa1 : 25.25.25.25   1/18     100   Yes    00d01h10m06s 0    N/A
```

Syntax: **show auth-mac-address** <mac-address> | <ip-addr> | <port>

The <ip-addr> variable lists the MAC address associated with the specified IP address.

The <slotnum> variable is required on chassis devices.

The <port> variable is a valid port number. Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The following table describes the information displayed by the **show authenticated-mac-address** command for a specified MAC address or port.

TABLE 312 Output from the **show authenticated-mac-address <address>** command

Field	Description
MAC/IP Address	The MAC address for which information is displayed. If the packet for which multi-device port authentication was performed also contained an IP address, then the IP address is displayed as well.
Port	The port on which the MAC address was learned.
Vlan	The VLAN to which the MAC address was assigned.
Authenticated	Whether the MAC address was authenticated.
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	If the MAC address is blocked, this is the index entry for the Layer 2 CAM entry created for this MAC address. If the MAC address is not blocked, either through successful authentication or through being placed in the restricted VLAN, then "N/A" is displayed. If the hardware aging period has expired, then "ffff" is displayed for the MAC address during the software aging period.

Displaying the authenticated MAC addresses

To display the MAC addresses that have been successfully authenticated, enter the **show auth-mac-addresses authorized-mac** command.

The following example output is from a FastIron X Series device. The display output on your device may differ, depending on the software version running on the device.

```
Brocade#show auth-mac-addresses authorized-mac
```

```
-----
MAC Address      Port      Vlan  Authenticated Time      Age      dot1x
-----
0030.4874.3181   15/23     101   Yes  00d01h03m17s  Ena      Ena
000f.ed00.0001   18/1      87    Yes  00d01h03m17s  Ena      Ena
000f.ed00.012d   18/1      87    Yes  00d01h03m17s  Ena      Ena
000f.ed00.0065   18/1      87    Yes  00d01h03m17s  Ena      Ena
000f.ed00.0191   18/1      87    Yes  00d01h03m17s  Ena      Ena
000f.ed00.01f5   18/1      87    Yes  00d01h03m17s  Ena      Ena
-----
```

Syntax: **show auth-mac-addresses authorized-mac**

Displaying the non-authenticated MAC addresses

To display the MAC addresses for which authentication was not successful, enter the **show auth-mac-addresses unauthorized-mac** command

```
Brocade#show auth-mac-addresses unauthorized-mac
```

```
-----
MAC Address      Port      Vlan  Authenticated Time   Age  dot1x
-----
000f.ed00.0321   18/1      87    No    00d01h03m17s  H44  Ena
000f.ed00.0259   18/1      87    No    00d01h03m17s  H44  Ena
000f.ed00.0385   18/1      87    No    00d01h03m17s  H44  Ena
000f.ed00.02bd   18/1      87    No    00d01h03m17s  H44  Ena
000f.ed00.00c9   18/1      87    No    00d01h03m17s  H44  Ena
```

Syntax: **show auth-mac-addresses unauthorized-mac**

[Table 313](#) explains the information in the output.

Displaying multi-device port authentication information for a port

To display a summary of Multi-Device Port Authentication for ports on a device, enter the following command

```
Brocade#show auth-mac-addresses ethernet 18/1
```

```
-----
MAC Address      Port      Vlan  Authenticated Time   Age  Dot1x
-----
000f.ed00.0001   18/1      87    Yes    00d01h03m17s  Ena  Ena
000f.ed00.012d   18/1      87    Yes    00d01h03m17s  Ena  Ena
000f.ed00.0321   18/1      87    No     00d01h03m17s  H52  Ena
000f.ed00.0259   18/1      87    No     00d01h03m17s  H52  Ena
000f.ed00.0065   18/1      87    Yes    00d01h03m17s  Ena  Ena
000f.ed00.0385   18/1      87    No     00d01h03m17s  H52  Ena
000f.ed00.0191   18/1      87    Yes    00d01h03m17s  Ena  Ena
000f.ed00.02bd   18/1      87    No     00d01h03m17s  H52  Ena
000f.ed00.00c9   18/1      87    No     00d01h03m17s  H52  Ena
000f.ed00.01f5   18/1      87    Yes    00d01h03m17s  Ena  Ena
```

Syntax: **show auth-mac-address ethernet <port>**

[Table 313](#) explains the information in the output.

TABLE 313 Output of show auth-mac-address

Field	Description
MAC Address	The MAC addresses learned on the port. If the packet for which multi-device port authentication was performed also contained an IP address, the IP address is also displayed.
Port	ID of the port on which the MAC address was learned.
VLAN	VLAN of which the port is a member.
Authenticated	Whether the MAC address has been authenticated by the RADIUS server.

TABLE 313 Output of show auth-mac-address (Continued)

Field	Description
Time	The time the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates if 802.1X authentication is enabled or disabled for the MAC address

Displaying multi-device port authentication settings and authenticated MAC addresses

To display the multi-device port authentication settings and authenticated MAC addresses for a port where the feature is enabled, enter the following command.

Syntax: `show auth-mac-address [detail] [ethernet <port>]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Omitting the **ethernet** *<port>* parameter displays information for all interfaces where the multi-device port authentication feature is enabled.

Displaying multi-device port authentication information

```

Brocade#show auth-mac-addresses detailed ethernet 15/23
Port : 15/23
Dynamic-Vlan Assignment : Enabled
RADIUS failure action : Block Traffic
    Failure restrict use dot1x : No
Override-restrict-vlan : Yes
Port Default VLAN : 101 ( RADIUS assigned: No) (101)
Port Vlan State : DEFAULT
802.1x override Dynamic PVID : YES
    override return to PVID : 101
Original PVID : 101
DOS attack protection : Disabled
Accepted Mac Addresses : 1
Rejected Mac Addresses : 0
Authentication in progress : 0
Authentication attempts : 0
RADIUS timeouts : 0
RADIUS timeouts action : Success
MAC Address on PVID : 1
MAC Address authorized on PVID : 1
Aging of MAC-sessions : Enabled
Port move-back vlan : Port-configured-vlan
Max-Age of sw mac session : 120 seconds
hw age for denied mac : 70 seconds
MAC Filter applied : No
Dynamic ACL applied : No
num Dynamic Tagged Vlan : 2
Dynamic Tagged Vlan list : 1025 (1/1) 4060 (1/0)

```

```

-----
MAC Address      RADIUS Server  Authenticated  Time           Age  Dot1x
-----
0030.4874.3181  64.12.12.5    Yes           00d01h03m17s  Ena  Ena

```

The following table describes the information displayed by the **show auth-mac-addresses detailed** command.

TABLE 314 Output from the **show auth-mac-addresses detailed** command

Field	Description
Port	The port to which this information applies.
Dynamic-Vlan Assignment	Whether RADIUS dynamic VLAN assignment has been enabled for the port.
RADIUS failure action	What happens to traffic from a MAC address for which RADIUS authentication has failed either block the traffic or assign the MAC address to a restricted VLAN.
Failure restrict use dot1x	Indicates if 802.1x traffic that failed multi-device port authentication, but succeeded 802.1x authentication to gain access to the network.
Override-restrict-vlan	Whether a port can be dynamically assigned to a VLAN specified by a RADIUS server, if the port had been previously placed in the restricted VLAN because a previous attempt at authenticating a MAC address on that port failed.
Port Default Vlan	The VLAN to which the port is assigned, and whether the port had been dynamically assigned to the VLAN by a RADIUS server.
Port VLAN state	Indicates the state of the port VLAN. The State can be one of the following "Default", "RADIUS Assigned" or "Restricted".

TABLE 314 Output from the **show auth-mac-addresses detailed** command (Continued)

Field	Description
802.1X override Dynamic PVID	Indicates if 802.1X can dynamically assign a Port VLAN ID (PVID).
override return to PVID	If a port PVID is assigned through the multi-device port authentication feature, and 802.1X authentication subsequently specifies a different PVID, then the PVID specified through 802.1X authentication overrides the PVID specified through multi-device port authentication. This line indicates the PVID the port will use if 802.1X dynamically assigns PVID.
Original PVID	The originally configured (not dynamically assigned) PVID for the port.
DOS attack protection	Whether denial of service attack protection has been enabled for multi-device port authentication, limiting the rate of authentication attempts sent to the RADIUS server.
Accepted Mac Addresses	The number of MAC addresses that have been successfully authenticated.
Rejected Mac Addresses	The number of MAC addresses for which authentication has failed.
Authentication in progress	The number of MAC addresses for which authentication is pending. This is the number of MAC addresses for which an Access-Request message has been sent to the RADIUS server, and for which the RADIUS server has not yet sent an Access-Accept message.
Authentication attempts	The total number of authentication attempts made for MAC addresses on an interface, including pending authentication attempts.
RADIUS timeouts	The number of times the session between the Brocade device and the RADIUS server timed out.
RADIUS timeout action	Action to be taken by the RADIUS server if it times out.
MAC address on the PVID	Number of MAC addresses on the PVID.
MAC address authorized on PVID	Number of authorized MAC addresses on the PVID.
Aging of MAC-sessions	Whether software aging of MAC addresses is enabled.
Port move-back VLAN	Indicates the destination VLAN when a RADIUS assigned VLAN is removed. By default, it would return the configured VLAN.
Max-Age of sw MAC-sessions	The configured software aging period for MAC addresses.
hw age for denied MAC	The hardware aging period for blocked MAC addresses. The MAC addresses are dropped in hardware ones the aging period expires.
MAC Filter applied	Indicates whether a MAC address filter has been applied to this port to specify pre-authenticated MAC addresses.
Dynamic ACL applied	Indicates whether a dynamic ACL was applied to this port.
num Dynamic Tagged Vlan	The number of dynamically tagged VLANs on this port.
Dynamic Tagged Vlan list	The list of dynamically tagged VLANs on this port. In this example, 1025 (1/1) indicates that there was one MAC session and one learned MAC address for VLAN 1025. Likewise, 4060 (1/0) indicates that there was one MAC session and no learned MAC addresses for VLAN 4060.
MAC Address	The MAC addresses learned on the port. If the packet for which multi-device port authentication was performed also contained an IP address, then the IP address is displayed as well.
RADIUS Server	The IP address of the RADIUS server used for authenticating the MAC addresses.

TABLE 314 Output from the **show auth-mac-addresses detailed** command (Continued)

Field	Description
Authenticated	Whether the MAC address has been authenticated by the RADIUS server.
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicated if 802.1X authentication is enabled or disabled for the MAC address

Displaying the MAC authentication table for FCX devices

For FCX devices, there are three commands you can use to display MAC authentication information:

- `show table <mac address>`
- `show table allowed-mac`
- `show table denied-mac`

This section describes the output for these commands.

To display MAC authentication information for FCX devices, enter the **show table <mac address>** command as shown.

```
Brocade#show table 0000.0010.1002
```

Syntax: `show table <mac address>`

The `<mac address>` variable is the specified MAC address.

```
Brocade#show table 0000.0010.1002
```

```
-----
MAC Address Port Vlan Authenticated Time Age dot1x
-----
0000.0010.1002 2/1/48 2 Yes 00d00h30m57s Ena Dis
Brocade#
```

To display the table of allowed (authenticated) mac addresses enter the **show table allowed-mac** command as shown.

Syntax: `show table allowed-mac`

```
Brocade#show table allowed-mac
```

```
-----
MAC Address PortVlanAuthenticatedTimeAgedot1x
-----
0000.0010.100a 1/1/1 2 Yes 00d00h30m57s Ena Dis
0000.0010.100b 1/1/1 2 Yes 00d00h31m00s Ena Dis
0000.0010.1002 2/1/48 2 Yes 00d00h30m57s Ena Dis
0000.0010.1003 2/1/48 2 Yes 00d00h30m57s Ena Dis
0000.0010.1004 2/1/48 2 Yes 00d00h30m57s Ena Dis
Brocade#
```

To display the table of allowed mac addresses enter the **show table denied-mac** command as shown.

Syntax: **show table** <mac address>

The <mac address> variable is the specified MAC address.

```
Brocade#show table denied-mac
```

MAC Address	Port	Vlan	Authenticated	Time	Age	dot1x
0000.0010.1021	2/1/48	4092	No	00d00h32m48s	H8	Dis
0000.0010.1022	2/1/48	4092	No	00d00h32m48s	H8	Dis

```
Brocade#
```

To display MAC authentication for a specific port, enter the **show table ethernet** <stack-unit/slot/port> command as shown.

```
Brocade#show table eth 2/1/48
```

MAC Address	Port	Vlan	Authenticated	Time	Age	CAM	MAC	Dot1x	Type	Pri	Index	Index
0000.0010.1002	2/1/48	2	Yes	00d00h30m	57s	Ena	0000	70d4	Dis	Dyn	0	
0000.0010.1003	2/1/48	2	Yes	00d00h30m	57s	Ena	0002	3df0	Dis	Dyn	0	
0000.0010.1004	2/1/48	2	Yes	00d00h30m	57s	Ena	0001	1e74	Dis	Dyn	0	
0000.0010.1021	2/1/48	4092	No	00d00h36m	22s	H60	0003	7a2c	Dis	Dyn	0	
0000.0010.1022	2/1/48	4092	No	00d00h36m	22s	H60	0004	4d7c	Dis	Dyn	0	

```
Brocade#
```

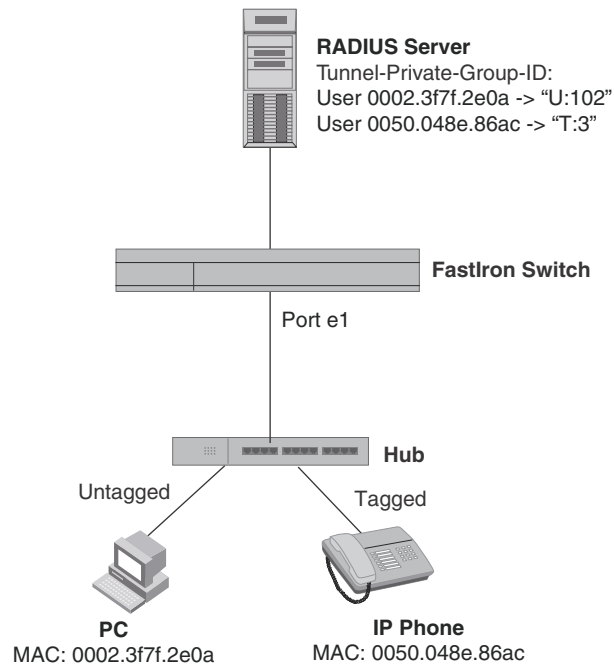
Example port authentication configurations

This section includes configuration examples of multi-device port authentication with dynamic VLAN assignment, and multi-device port authentication and 802.1X authentication.

Multi-device port authentication with dynamic VLAN assignment

[Figure 197](#) illustrates multi-device port authentication with dynamic VLAN assignment on a Brocade device. In this configuration, a PC and an IP phone are connected to a hub, which is connected to port e1 on a Brocade device. The profile for the PC MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN 102, and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to VLAN 3.

FIGURE 196 Using multi-device port authentication with dynamic VLAN assignment



In this example, multi-device port authentication is performed for both devices. If the PC is successfully authenticated, port e1 PVID is changed from VLAN 1 (the DEFAULT-VLAN) to VLAN 102. If authentication for the PC fails, then the PC can be placed in a specified "restricted" VLAN, or traffic from the PC can be blocked in hardware. In this example, if authentication for the PC fails, the PC would be placed in VLAN 1023, the restricted VLAN.

If authentication for the IP phone is successful, then port e1 is added to VLAN 3. If authentication for the IP phone fails, then traffic from the IP phone would be blocked in hardware. (Devices sending tagged traffic cannot be placed in the restricted VLAN.)

The portion of the running-config related to multi-device port authentication is as follows.

```
mac-authentication enable
mac-authentication auth-fail-vlan-id 1023

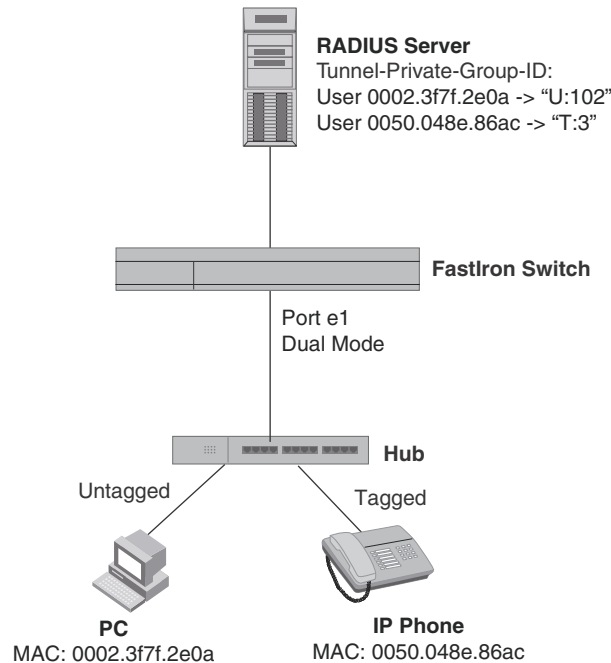
interface ethernet 1
  dual-mode
  mac-authentication enable
  mac-authentication auth-fail-action restrict-vlan
  mac-authentication enable-dynamic-vlan
  mac-authentication disable-ingress-filtering
```

The **mac-authentication disable-ingress-filtering** command enables tagged packets on the port, even if the port is not a member of the VLAN. If this feature is not enabled, authentication works as in ["Example 2 – multi-device port authentication with dynamic VLAN assignment"](#)

Example 2 – multi-device port authentication with dynamic VLAN assignment

Figure 197 illustrates multi-device port authentication with dynamic VLAN assignment on a Brocade device. In this configuration, a PC and an IP phone are connected to a hub, which is connected to port e1 on a Brocade device. Port e1 is configured as a dual-mode port. Also, **mac-authentication disable-ingress-filtering** is enabled on the port. The profile for the PC MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN 102, and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to VLAN 3.

FIGURE 197 Using multi-device port authentication with dynamic VLAN assignment



In this example, multi-device port authentication is performed for both devices. If the PC is successfully authenticated, dual-mode port e1 PVID is changed from the VLAN 1 (the DEFAULT-VLAN) to VLAN 102. If authentication for the PC fails, then the PC can be placed in a specified "restricted" VLAN, or traffic from the PC can be blocked in hardware. In this example, if authentication for the PC fails, the PC would be placed in VLAN 1023, the restricted VLAN.

If authentication for the IP phone is successful, then dual-mode port e1 is added to VLAN 3. If authentication for the IP phone fails, then traffic from the IP phone would be blocked in hardware. (Devices sending tagged traffic cannot be placed in the restricted VLAN.)

NOTE

This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Brocade device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e1) is not a member of that VLAN, authentication would not occur. In this case, port e1 must be added to that VLAN prior to authentication.

The part of the running-config related to multi-device port authentication would be as follows.

Example port authentication configurations

```
mac-authentication enable
mac-authentication auth-fail-vlan-id 1023

interface ethernet 1
  mac-authentication enable
  mac-authentication auth-fail-action restrict-vlan
  mac-authentication enable-dynamic-vlan
  dual-mode
```

Examples of multi-device port authentication and 802.1X authentication configuration on the same port

The following examples show configurations that use multi-device port authentication and 802.1X authentication on the same port.

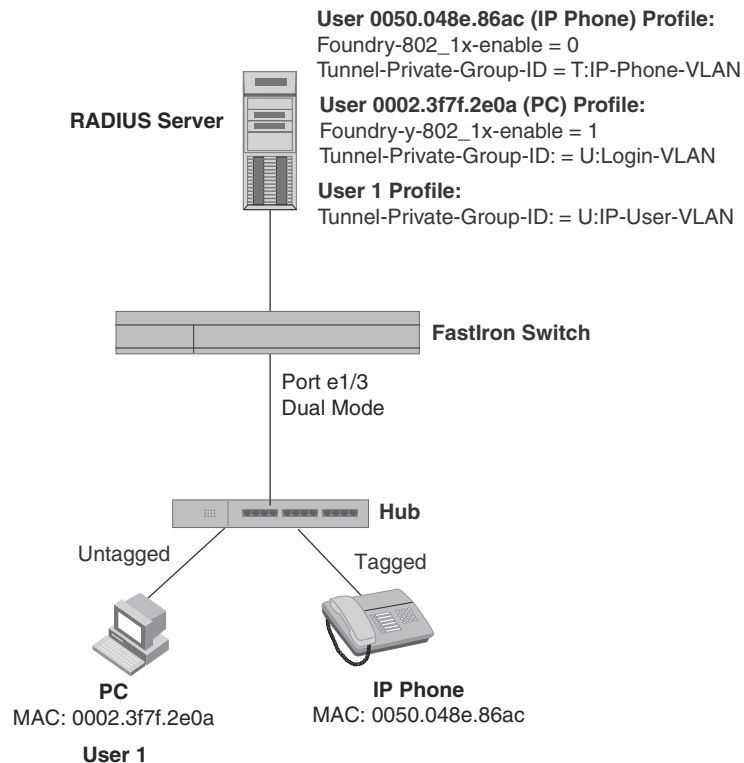
Example 1 – Multi-device port authentication and 802.1x authentication on the same port

[Figure 198](#) illustrates an example configuration that uses multi-device port authentication and 802.1X authentication on the same port. In this configuration, a PC and an IP phone are connected to port e 1/3 on a Brocade device. Port e 1/3 is configured as a dual-mode port.

The profile for the PC MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN "Login-VLAN", and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to the VLAN named "IP-Phone-VLAN". When User 1 is successfully authenticated using 802.1X authentication, the PC is then placed in the VLAN named "User-VLAN".

NOTE

This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Brocade device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/3) is not a member of that VLAN, authentication would not occur. In this case, port e 1/3 must be added to that VLAN prior to authentication.

FIGURE 198 Using multi-device port authentication and 802.1X authentication on the same port

When the devices attempt to connect to the network, they are first subject to multi-device port authentication.

When the MAC address of the IP phone is authenticated, the Access-Accept message from the RADIUS server specifies that the IP phone port be placed into the VLAN named “IP-Phone-VLAN”, which is VLAN 7. The Foundry-802_1x-enable attribute is set to 0, meaning that 802.1X authentication is skipped for this MAC address. Port e 1/3 is placed in VLAN 7 as a tagged port. No further authentication is performed.

When the PC MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for the PC port be changed to the VLAN named “Login-VLAN”, which is VLAN 1024. The Foundry-802_1x-enable attribute is set to 1, meaning that 802.1X authentication is required for this MAC address. The PVID of the port e 1/3 is temporarily changed to VLAN 1024, pending 802.1X authentication.

When User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for User 1 port be changed to the VLAN named “User-VLAN”, which is VLAN 3. If 802.1X authentication for User 1 is unsuccessful, the PVID for port e 1/3 is changed to that of the restricted VLAN, which is 1023, or untagged traffic from port e 1/3 can be blocked in hardware.

The part of the running-config related to port e 1/3 would be as follows.

```
interface ethernet 1/3
dot1x port-control auto
mac-authentication enable
dual-mode
```

When the PC is authenticated using multi-device port authentication, the port PVID is changed to “Login-VLAN”, which is VLAN 1024 in this example.

When User 1 is authenticated using 802.1X authentication, the port PVID is changed to “User-VLAN”, which is VLAN 3 in this example.

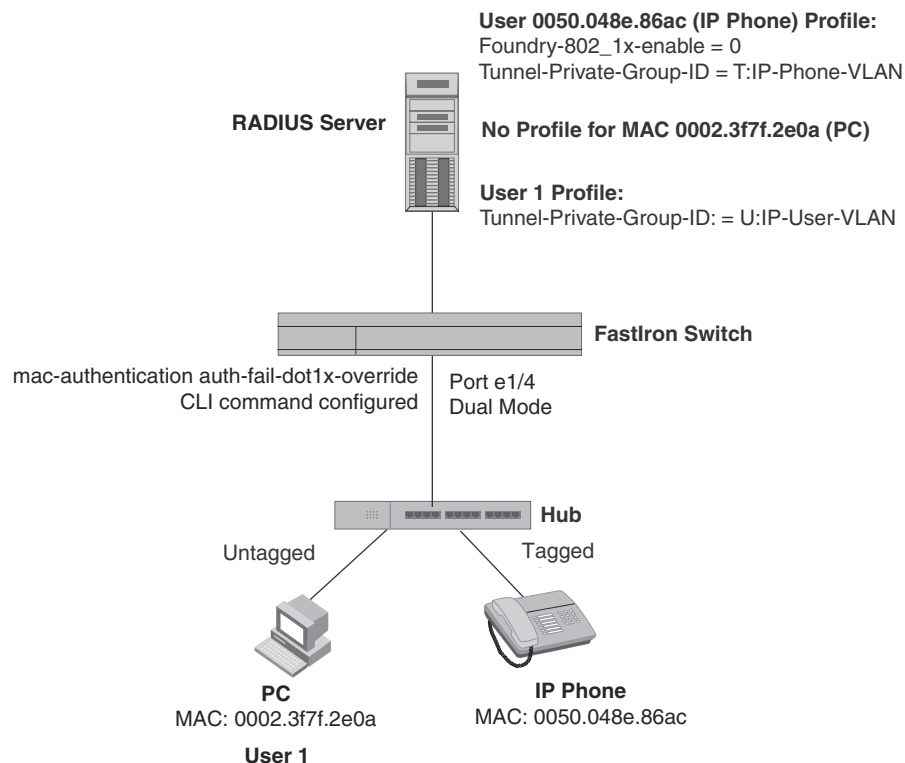
Example 2 – Creating a profile on the RADIUS server for each MAC address

The configuration in [Figure 199](#) requires that you create a profile on the RADIUS server for each MAC address to which a device or user can connect to the network. In a large network, this can be difficult to implement and maintain.

As an alternative, you can create MAC address profiles only for those devices that do not support 802.1X authentication, such as IP phones and printers, and configure the device to perform 802.1X authentication for the other devices that do not have MAC address profiles, such as user PCs. To do this, you configure the device to perform 802.1X authentication when a device fails multi-device port authentication.

[Figure 199](#) shows a configuration where multi-device port authentication is performed for an IP phone, and 802.1X authentication is performed for a user PC. There is a profile on the RADIUS server for the IP phone MAC address, but not for the PC MAC address.

FIGURE 199 802.1X Authentication is performed when a device fails multi-device port authentication



Multi-device port authentication is initially performed for both devices. The IP phone MAC address has a profile on the RADIUS server. This profile indicates that 802.1X authentication should be skipped for this device, and that the device port be placed into the VLAN named “IP-Phone-VLAN”.

Since there is no profile for the PC MAC address on the RADIUS server, multi-device port authentication for this MAC address fails. Ordinarily, this would mean that the PVID for the port would be changed to that of the restricted VLAN, or traffic from this MAC would be blocked in hardware. However, the device is configured to perform 802.1X authentication when a device fails multi-device port authentication, so when User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the PVID for port e 1/4 is changed to the VLAN named "User-VLAN".

NOTE

This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Brocade device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/4) is not a member of that VLAN, authentication would not occur. In this case, port e 1/4 must be added to that VLAN prior to authentication.

To configure the device to perform 802.1X authentication when a device fails multi-device port authentication, enter the following command.

```
Brocade(config)#mac-authentication auth-fail-dot1x-override
```

Syntax: [no] mac-authentication auth-fail-dot1x-override

Example port authentication configurations

Web Authentication

[Table 315](#) lists individual Brocade switches and the Web Authentication features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 315 Supported Web Authentication features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Enabling and disabling Web Authentication	Yes	Yes	Yes	Yes	Yes
Configuring the Web Authentication mode	Yes	Yes	Yes	Yes	Yes
Web Authentication options in this chapter	Yes	Yes	Yes	Yes	Yes

Web authentication overview

Authentication is important in enterprise networks because the network is considered a secure area: it contains sensitive data and a finite amount of resources. Unauthorized users must be prevented from accessing the network to protect the sensitive data and prevent the unnecessary consumption of resources.

The ideal authentication method blocks unauthorized users at the earliest possible opportunity. For internal enterprise networks, this can be controlled at the edge switch port. Two popular forms of port-based security authentication used at the edge switch are multi-device port authentication and 802.1x. Multi-device port authentication authenticates the MAC addresses of hosts or users that are attempting to access the network. This type of authentication requires no intervention from the host or user who is attempting to be authenticated. It is easy to use, but it can only authorize hosts; it cannot be used to authorize users. 802.1x authentication can authorize users or hosts. It is more flexible than the multi-device port authentication method; however, it requires more support, configuration, maintenance and user intervention than multi-device port authentication.

The Brocade Web authentication method provides an ideal port-based authentication alternative to multi-device port authentication without the complexities and cost of 802.1x authentication. Hosts gain access to the network by opening a Web browser and entering a valid URL address using HTTP or HTTPS services. Instead of being routed to the URL, the host browser is directed to an authentication Web page on the FastIron switch. The Web page prompts the host to enter a user ID and password or a passcode. The credentials a host enters are used by a trusted source to authenticate the host MAC address. (Multiple MAC addresses can be authenticated with the same user name and password.)

If the authentication is unsuccessful, the appropriate page is displayed on the host browser. The host is asked to try again or call for assistance, depending on what message is configured on the Web page. If the host MAC address is authenticated by the trusted source, a Web page is displayed with a hyperlink to the URL the host originally entered. If the user clicks on the link, a new window is opened and the user is directed to the requested URL.

While a MAC address is in the authenticated state, the host can forward data through the FastIron switch. The MAC address remains authenticated until one of the following events occurs:

- The host MAC address is removed from a list of MAC addresses that are automatically authenticated. (Refer to [“Specifying hosts that are permanently authenticated”](#) on page 1902).
- The re-authentication timer expires and the host is required to re-authenticate (Refer to [“Configuring the re-authentication period”](#) on page 1903).
- The host has remained inactive for a period of time and the inactive period timer has expired. (Refer to [“Forcing re-authentication after an inactive period”](#) on page 1906.)
- All the ports on the VLAN on which Web Authentication has been configured are in a down state. All MAC addresses that are currently authenticated are de-authenticated (Refer to [“Forcing re-authentication when ports are down”](#) on page 1905.)
- The authenticated client is cleared from the Web Authentication table. (Refer to [“Clearing authenticated hosts from the web authentication table”](#) on page 1903).

The FastIron switch can be configured to automatically authenticate a host MAC address. The host will not be required to login or re-authenticate (depending on the re-authentication period) once the MAC address passes authentication.

A host that is logged in and authenticated remains logged in indefinitely, unless a re-authentication period is configured. When the re-authentication period ends, the host is logged out. A host can log out at any time by pressing the Logout button in the Web Authentication Success page.

NOTE

The host can log out as long as the Logout window (Success page) is visible. If the window is accidentally closed, the host cannot log out unless the re-authentication period ends or the host is manually cleared from the Web Authentication table.

Web authentication configuration considerations

Web Authentication is modeled after other RADIUS-based authentication methods currently available on Brocade edge switches. However, Web Authentication requires a Layer 3 protocol (TCP/IP) between the host and the authenticator. Therefore, to implement Web Authentication, you must consider the following configuration and topology configuration requirements:

- Web authentication works only when both the HTTP and HTTPS servers are enabled on the device.
- Web Authentication works only on the default HTTP or HTTPS port.
- The host must have an IP address prior to Web Authentication. This IP address can be configured statically on the host; however, DHCP addressing is also supported.
- If you are using DHCP addressing, a DHCP server must be in the same broadcast domain as the host. This DHCP server does not have to be physically connected to the switch. Also, DHCP assist from a router may be used.

- Web Authentication, 802.1X port security, and multi-device port authentication are not supported concurrently on the same port.
- Web Authentication is not supported on an MCT VLAN.

The following applies to Web Authentication in the Layer 2 switch image:

- If the management VLAN and Web Authentication VLAN are in different IP networks, make sure there is at least one routing element in the network topology that can route between these IP networks.

The following are required for Web Authentication in the base Layer 3 and full Layer 3 images:

- Each Web Authentication VLAN must have a virtual interface (VE).
- The VE must have at least one assigned IPv4 address.

Web Authentication is enabled on a VLAN. That VLAN becomes a Web Authentication VLAN that does the following:

- Forwards traffic from authenticated hosts, just like a regular VLAN.
- Blocks traffic from unauthenticated hosts except from ARP, DHCP, DNS, HTTP, and HTTPs that are required to perform Web Authentication.

Figure 200 shows the basic components of a network topology where Web Authentication is used. You will need:

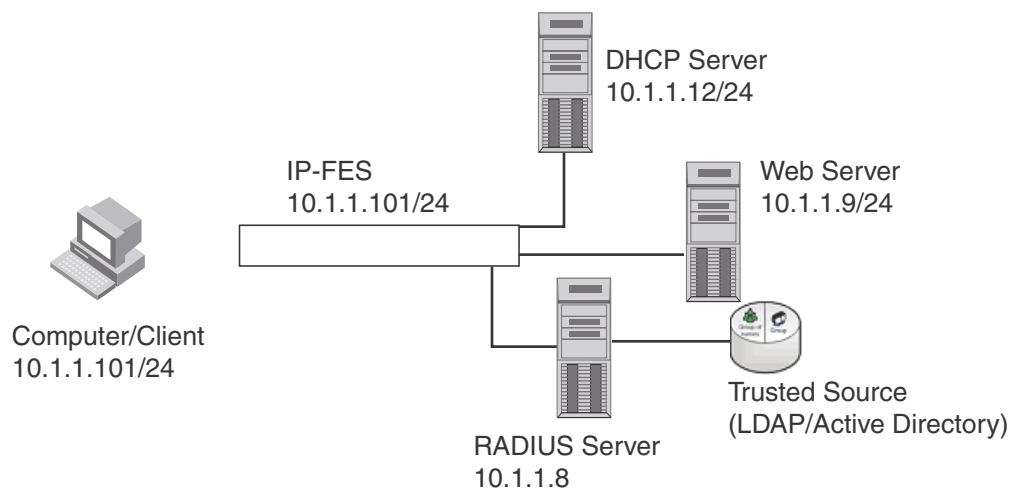
- A Brocade FastIron switch running a software release that supports Web Authentication
- DHCP server, if dynamic IP addressing is to be used
- Computer/host with a web browser

Your configuration may also require a RADIUS server with some Trusted Source such as LDAP or Active Directory.

NOTE

The Web server, RADIUS server, and DHCP server can all be the same server.

FIGURE 200 Basic topology for web authentication



Web authentication configuration tasks

Follow the steps given below to configure Web Authentication on a device.

1. Set up any global configuration required for the FastIron switch, RADIUS server, Web server and other servers.
 - On a Layer 2 FastIron switch, make sure the FastIron switch has an IP address.

```
Brocade# configure terminal
Brocade(config)#ip address 10.1.1.10/24
```

- On a Layer 3 FastIron switch, assign an IP address to a virtual interface (VE) for each VLAN on which Web Authentication will be enabled.

```
Brocade#configure terminal
Brocade(config)#vlan 10
Brocade(config-vlan-10)#router-interface ve1
Brocade(config-vlan-10)#untagged e 1/1/1 to 1/1/10
Brocade(config-vlan-10)#interface ve1
Brocade(config-vif-1)#ip address 1.1.2.1/24
```

2. By default, Web Authentication will use a RADIUS server to authenticate host usernames and passwords, unless it is configured to use a local user database. If Web Authentication will use a RADIUS server, you must configure the RADIUS server and other servers. For example, if your RADIUS server has an IP address of 192.168.1.253, then use the CLI to configure the following global CLI commands on the FastIron switch.

```
Brocade(config)# radius-server host 10.1.1.8
Brocade(config)# radius-server key $GSig@U\
```

NOTE

Remember the RADIUS key you entered. You will need this key when you configure your RADIUS server.

3. Web authentication can be configured to use secure (HTTPS) or non-secure (HTTP) login and logout pages. By default, HTTPS is used.

To enable the non-secure Web server on the FastIron switch, enter the following command.

```
Brocade(config)# web-management HTTP
Brocade(config)#vlan 10
Brocade(config-vlan-10)webauth
Brocade(config-vlan-10-webauth)#no secure-login
```

To enable the secure Web server on the FastIron switch, enter the following command.

```
Brocade(config)# web-management HTTPS
Brocade(config)#vlan 10
Brocade(config-vlan-10)webauth
Brocade(config-vlan-10-webauth)#secure-login
```

4. If the secure Web server is used, in order to access a secure Web page, the Web server needs to provide a key. This key is exchanged using a certificate. A certificate is a digital document that is issued by a trusted source that can validate the authenticity of the certificate and the Web server that is presenting it. Therefore the switch must have a certificate for web authentication to work. There are two choices for providing the switch with a certificate:

- Upload one using the following global CLI command.


```
Brocade(config)# ip ssl private-key-file tftp <ip-addr> <key-filename>
```

- Generate one using the following global CLI command.

```
Brocade(config)#crypto-ssl certificate generate default_cert
```

5. Create a Web Authentication VLAN and enable Web Authentication on that VLAN.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#webauth
Brocade(config-vlan-10-webauth)#enable
```

Once enabled, the CLI changes to the "webauth" configuration level. In the example above, VLAN 10 will require hosts to be authenticated using Web Authentication before they can forward traffic.

6. Configure the Web Authentication mode:

- **Username and password** – Blocks users from accessing the switch until they enter a valid username and password on a web login page.
- **Passcode** – Blocks users from accessing the switch until they enter a valid passcode on a web login page.
- **None** – Blocks users from accessing the switch until they press the 'Login' button. A username and password or passcode is not required.

Refer to [“Web authentication mode configuration”](#) on page 1892.

7. Configure other Web Authentication options (refer to [“Web authentication options configuration”](#) on page 1901).

Enabling and disabling web authentication

Web Authentication is disabled by default. To enable it, enter the following commands.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#webauth
Brocade(config-vlan-10-webauth)#enable
```

The first command changes the CLI level to the VLAN configuration level. The second command changes the configuration level to the Web Authentication VLAN level. The last command enables Web Authentication. In the example above, VLAN 10 will require hosts to be authenticated using Web Authentication before they can forward traffic.

Syntax: webauth

FastIron devices support a maximum of two Web Authentication VLANs.

Syntax: [no] enable

Enter the **no enable** command to disable Web Authentication.

Web authentication mode configuration

You can configure the FastIron switch to use one of three Web Authentication modes:

- **Username and password** – Block users from accessing the switch until they enter a valid username and password on a web login page. Refer to [“Using local user databases”](#) on page 1892.
- **Passcode** – Blocks users from accessing the switch until they enter a valid passcode on a web login page. Refer to [“Passcodes for user authentication”](#) on page 1896.
- **None** – Blocks users from accessing the switch until they press the 'Login' button. A username and password or passcode is not required. Refer to [“Automatic authentication”](#) on page 1900.

The following sections describe how to configure these Web Authentication modes.

Using local user databases

Web Authentication supports the use of local user databases consisting of usernames and passwords, to authenticate devices. Users are blocked from accessing the switch until they enter a valid username and password on a web login page.

Once a user successfully authenticates through username and password, the user is subjected to the same policies as for RADIUS-authenticated devices (for example, the re-authentication period, maximum number of users allowed, etc.). Similarly, once a user fails username and password authentication, the user is subjected to the same policies as for devices that fail RADIUS authentication.

You can create up to ten local user databases on the FastIron switch either by entering a series of CLI commands, or by uploading a list of usernames and passwords from a TFTP file to the FastIron switch. The user databases are stored locally, on the FastIron switch.

Configuring a local user database

Follow the steps given below to configure a local user database.

1. Create the local user database.
2. Add records to the local user database either by entering a series of CLI commands, or by importing a list of user records from an ASCII text file on the TFTP server to the FastIron switch.
3. Set the local user database authentication mode.
4. If desired, set the authentication method (RADIUS/local) failover sequence.
5. Assign a local user database to a Web Authentication VLAN.

Creating a local user database

The FastIron switch supports a maximum of ten local user databases, each containing up to 30 user records. Each user record consists of a username and password.

To create a local user database, enter a command such as the following.

```
Brocade#(config)#local-userdb userdb1
Brocade#(config-localuserdb-userdb1)#
```

This command creates a local user database named **userdb1**. To add user records to this database, refer to [“Adding a user record to a local user database”](#) on page 1893.

Syntax: **local-userdb** <db-name>

You can create up to ten local user databases for Web Authentication.

For <db-name>, enter up to 31 alphanumeric characters.

Adding a user record to a local user database

To add a user record, enter commands such as the following.

```
Brocade#(config)#local-userdb userdb1
Brocade#(config-localuserdb-userdb1)#username marcia password bunch4
```

The first command changes the configuration level to the local user database level for **userdb1**. If the database does not already exist, it is created. The second command adds the user record **marcia** to the **userdb1** database.

Syntax: **username** <username> **password** <password>

For <username>, enter up to 31 ASCII characters.

For <password>, enter up to 29 ASCII characters.

You can add up to 30 usernames and passwords to a local user database.

To view a list of users in a local user database, use the CLI command **vlan-mod-port-userdb**. Refer to [“Displaying a list of local user databases”](#) on page 1919.

Deleting a user record from a local user database

To delete a user record from the local user database, enter commands such as the following.

```
Brocade#(config)#local-userdb userdb1
Brocade#(config-localuserdb-userdb1)#no username marcia
```

The first command changes the configuration level to the local user database level for **userdb1**. The second command deletes the user record **marcia** from the **userdb1** database.

Syntax: **no username** <username>

Deleting All user records from a local user database

To delete all user records from a local user database, enter the **delete-all** command.

```
Brocade#(config-localuserdb-userdb1)#delete-all
```

Syntax: **delete-all**

Creating a text file of user records

If desired, you can use the TFTP protocol to import a list of usernames and passwords from a text file on a TFTP server to the FastIron switch. The text file to be imported must be in the following ASCII format.

```
[delete-all]
[no] username <username1> password <password1> <cr>
```

```
[no] username <username2> password <password2> <cr>
...
```

The [delete-all] keyword indicates that the user records in the text file will replace the user records in the specified local user database on the FastIron switch. If the [delete-all] keyword is not present, the new user records will be *added* to the specified local user database on the FastIron switch. The [delete-all] keyword is optional. If present, it must appear on the first line, before the first user record in the text file.

The optional [no] keyword indicates that the user entry will be deleted from the specified local user database on the FastIron switch.

User records that already exist in the local user database will be updated with the information in the text file when it is uploaded to the switch.

For <username1>, <username2>, etc., enter up to 31 ASCII characters.

For <password1>, <password2>, etc., enter up to 29 ASCII characters.

Be sure to Insert a cursor return (<cr>) after each user record.

You can enter up to 30 user records per text file.

Importing a text file of user records from a TFTP server

NOTE

Before importing the file, make sure it adheres to the ASCII text format described in the previous section, [“Creating a text file of user records”](#) on page 1893.

To import a text file of user records from a TFTP server to the FastIron switch, enter a command such as the following.

```
Brocade#(config-localuserdb-userdb1)#import-users tftp 192.168.1.1 filename
userdb1
```

Syntax: `import-users tftp <ip-address> filename <filename>`

The <ip-address> parameter specifies the IPv4 address of the TFTP server on which the desired text file resides.

The <filename> parameter specifies the name of the image on the TFTP server.

Using a RADIUS server as the web authentication method

By default, Web Authentication will use a RADIUS server to authenticate hosts' usernames and passwords, unless the device is configured to use the local user database (see the previous section). To configure the FastIron switch to use a RADIUS server, refer to [“RADIUS security”](#) on page 157. You must also perform the following steps.

1. Configure the RADIUS server information on the FastIron switch. Enter a command such as the following.

```
Brocade(config)#radius-server host 10.1.1.8 auth-port 1812 acct-port 1813
default key $GSig@U\
```

NOTE

Web Authentication will use the first reachable RADIUS server listed in the configuration. The **use-radius-server** on individual ports is not supported for Web Authentication.

2. Enable the username and password authentication mode.

```
Brocade(config-vlan-10-webauth)#auth-mode username-password
```

3. Enable the RADIUS authentication method. Refer to [“Setting the local user database authentication method”](#) on page 1895 or [“Setting the web authentication failover sequence”](#) on page 1895

Setting the local user database authentication method

By default, the FastIron switch uses a RADIUS server to authenticate users in a VLAN. The previous section describes how to configure a RADIUS server to authenticate users in a VLAN. To configure the switch to instead use a local user database to authenticate users in a VLAN, enter the following command.

```
Brocade(config-vlan-10-webauth)#auth-mode username-password auth-methods local
```

Syntax: `auth-mode username-password auth-methods local`

To revert back to using the RADIUS server, enter the following command.

```
Brocade(config-vlan-10-webauth)#auth-mode username-password auth-methods radius
```

Syntax: `auth-mode username-password auth-methods radius`

Setting the web authentication failover sequence

You can optionally specify a failover sequence for RADIUS and local user database authentication methods. For example, you can configure Web Authentication to first use a local user database to authenticate users in a VLAN. If the local user database is not available, it will use a RADIUS server. Enter the following command.

```
Brocade(config-vlan-10-webauth)#auth-mode username-password auth-methods local radius
```

Syntax: `auth-mode username-password auth-methods <method1> <method2>`

For `<method1>` `<method2>`, enter **radius local** or **local radius**.

Assigning a local user database to a web authentication VLAN

After creating or importing a local user database on the FastIron switch and setting the local user database authentication method to **local**, you can configure a Web Authentication VLAN to use the database to authenticate users in a VLAN. To do so, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#auth-mode username-password local-user-database userdb1
```

These commands configure Web Authentication to use the usernames and passwords in the userdb1 database to authenticate users in VLAN 10.

Syntax: `[no] auth-mode username-password local-user-database <db-name>`

For `<db-name>`, enter a valid local user database.

Use the **no** form of the command to remove the database from the Web Authentication VLAN.

Passcodes for user authentication

Web Authentication supports the use of passcodes to authenticate users. Users are blocked from accessing the switch until they enter a valid passcode on a web login page. Unlike username and password authentication, passcode authentication uses a simple number to authenticate users. The simplicity of a passcode reduces user errors and lowers the overhead of supporting and managing simple tasks, such as Internet access for guests and visitors in the office.

When passcodes are enabled, the system will automatically generate them every 1440 minutes (24 hours), and when the system boots up. You can optionally create up to four static passcodes which will be used in conjunction with the dynamic passcodes generated by the system.

Configuring passcode authentication

Follow the steps given below to configure the device to use the passcode authentication mode.

1. Optionally create up to four static passcodes
2. Enable passcode authentication
3. Configure other options

Creating static passcodes

Static passcodes can be used for troubleshooting purposes, or for networks that want to use passcode authentication, but do not have the ability to support automatically-generated passcodes (for example, the network does not fully support the use of SNMP traps or Syslog messages with passcodes).

Manually-created passcodes are used in conjunction with dynamic passcodes. You can configure up to four static passcodes that never expire. Unlike dynamically-created passcodes, static passcodes are saved to flash memory. By default, there are no static passcodes configured on the switch.

To create static passcodes, enter commands such as the following.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode static 3267345
Brocade(config-vlan-10-webauth)#auth-mode passcode static 56127
```

Syntax: `auth-mode passcode static <passcode>`

For `<passcode>`, enter a number from 4 to 16 digits in length. You can create up to four static passcodes, each with a different length. Static passcodes do not have to be the same length as passcodes that are automatically generated.

After creating static passcodes, you can enable passcode authentication as described in the next section.

To view the passcodes configured on the switch, use the **show webauth vlan <vlan-id> passcode** command. Refer to [“Displaying passcodes”](#) on page 1920.

Enabling passcode authentication

To enable passcode authentication, enter the following command.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode
```

This command enables Web Authentication to use dynamically-created passcodes to authenticate users in the VLAN. If the configuration includes static passcodes, they are used in conjunction with dynamically-created passcodes.

Syntax: [no]auth-mode passcode

Enter **no auth-mode passcode** to disable passcode authentication.

Configuring the length of dynamically-generated passcodes

By default, dynamically-generated passcodes are 4 digits in length, for example, 0123. If desired, you can increase the passcode length to up to 16 digits. To do so, enter a command such as the following at the Web Authentication level of the CLI.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode length 10
```

The next dynamically-created passcode will be 10 digits in length, for example, 0123456789.

Syntax: auth-mode passcode length <value>

For <value>, enter a number from 4 to 16.

Configuring the passcode refresh method

Passcode authentication supports two passcode refresh methods:

- **Duration of time** – By default, dynamically-created passcodes are refreshed every 1440 minutes (24 hours). When refreshed, a new passcode is generated and the old passcode expires. You can increase or decrease the *duration of time* after which passcodes are refreshed, or you can configure the device to refresh passcodes at a certain *time of day* instead of after a duration of time.
- **Time of day** – When initially enabled, the *time of day* method will cause passcodes to be refreshed at 0:00 (12:00 midnight). If desired, you can change this time of day, and you can add up to 24 refresh periods in a 24-hour period.

When a passcode is refreshed, the old passcode will no longer work, unless a grace period is configured (refer to [“Configuring a grace period for an expired passcode”](#) on page 1898).

If a user changes the passcode refresh value, the configuration is immediately applied to the current passcode. For example, if the passcode duration is 100 minutes and the passcode was last generated 60 minutes prior, a new passcode will be generated in 40 minutes. However, if the passcode duration is changed from 100 to 75 minutes, and the passcode was last generated 60 minutes prior, a new passcode will be generated in 15 minutes. Similarly, if the passcode duration is changed from 100 to 50 minutes, and the passcode was last generated 60 minutes prior, the passcode will immediately expire and a new passcode will be generated. The same principles apply to the time of day passcode refresh method.

If you configure both *duration of time* and *time of day* passcode refresh values, they are saved to the configuration file. You can switch back and forth between the passcode refresh methods, but only one method can be enabled at a time.

NOTE

Passcodes are not stateful, meaning a software reset or reload will cause the system to erase the passcode. When the FastIron switch comes back up, a new passcode will be generated.

Changing the passcode refresh duration

To change the duration of time after which passcodes are refreshed, enter commands such as the following.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode refresh-type duration 4320
```

The passcode will be refreshed after 4320 minutes (72 hours).

Syntax: `auth-mode passcode refresh-type duration <value>`

For <value>, enter a number from 5 to 9999 minutes. The default is 1440 minutes (24 hours).

Refreshing passcodes at a certain time of the day

You can configure the FastIron switch to refresh passcodes at a certain *time of day*, up to 24 times each day, instead of after a duration of time. When this feature is enabled, by default passcodes will be refreshed at 00:00 (12 midnight).

To configure the switch to refresh passcodes at a certain time of day, enter commands such as the following.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode refresh-type time 6:00
Brocade(config-vlan-10-webauth)#auth-mode passcode refresh-type time 14:30
```

The passcode will be refreshed at 6:00am, 2:30pm, and 0:00 (12 midnight).

Syntax: `[no] auth-mode passcode refresh-type time <hh:mm>`.

<hh:mm> is the hour and minutes. If you do not enter a value for <hh:mm>, by default, passcodes will be refreshed at 00:00 (12:00 midnight). You can configure up to 24 refresh times. Each must be at least five minutes apart.

Enter the **no** form of the command to remove the passcode refresh time of day.

Resetting the passcode refresh time of day configuration

If the FastIron switch is configured to refresh passcodes several times during the day (*time of day* configuration), you can use the following command to delete all of the configured times and revert back to the default time of 00:00 (12 midnight).

```
Brocade(config-vlan-10-webauth)#auth-mode passcode refresh-type time delete-all
```

Syntax: `auth-mode passcode refresh-type time delete-all`

Configuring a grace period for an expired passcode

You can optionally configure a grace period for an expired passcode. The grace period is the period of time that a passcode will remain valid, even after a new passcode is generated. For example, if a five minute grace period is set and the passcode 1234 is refreshed to 5678, both passcodes will be valid for five minutes, after which the 1234 passcode will expire and the 5678 passcode will remain in effect.

To configure the grace period for an expired passcode, enter a command such as the following.


```
Brocade(config-vlan-10-webauth)#auth-mode passcode grace-period 5
```

Syntax: `auth-mode passcode grace-period <value>`

`<value>` is a number between 0 and 5 minutes. 0 means there is no grace period.

NOTE

If the grace period is re-configured while a passcode is already in the grace period, the passcode is not affected by the configuration change. The new grace period will apply only to passcodes that expire *after* the new grace period is set.

Flushing all expired passcodes that are in the grace period

You can delete old passcodes that have expired but are still valid because they are in the grace period. This feature is useful in situations where the old passcodes have been compromised but are still valid because of the grace period. This feature does not affect current valid passcodes or passcodes that newly expire.

To flush out all expired passcodes that are currently in the grace period, enter the following command.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode flush-expired
```

Syntax: `auth-mode passcode flush-expired`

Disabling and re-enabling passcode logging

The software generates a Syslog message and SNMP trap message every time a new passcode is generated and passcode authentication is attempted,. This is the default behavior. If desired, you can disable passcode-related Syslog messages or SNMP trap messages, or both.

The following shows an example Syslog message and SNMP trap message related to passcode authentication.

```
New passcode: 01234567. Expires in 1440 minutes. Old passcode is valid for another 5 minutes.
```

To disable Syslog messages for passcodes, enter the **no auth-mode passcode log syslog** command.

```
Brocade(config-vlan-10-webauth)#no auth-mode passcode log syslog
```

Enter the following command to disable SNMP trap messages for passcodes.

```
Brocade(config-vlan-10-webauth)#no auth-mode passcode log snmp-trap
```

Enter the following command to re-enable Syslog messages for passcodes after they have been disabled.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode log syslog
```

Enter the following command to re-enable SNMP trap messages for passcodes after they have been disabled.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode log snmp-trap
```

Syntax: `[no] auth-mode passcode log syslog | snmp-trap`

Re-sending the passcode log message

If passcode logging is enabled, you can enter a CLI command to retransmit the current passcode to a Syslog message or SNMP trap. To do so, enter the **auth-mode passcode resend-log** command.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode resend-log
```

Syntax: **auth-mode passcode resend-log**

NOTE

The switch retransmits the current passcode only. Passcodes that are in the grace period are not sent.

Manually refreshing the passcode

You can manually refresh the passcode instead of waiting for the system to automatically generate one. When manually refreshed, the old passcode will no longer work, even if a grace period is configured. Also, if the passcode refresh method *duration of time* is used, the duration counter is reset when the passcode is manually refreshed. The passcode refresh method *time of day* is not affected when the passcode is manually refreshed.

To immediately refresh the passcode, enter the **auth-mode passcode generate** command.

```
Brocade(config-vlan-10-webauth)#auth-mode passcode generate
```

Syntax: **auth-mode passcode generate**

Automatic authentication

By default, if Web Authentication is enabled, hosts need to login and enter authentication credentials in order to gain access to the network. If a re-authentication period is configured, the host will be asked to re-enter authentication credentials once the re-authentication period ends.

You can configure Web Authentication to authenticate a host when the user presses the 'Login' button. When a host enters a valid URL address, Web Authentication checks the list of blocked MAC addresses. If the hosts' MAC address is not on the list and the number of allowable hosts has not been reached, after pressing the 'Login' button, the host is automatically authenticated for the duration of the configured re-authentication period, if one is configured. Once the re-authentication period ends, the host is logged out and needs to enter the URL address again.

NOTE

Automatic authentication is not the same as permanent authentication. (Refer to [“Specifying hosts that are permanently authenticated”](#) on page 1902). You must still specify devices that are to be permanently authenticated even if automatic authentication is enabled.

To enable automatic authentication, enter the following command.

```
Brocade(config)#vlan 10
Brocade(config-vlan-10)#webauth
Brocade(config-vlan-10-webauth)#auth-mode none
```

Syntax: **[no] auth-mode none**

If automatic authentication is enabled and a host address is not in the blocked MAC address list, Web Authentication authenticates the host and displays the Login page without user credentials, then provides a hyperlink to the requested URL site..

To determine if automatic authentication is enabled on your device, issue the **show webauth vlan <vlan-id>** command at the VLAN configuration level.

Syslog messages are generated under the following conditions:

- The feature is enabled
- The feature is disabled
- A MAC address is successfully authenticated
- Automatic authentication cannot occur because the maximum number of hosts allowed has been reached

Web authentication options configuration

The sections below explain other configuration options for Web Authentication.

Enabling RADIUS accounting for web authentication

When Web Authentication is enabled, you can enable RADIUS accounting to record login (start) and logout (stop) events per host. The information is sent to a RADIUS server. Note that packet/byte count is not supported.

To enable RADIUS accounting, enter the **accounting** command.

```
Brocade(config-vlan-10-webauth)#accounting
```

Syntax: [no] **accounting**

Enter the **no accounting** command to disable RADIUS accounting for Web Authentication.

Changing the login mode (HTTPS or HTTP)

Web Authentication can be configured to use secure (HTTPS) or non-secure (HTTP) login and logout pages. By default, HTTPS is used. [Figure 202](#) shows an example Login page.

To change the login mode to non-secure (HTTP), enter the **no secure-login** command.

```
Brocade(config-vlan-10-webauth)#no secure-login
```

To revert back to secure mode, enter the **secure-login** command.

```
Brocade(config-vlan-10-webauth)#secure-login
```

Syntax: [no] **secure-login**

Specifying trusted ports

You can configure certain ports of a Web Authentication VLAN as trusted ports. All hosts connected to the trusted ports need not authenticate and are automatically allowed access to the network.

To create a list of trusted ports, enter commands such as the following.

```
Brocade(config-vlan-10-webauth)#trust-port ethernet 3
Brocade(config-vlan-10-webauth)#trust port ethernet 6 to 10
```

The above commands configure ports 3 and 6 – 10 as trusted ports.

Syntax: `trust-port ethernet <port> [to <port>]`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Specifying hosts that are permanently authenticated

Certain hosts, such as DHCP server, gateway, printers, may need to be permanently authenticated. Typically, these hosts are managed by the network administrator and are considered to be authorized hosts. Also, some of these hosts (such as printers) may not have a Web browser and will not be able to perform the Web Authentication.

To permanently authenticate these types of hosts, enter a command such as the following at the "webauth" configuration level.

```
Brocade(config-vlan-10-webauth)#add mac 0004.80eb.2d14 duration 0
Brocade(config-vlan-10-webauth)#add mac 0007.e90e.de3b duration 0
```

Syntax: `[no] add mac <mac-address> duration <seconds> | ethernet <port> duration <seconds>`

Syntax: `no add mac <mac-address>`

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

<seconds> specifies how long the MAC address remains authenticated. Enter 0 – 128000 seconds. The default is the current value of **reauth-time**. A value of "0" means that Web Authentication for the MAC address will not expire.

Instead of just entering a duration for how long the MAC address remains authenticated, you can specify the MAC address to be added by the specified port that is a member of the VLAN. To do this, enter values for the **ethernet <port> duration <seconds>** option. Enter the port number and the number of seconds the MAC address remains authenticated.

Entering a **no add mac <mac-address> duration <seconds> | ethernet <port> duration <seconds>** command sets duration and ethernet to their default values. If you want to remove a host, enter the **no add mac <mac-address>** command.

NOTE

If a MAC address is statically configured, this MAC address will not be allowed to be dynamically configured on any port.

Configuring the re-authentication period

After a successful authentication, a user remains authenticated for a duration of time. At the end of this duration, the host is automatically logged off. The user must be re-authenticated again. To set the number of seconds a host remains authenticated before being logged off, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#reauth-time 10
```

Syntax: [no] **reauth-time** <seconds>

You can specify 0 – 128000 seconds. The default is 28800 seconds, and 0 means the user is always authenticated and will never have to re-authenticate, except if an inactive period less than the re-authentication period is configured on the Web Authentication VLAN. If this is the case, the user becomes de-authenticated if there is no activity and the timer for the inactive period expires.

Defining the web authentication cycle

You can set a limit as to how many seconds users have to be Web Authenticated by defining a cycle time. This time begins at a user first Login attempt on the Login page. If the user has not been authenticated successfully when this time expires, the user must enter a valid URL again to display the Web Authentication Welcome page.

To define a cycle time, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#cycle time 20
```

Syntax: [no] **cycle time** <seconds>

Enter 0 – 3600 seconds, where 0 means there is no time limit. The default is 600 seconds

Limiting the number of web authentication attempts

You can set a limit on the number of times a user enters an invalid user name and password during the specified cycle time. If the user exceeds the limit, the user is blocked for a duration of time, which is defined by the **block duration** command. Also, the Web browser will be redirected to the Exceeded Allowable Attempts webpage.

To limit the number of Web Authentication attempts, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#attempt-max-num 4
```

Syntax: [no] **attempt-max-num** <number>

Enter a number from 0 to 64, where 0 means there is no limit to the number of Web Authentication attempts. The default is 5.

Clearing authenticated hosts from the web authentication table

Use the following commands to clear dynamically-authenticated hosts from the Web Authentication table.

To clear all authenticated hosts in a Web authentication VLAN, enter a command such as the following.

```
Brocade#clear webauth vlan 25 authenticated-mac
```

This command clears all the authenticated hosts in VLAN 25.

To clear a particular host in a Web authentication VLAN, enter a command such as the following.

```
Brocade#clear webauth vlan 25 authenticated-mac 1111.2222.3333
```

This command clears host 1111.2222.3333 from VLAN 25.

Syntax: `clear webauth vlan <vlan-id> authenticated-mac [<mac-address>]`

Setting and clearing the block duration for web authentication attempts

After users exceed the limit for Web Authentication attempts, specify how many seconds users must wait before the next cycle of Web Authenticated begins. Enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#block duration 4
```

Syntax: `[no] block duration <seconds>`

Users cannot attempt Web Authentication during this time.

Enter 0–128000 seconds. The default is 90 seconds, and entering 0 means that the MAC address is infinitely blocked.

To unblock the MAC address, wait until the block duration timer expires or enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#clear webauth vlan 10 block-mac 000.000.1234
```

Syntax: `clear webauth vlan <vlan-id> block-mac [<mac-address>]`

If you do not enter a `<mac-address>`, then all the entries for the specified VLAN will be cleared.

Manually blocking and unblocking a specific host

A host can be temporarily or permanently blocked from attempting Web Authentication by entering a command such as the following.

```
Brocade(config-vlan-10-webauth)#block mac 0123.17d1.0a3d duration 4
```

Syntax: `[no] block mac <mac-address> duration <seconds>`

Syntax: `no block mac <mac-address>`

Enter 0 – 128000 for `<seconds>`. The default is the current value of **block duration** command. Entering a value of "0" means the MAC address is blocked permanently.

Entering **no block mac <mac-address> duration <seconds>** resets duration to its default value.

You can unblock a host by entering the **no block mac <mac-address>** command.

Limiting the number of authenticated hosts

You can limit the number of hosts that are authenticated at any one time by entering a command such as the following.

```
Brocade(config-vlan-10-webauth)#host-max-num 300
```

Syntax: [no] **host-max-num** <number>

You can enter 0 – 8192, where 0 means there is no limit to the number of hosts that can be authenticated. The default is 0. The maximum is 8192 or the maximum number of MAC addresses the device supports.

When the maximum number of hosts has been reached, the FastIron switch redirects any new host that has been authenticated successfully to the Maximum Host webpage.

Filtering DNS queries

Many of the Web Authentication solutions allow DNS queries to be forwarded from unauthenticated hosts. To eliminate the threat of forwarding DNS queries from unauthenticated hosts to unknown or untrusted servers (also known as domain-casting), you can restrict DNS queries from unauthenticated hosts to be forwarded explicitly to defined servers by defining DNS filters. Any DNS query from an unauthenticated host to a server that is not defined in a DNS filter are dropped. Only DNS queries from unauthenticated hosts are affected by DNS filters; authenticated hosts are not. If the DNS filters are not defined, then any DNS queries can be made to any server.

You can have up to four DNS filters. Create a filter by entering the following command.

```
Brocade(config-vlan-10-webauth)#dns-filter 1 191.166.2.44/24
```

Syntax: [no] **dns-filter** <number> <ip-address> <subnet-mask> | <wildcard>

For <number>, enter a number from 1 to 4 to identify the DNS filter.

Enter the IP address and subnet mask of unauthenticated hosts that will be forwarded to the unknown/untrusted servers. Use the <ip-address> <subnet-mask> or <ip-address>/<subnet-mask> format.

You can use a <wildcard> for the filter. The <wildcard> is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the IP address. Ones mean any value matches. For example, the <ip-address> and <subnet-mask> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C sub-net 209.157.22.x match the policy.

Forcing re-authentication when ports are down

If all ports on the device go down, you may want to force all authenticated hosts to be re-authenticated. You can do this by entering the **port-down-auth-mac-cleanup** command.

```
Brocade(config-vlan-10-webauth)#port-down-auth-mac-cleanup
```

Syntax: [no] **port-down-auth-mac-cleanup**

While this command is enabled, the device checks the link state of all ports that are members of the Web Authentication VLAN. If the state of all the ports is down, then the device forces all authenticated hosts to re-authenticate. However, hosts that were authenticated using the add mac command will remain authenticated; they are not affected by the **port-down-auth-mac-cleanup** command.

Forcing re-authentication after an inactive period

You can force Web Authenticated hosts to be re-authenticated if they have been inactive for a period of time. The inactive duration is calculated by adding the `mac-age-time` that has been configured for the device and the configured **authenticated-mac-age-time**. (The **mac-age-time** command defines how long a port address remains active in the address table.) If the authenticated host is inactive for the sum of these two values, the host is forced to be re-authenticated.

To force authenticated hosts to re-authenticate after a period of inactivity, enter commands such as the following.

```
Brocade(config)#mac-age-time 600
Brocade(config)#vlan 23
Brocade(config-vlan-23)webauth
Brocade(config-vlan-23-webauth)#reauth-time 303
Brocade(config-vlan-23-webauth)#authenticated-mac-age-time 300
```

Syntax: `[no] authenticated-mac-age-time <seconds>`

You can enter a value from 0 to the value entered for **reauth-time**. The default is 3600.

Refer to [“Changing the MAC age time and disabling MAC address learning”](#) on page 560 for details on the **mac-age-time** command. The default **mac-age-time** is 300 seconds and can be configured to be between 60 and 600 on the FastIron switch. If it is configured to be 0, then the MAC address does not age out due to inactivity.

Defining the web authorization redirect address

When a user enters a valid URL address (one that exists), the user is redirected to a Web Authentication address and the Welcome page for Web Authentication is displayed. By default, this Web Authentication address is the IP address of the FastIron switch. You can change this address so that the address matches the name on the security certificates.

To change the address on a Layer 2 switch, enter a command such as the following at the global configuration level.

```
Brocade(config)#webauth-redirect-address my.domain.net
```

To change the address on a Layer 3 switch, enter a command such as the following at the Web Authentication VLAN level.

```
Brocade(config-vlan-10-webauth)#webauth-redirect-address my.domain.net
```

Entering "my.domain.net" redirects the browser to `https://my.domain.net/` when the user enters a valid URL on the Web browser.

Syntax: `[no] webauth-redirect-address <string>`

For `<string>`, enter up to 64 alphanumeric characters. You can enter any value for `<string>`, but entering the name on the security certificate prevents the display of error messages saying that the security certificate does not match the name of the site.

Deleting a web authentication VLAN

To delete a Web Authentication VLAN, enter the following commands:


```
Brocade(config)#vlan 10  
Brocade(config-vlan-10)no webauth
```

Syntax: no webauth

Web authentication pages

There are several pages that can be displayed for Web Authentication.

When a user first enters a valid URL address on the Web browser, the browser is redirected to the Web Authentication URL (refer to [“Defining the web authorization redirect address”](#) on page 1906).

If Automatic Authentication is enabled, the following Welcome page appears:

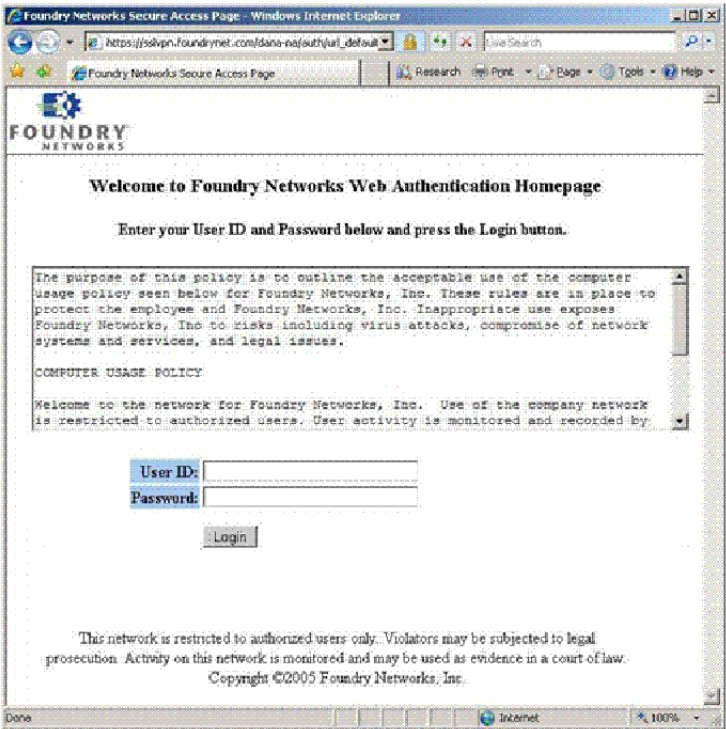
FIGURE 201 Example of a welcome page when automatic authentication is enabled



The browser will then be directed to the requested URL.

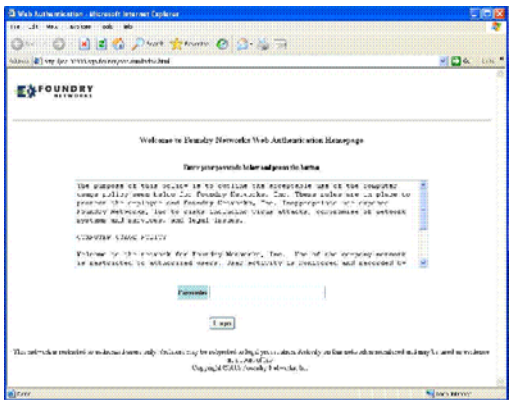
If username and password (Local User Database) authentication is enabled, the following Login page appears.

FIGURE 202 Example of a login page when automatic authentication is disabled and local user database is enabled



The user enters a user name and password, which are then sent for authentication.
If passcode authentication is enabled, the following Login page appears.

FIGURE 203 Example of a login page when automatic authentication is disabled and passcode Authentication is Enabled



The user enters a passcode, which is then sent for authentication.
If the Web Authentication fails, the page to try again is displayed (Figure 204).

FIGURE 204 Example of a try again page



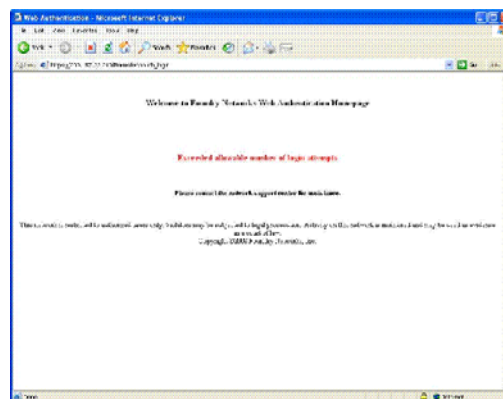
If the limit for the number of authenticated users on the network is exceeded, the Maximum Host Limit page is displayed (Figure 205).

FIGURE 205 Example of a maximum Host limit page



If the number of Web Authentication attempts by a user has been exceeded, the Maximum Attempts Limit page is displayed (Figure 206). The user is blocked from attempting any Web Authentication unless either the user MAC address is removed from the blocked list (using the **clear webauth block-mac <mac-address>** command) or when the block duration timer expires.

FIGURE 206 Example of a maximum attempts limit page



If the user Web Authentication attempt is successful, the Success page is displayed (Figure 207).

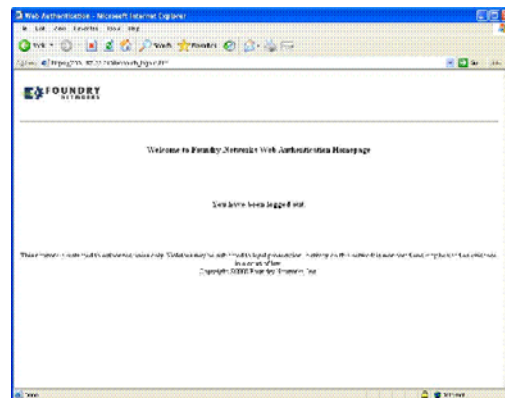
FIGURE 207 Example of a web authentication success page

Once a host is authenticated, that host can manually de-authenticate by clicking the **Logout** button in the Login Success page. The host remains logged in until the re-authentication period expires. At that time, the host is automatically logged out. However, if a re-authentication period is not configured, then the host remains logged in indefinitely.

NOTE

If you accidentally close the Success page, you will not be able to log out. If a re-authentication period is configured, you will be logged out once the re-authentication period ends.

The host can log out of the Web session by simply clicking the Logout button. Once logged out, the following window appears.



You can customize the top and bottom text for the all of the windows shown in [Figure 201](#) through [Figure 207](#).

Displaying text for web authentication pages

Use the **show webauth vlan <vlan-ID> webpage** command to determine what text has been configured for Web Authentication pages.

```
Brocade#show webauth vlan 25 webpage
=====
Web Page Customizations (VLAN 25):
  Top (Header): Default Text
    "<h3>Welcome to Brocade Communications, Inc. Web Authentication
Homepage</h3>"
  Bottom (Footer): Custom Text
    "Copyright 2009 SNL"
  Title: Default Text
    "Web Authentication"
  Login Button: Custom Text
    "Sign On"
  Web Page Logo: blogo.gif
    align: left (Default)
  Web Page Terms and Conditions: policy1.txt
```

Syntax: **show webauth vlan <vlan-id> webpage**

Customizing web authentication pages

You can customize the following objects in the Web Authentication pages shown in [Figure 201](#) through [Figure 207](#):

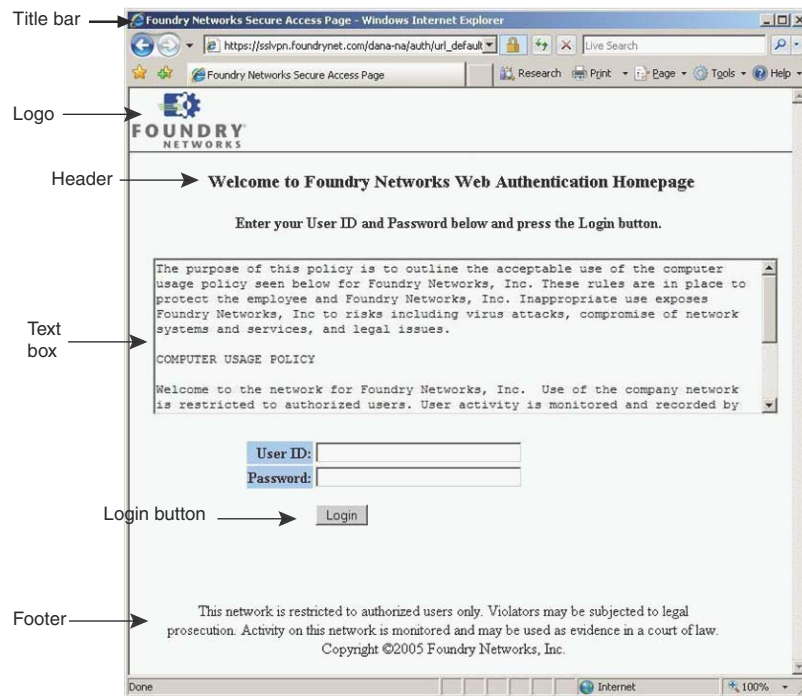
- Title bar
- Banner image (the logo)
- Header
- Text box
- Login button
- Footer

You can use the CLI commands **show webauth** and **show webauth vlan <vlan-id> webpage** to determine what text has been configured for Web Authentication pages.

NOTE

The banner image does not apply to the Web Authentication Maximum Attempts Limit page ([Figure 206](#)). The text box and Login button apply to the Login page only.

[Figure 208](#) shows the placement of these objects in the Login page.

FIGURE 208 Objects in the web authentication pages that can be customized

Customizing the title bar

You can customize the title bar that appears on all Web Authentication pages (refer to [Figure 208](#)). To do so, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#webpage custom-text title "Brocade Secure Access Page"
```

Syntax: [no] webpage custom-text title "<title>"

For <title>, enter up to 128 alphanumeric characters. The default title bar is "Web Authentication".

To reset the title bar back to the default value, enter the command **no webpage custom-text title**.

Customizing the banner image (Logo)

You can customize the logo that appears on all Web Authentication pages. [Figure 208](#) shows placement of the banner image in the Login page.

NOTE

The banner image does not display in the Maximum Attempts Limit page ([Figure 206](#)).

To customize the banner image, use the TFTP protocol to upload an image file from a TFTP server to the FastIron switch. The image file can be in the format jpg, bmp, or gif, and its size must be 64K or less. When you upload a new image file, it will overwrite the existing image file.

To replace the existing logo with a new one, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#webpage logo copy tftp 10.10.5.1 brocadelogo.gif
```

Syntax: [no] webpage logo copy tftp <ip-address> <filename>

NOTE

This command downloads the image file and stores it in the device flash memory. Therefore, it is not necessary to follow this command with a **write memory**.

The `<ip-address>` parameter specifies the address of the TFTP server on which the image file resides.

The `<filename>` parameter specifies the name of the image file on the TFTP server.

Use the `no webpage logo` command to delete the logo from all Web Authentication pages and remove it from flash memory.

Aligning the banner image (Logo)

You can optionally configure the placement of the logo that appears on all Web Authentication pages (refer to [Figure 208](#)). By default, the logo is left-aligned at the top of the page. To center the logo at the top of the page, enter the following command.

```
Brocade(config-vlan-10-webauth)#webpage logo align center
```

To right-justify the log at the top of the page, enter the following command.

```
Brocade(config-vlan-10-webauth)#webpage logo align right
```

Syntax: `[no] webpage logo align center | left | right`

Use the `no webpage logo align` command to reset the logo back to its default position (left).

Customizing the header

You can customize the header that appears on all Web Authentication pages. [Figure 208](#) shows placement of the header in the Login page.

To customize the header, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#webpage custom-text top "Welcome to Network One"
```

Syntax: `[no] webpage custom-text top <text>`

For `<text>`, enter up to 255 alphanumeric characters.

To reset the header back to the default text, enter the command `no webpage custom-text top`. The default text is "Welcome to Brocade Communications, Inc. Web Authentication Homepage".

Customizing the text box

You can customize the text box that appears on the Web Authentication Login page. [Figure 208](#) shows placement of the text box in the Login page. By default, the text box is empty and is not visible. To create a text box or to replace the existing one, upload an ASCII text file from a TFTP server to the FastIron switch. The text file size must not exceed 2K.

To create or replace a text box, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#webpage terms copy tftp 10.10.5.1 policy.txt
```

Syntax: `[no] webpage terms copy tftp <ip-address> <filename>`

NOTE

This command downloads the text file and stores it in the device flash memory. Therefore, it is not necessary to follow this command with a **write memory**.

The `<ip-address>` parameter is the address of the TFTP server on which the image resides.

The `<filename>` parameter is the name of the text file on the TFTP server.

To revert back to the default text box (none), enter the command **no webpage terms**.

Customizing the login button

You can customize the Login button that appears on the bottom of the Web Authentication Login page (refer to [Figure 208](#)). To do so, enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#webpage custom-text login-button "Press to Log In"
```

Syntax: `[no] webpage custom-text login-button "<text>"`

For `<text>`, enter up to 32 alphanumeric characters.

To reset the Login button back to the default value ("Login"), enter the command **no webpage custom-text login-button**.

Customizing the footer

You can customize the footer that appears on all Web Authentication pages. [Figure 208](#) shows placement of the footer in the Login page.

To customize the footer enter a command such as the following.

```
Brocade(config-vlan-10-webauth)#webpage custom-text bottom "Network One Copyright 2010"
```

Syntax: `[no] webpage custom-text bottom "<text>"`

For `<text>`, enter up to 255 alphanumeric characters.

To reset the footer back to the default text, enter the command **no webpage custom-text bottom**. The default text is "This network is restricted to authorized users only. Violators may be subjected to legal prosecution. Activity on this network is monitored and may be used as evidence in a court of law. Copyright 2009 Brocade Communications, Inc., Inc."

Displaying web authentication information

The following sections present the **show** commands you can use to display information about the Web Authentication feature.

Displaying the web authentication configuration

Enter the **show webauth** command to display the configuration for the Web Authentication feature.

```
Brocade#show webauth
```

```
=====
WEB AUTHENTICATION (VLAN 25): Enable
attempt-max-num: 5 (Default)
host-max-num: 0 (Default)
block duration: 90 (Default)
cycle-time: 600 (Default)
port-down-authenticated-mac-cleanup: Enable (Default)
reauth-time: 28800 (Default)
```

Displaying web authentication information

```
authenticated-mac-age-time: 3600 (Default)
dns-filter: Disable (Default)
authentication mode: username and password (Default)
  authentication methods: radius
    Local user database name: <none>
Radius accounting: Enable (Default)
Trusted port list: None
Secure Login (HTTPS): Enable (Default)
Web Page Customizations:
  Top (Header): Default Text
  Bottom (Footer): Custom Text
    "SNL Copyright 2009"
  Title: Default Text
  Login Button: Custom Text
    "Sign On"
  Web Page Logo: blogo.gif
    align: left (Default)
  Web Page Terms and Conditions: policy1.txt
Host statistics:
  Number of hosts dynamically authenticated: 0
  Number of hosts statically authenticated: 2
  Number of hosts dynamically blocked: 0
  Number of hosts statically blocked: 0
  Number of hosts authenticating: 1
```

The **show webauth** command displays the following information.

Field	Description
WEB AUTHENTICATION (VLAN 10)	Identifies the VLAN on which Web Authentication is enabled.
attempt-max-num	The maximum number of Web Authentication attempts during a cycle.
host-max-num	The maximum number of users that can be authenticated at one time.
block duration	How many seconds a user who failed Web Authentication must wait before attempting to be authenticated.
cycle-time	The number of seconds in one Web Authentication cycle.
port-down-authenticated-mac-cleanup	Indicates if this option is enabled or disabled. If enabled, all authenticated users are de-authenticated if all the ports in the VLAN go down.
reauth-time	The number of seconds an authenticated user remains authenticated. Once this timer expires, the user must re-authenticate.
authenticated-mac-age-time	If a user is inactive, this time shows how many seconds a user has before the user associated MAC address is aged out. The user will be forced to re-authenticate.
dns-filter	Shows the definition of any DNS filter that have been set. (Refer to "Filtering DNS queries" on page 1905)
authentication mode	The authentication mode: <ul style="list-style-type: none">• username and password (default)• passcode• none Also displays configuration details for the authentication mode.
RADIUS accounting	Whether RADIUS accounting is enabled or disabled.

Field	Description
Trusted port list	The statically-configured trusted ports of the Web Authentication VLAN.
Secure login (HTTPS)	Whether HTTPS is enabled or disabled.
Web Page Customizations	<p>The current configuration for the text that appears on the Web Authentication pages. Either "Custom Text" or "Default Text" displays for each page type:</p> <ul style="list-style-type: none"> "Custom Text" means the message for the page has been customized. The custom text is also displayed. "Default Text" means the default message that ships with the FastIron switch is used. <p>The actual text on the Web Authentication pages can be displayed using the show webauth vlan <vlan-id> webpage command. Refer to "Displaying text for web authentication pages" on page 1912.</p>
Host statistics	The authentication status and the number of hosts in each state.

Syntax: **show webauth** [vlan <vlan-id>]

The show webauth command by itself displays information for all VLANs on which Web Authentication is enabled. Use the vlan <vlan-id> parameter to display information for a specific VLAN.

Displaying a list of authenticated hosts

Enter the **show webauth allowed-list** command to display a list of hosts that are currently authenticated.

```

Brocade#show webauth allowed-list
=====
VLAN 1: Web Authentication
-----
Web Authenticated List      Configuration      Authenticated Duration Remaining
MAC Address      User Name      Static/Dynamic      HH:MM:SS
-----
00a0.f86c.2807      N/A            D                    00:03:05
0009.5b69.79ea      fdry1          D                    04:58:01
000c.db82.8bca      N/A            S                    Infinite
0007.e90e.de3b      N/A            S                    Infinite
000a.e442.a50e      fdry2          D                    00:25:25
    
```

The displays shows the following information.

Field	Description
VLAN #: Web Authentication	The ID of the VLAN on which Web Authentication is enabled.
Web Authenticated List MAC Address	The MAC addresses that have been authenticated.
User Name	The authenticated username.
Configuration Static/Dynamic	If the MAC address was dynamically (passed Web Authentication) or statically (added to the authenticated list using the add mac command) authenticated.
Authenticated Duration	The remainder of time the MAC address will remain authenticated

Syntax: show webauth allowed-list

Displaying a list of hosts attempting to authenticate

Enter the **show webauth authenticating-list** command to display a list of hosts that are trying to authenticate.

```
Brocade#show webauth authenticating-list
=====
VLAN 25: Web Authentication
-----
Web Authenticating List          # of Failed   Cycle Time
Remaining
MAC Address      User Name          Attempts      HH:MM:SS
-----
0012.3ff9.1fc6  N/A                      0             00:09:46
```

The report shows the following information.

This field...	Displays...
VLAN #: Web Authentication	The ID of the VLAN on which Web Authentication is enabled.
MAC Address	The MAC addresses that are trying to be authenticated.
User Name	The User Name associated with the MAC address.
# of Failed Attempts	Number of authentication attempts that have failed.
Cycle Time Remaining	The remaining time the user has to be authenticated before the current authentication cycle expires. Once it expires, the user must enter a valid URL again to display the Web Authentication Welcome page.

Syntax: show webauth authenticating-list

Displaying a list of blocked hosts

Enter the **show webauth blocked-list** command to display a list of hosts that are currently blocked from any Web Authentication Attempt.

```
Brocade#show webauth blocked-list
=====
VLAN 1: Web Authentication
-----
Web Block List          Configuration      Block Duration Remaining
MAC Address            User Name        Static/Dynamic      HH:MM:SS
-----
0009.a213.ff09         bauser           S                    00:31:27
00a0.f86c.2807         causer           D                    00:01:24
00a0.f890.1ab3         dauser           S                    infinite
```

The report shows the following information.

Field	Description
VLAN #: Web Authentication	The ID of the VLAN on which Web Authentication is enabled.
Web Block List MAC Address	The MAC addresses that have been blocked from Web Authentication.
User Name	The User Name associated with the MAC address.
Configuration Static/Dynamic	If the MAC address was dynamically or statically blocked. The block mac command statically blocks MAC addresses.
Block Duration Remaining	The remaining time the MAC address has before the user with that MAC address can attempt Web Authentication.

Syntax: show webauth blocked-list

Displaying a list of local user databases

The **show local-userdb** command displays a list of all local user databases configured on the FastIron switch and the number of users in each database.

```
Brocade#show local-userdb
=====
Local User Database Name      : My_Database
Number of users in the database : 4
=====
Local User Database Name      : test
Number of users in the database : 3
=====
Local User Database Name      : test123
Number of users in the database : 3
```

Syntax: show local-userdb

Displaying a list of users in a local user database

The **show local-userdb test** command displays a list of all users in a particular local user database.

```
Brocade#show local-userdb test

=====
Local User Database : test
Username                Password
-----                -
user1                   $e$&Z9'%'*%+
user2                   $e$, )A= )65N,%-3*%1?@U
user3                   $e$5%&-5%YO&&A1%6%<@U
```

As shown in the above example, passwords are encrypted in the command output.

Syntax: show local-userdb <db-name>

Displaying passcodes

If the *passcode* Web authentication mode is enabled, you can use the following command to display current passcodes.

```
Brocade#show webauth vlan 25 passcode
Current Passcode : 1389
This passcode is valid for 35089 seconds
```

Syntax: show webauth vlan <vlan-id> passcode

DoS Attack Protection

Table 316 lists individual Brocade switches and the DoS protection features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where noted.

TABLE 316 Supported DoS protection features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Smurf attack (ICMP attack) protection	Yes	Yes	Yes	Yes	Yes
TCP SYN attack protection	Yes	Yes	Yes	Yes	Yes

This chapter explains how to protect your Brocade devices from Denial of Service (DoS) attacks.

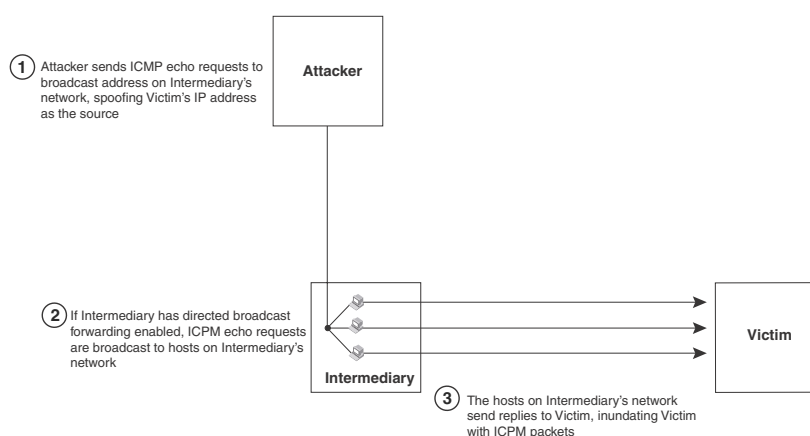
In a Denial of Service (DoS) attack, a router is flooded with useless packets, hindering normal operation. Brocade devices include measures for defending against two types of DoS attacks: Smurf attacks and TCP SYN attacks.

Smurf attacks

A *Smurf attack* is a kind of DoS attack in which an attacker causes a victim to be flooded with Internet Control Message Protocol (ICMP) echo (Ping) replies sent from another network.

Figure 209 illustrates how a Smurf attack works.

FIGURE 209 How a Smurf attack floods a victim with ICMP replies



The attacker sends an ICMP echo request packet to the broadcast address of an intermediary network. The ICMP echo request packet contains the spoofed address of a victim network as its source. When the ICMP echo request reaches the intermediary network, it is converted to a Layer 2 broadcast and sent to the hosts on the intermediary network. The hosts on the intermediary network then send ICMP replies to the victim network.

For each ICMP echo request packet sent by the attacker, a number of ICMP replies equal to the number of hosts on the intermediary network are sent to the victim. If the attacker generates a large volume of ICMP echo request packets, and the intermediary network contains a large number of hosts, the victim can be overwhelmed with ICMP replies.

Avoiding being an intermediary in a Smurf attack

A Smurf attack relies on the intermediary to broadcast ICMP echo request packets to hosts on a target subnet. When the ICMP echo request packet arrives at the target subnet, it is converted to a Layer 2 broadcast and sent to the connected hosts. This conversion takes place only when directed broadcast forwarding is enabled on the device.

To avoid being an intermediary in a Smurf attack, make sure forwarding of directed broadcasts is disabled on the Brocade device. Directed broadcast forwarding is disabled by default. To disable directed broadcast forwarding, do one of the following.

```
Brocade(config)#no ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

Avoiding being a victim in a Smurf attack

You can configure the Brocade device to drop ICMP packets when excessive numbers are encountered, as is the case when the device is the victim of a Smurf attack. You can set threshold values for ICMP packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

For example, to set threshold values for ICMP packets targeted at the router, enter the following command in global CONFIG mode.

```
Brocade(config)#ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

To set threshold values for ICMP packets received on interface 3/11, enter the following commands.

```
Brocade(config)#interface ethernet 3/11
Brocade(config-if-e1000-3/11)#ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

For Layer 3 router code, if the interface is part of a VLAN that has a router VE, you must configure ICMP attack protection at the VE level. Otherwise, you can configure this feature at the interface level as shown in the previous example. When ICMP attack protection is configured at the VE level, it will apply to routed traffic only. It will not affect switched traffic.

NOTE

You must configure VLAN information for the port *before* configuring ICMP attack protection. You cannot change the VLAN configuration for a port on which ICMP attack protection is enabled.

To set threshold values for ICMP packets received on VE 31, enter commands such as the following.

```
Brocade(config)#interface ve 31
Brocade(config-vif-31)#ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

Syntax: `ip icmp burst-normal <value> burst-max <value> lockup <seconds>`

The **burst-normal** <value> parameter can be from 1 through 100,000 packets per second.

The **burst-max** <value> parameter can be from 1 through 100,000 packets per second.

The **lockup** <value> parameter can be from 1 through 10,000 seconds.

This command is supported on Ethernet and Layer 3 interfaces.

The number of incoming ICMP packets per second is measured and compared to the threshold values as follows:

- If the number of ICMP packets exceeds the **burst-normal** value, the excess ICMP packets are dropped.
- If the number of ICMP packets exceeds the **burst-max** value, all ICMP packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

In the example, if the number of ICMP packets received per second exceeds 5,000, the excess packets are dropped. If the number of ICMP packets received per second exceeds 10,000, the device drops all ICMP packets for the next 300 seconds (5 minutes).

TCP SYN attacks

TCP SYN attacks exploit the process of how TCP connections are established to disrupt normal traffic flow. When a TCP connection starts, the connecting host first sends a TCP SYN packet to the destination host. The destination host responds with a SYN ACK packet, and the connecting host sends back an ACK packet. This process, known as a “TCP three-way handshake,” establishes the TCP connection.

While waiting for the connecting host to send an ACK packet, the destination host keeps track of the as-yet incomplete TCP connection in a connection queue. When the ACK packet is received, information about the connection is removed from the connection queue. Usually there is not much time between the destination host sending a SYN ACK packet and the source host sending an ACK packet, so the connection queue clears quickly.

In a TCP SYN attack, an attacker floods a host with TCP SYN packets that have random source IP addresses. For each of these TCP SYN packets, the destination host responds with a SYN ACK packet and adds information to the connection queue. However, because the source host does not exist, no ACK packet is sent back to the destination host, and an entry remains in the connection queue until it ages out (after approximately a minute). If the attacker sends enough TCP SYN packets, the connection queue can fill up, and service can be denied to legitimate TCP connections.

To protect against TCP SYN attacks, you can configure the Brocade device to drop TCP SYN packets when excessive numbers are encountered. You can set threshold values for TCP SYN packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

For example, to set threshold values for TCP SYN packets targeted at the router, enter the following command in global CONFIG mode.

```
Brocade(config)#ip tcp burst-normal 10 burst-max 100 lockup 300
```

To set threshold values for TCP SYN packets received on interface 3/11, enter the following commands.

```
Brocade(config)#interface ethernet 3/11
Brocade(config-if-e1000-3/11)#ip tcp burst-normal 10 burst-max 100 lockup 300
```

For Layer 3 router code, if the interface is part of a VLAN that has a router VE, you must configure TCP/SYN attack protection at the VE level. Otherwise, you can configure this feature at the interface level as shown in the previous example. When TCP/SYN attack protection is configured at the VE level, it will apply to routed traffic only. It will not affect switched traffic.

NOTE

You must configure VLAN information for the port *before* configuring TCP/SYN attack protection. You cannot change the VLAN configuration for a port on which TCP/SYN attack protection is enabled.

To set threshold values for TCP/SYN packets received on VE 31, enter commands such as the following.

```
Brocade(config)#interface ve 31
Brocade(config-vif-31)#ip tcp burst-normal 5000 burst-max 10000 lockup 300
```

Syntax: `ip tcp burst-normal <value> burst-max <value> lockup <seconds>`

NOTE

This command is available at the global CONFIG level on both Chassis devices and Compact devices. On Chassis devices, this command is available at the Interface level as well. This command is supported on Ethernet and Layer 3 interfaces.

The **burst-normal** <value> parameter can be from 1 – 100,000 packets per second.

The **burst-max** <value> parameter can be from 1 – 100,000 packets per second.

The **lockup** <value> parameter can be from 1 – 10,000 seconds.

The number of incoming TCP SYN packets per second is measured and compared to the threshold values as follows:

- If the number of TCP SYN packets exceeds the **burst-normal** value, the excess TCP SYN packets are dropped.
- If the number of TCP SYN packets exceeds the **burst-max** value, all TCP SYN packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

In the example, if the number of TCP SYN packets received per second exceeds 10, the excess packets are dropped. If the number of TCP SYN packets received per second exceeds 100, the device drops all TCP SYN packets for the next 300 seconds (5 minutes).

TCP security enhancement

TCP security enhancement improves upon the handling of TCP inbound segments. This enhancement eliminates or minimizes the possibility of a TCP reset attack, in which a perpetrator attempts to prematurely terminate an active TCP session, and a data injection attack, wherein an attacker injects or manipulates data in a TCP connection.

In both cases, the attack is blind, meaning the perpetrator does not have visibility into the content of the data stream between two devices, but blindly injects traffic. Also, the attacker does not see the direct effect, the continuing communications between the devices and the impact of the injected packet, but may see the indirect impact of a terminated or corrupted session.

The TCP security enhancement prevents and protects against the following three types of attacks:

- Blind TCP reset attack using the reset (RST) bit
- Blind TCP reset attack using the synchronization (SYN) bit
- Blind TCP packet injection attack

The TCP security enhancement is automatically enabled.

Protecting against a blind TCP reset attack using the RST bit

In a blind TCP reset attack using the RST bit, a perpetrator attempts to guess the RST bits to prematurely terminate an active TCP session.

To prevent a user from using the RST bit to reset a TCP connection, the RST bit is subject to the following rules when receiving TCP segments:

- If the RST bit is set and the sequence number is outside the expected window, the Brocade device silently drops the segment.
- If the RST bit is exactly the next expected sequence number, the Brocade device resets the connection.
- If the RST bit is set and the sequence number does not exactly match the next expected sequence value, but is within the acceptable window, the Brocade device sends an acknowledgement.

Protecting against a blind TCP reset attack using the SYN bit

In a blind TCP reset attack using the SYN bit, a perpetrator attempts to guess the SYN bits to prematurely terminate an active TCP session.

To prevent a user from using the SYN bit to tear down a TCP connection, in current software releases, the SYN bit is subject to the following rules when receiving TCP segments:

- If the SYN bit is set and the sequence number is outside the expected window, the Brocade device sends an acknowledgement (ACK) back to the peer.
- If the SYN bit is set and the sequence number is an exact match to the next expected sequence, the Brocade device sends an ACK segment to the peer. Before sending the ACK segment, the software subtracts one from the value being acknowledged.
- If the SYN bit is set and the sequence number is acceptable, the Brocade device sends an acknowledgement (ACK) segment to the peer.

Protecting against a blind injection attack

In a blind TCP injection attack, a perpetrator tries to inject or manipulate data in a TCP connection.

To reduce the chances of a blind injection attack, an additional check on all incoming TCP segments is performed.

Displaying statistics about packets dropped because of DoS attacks

To display information about ICMP and TCP SYN packets dropped because burst thresholds were exceeded, enter the **show statistics dos-attack** command.

```
Brocade#show statistics dos-attack
----- Local Attack Statistics -----
ICMP Drop Count      ICMP Block Count      SYN Drop Count      SYN Block Count
-----
0                    0                    0                    0
-----
----- Transit Attack Statistics -----
Port    ICMP Drop Count      ICMP Block Count      SYN Drop Count      SYN Block Count
-----
3/11    0                    0                    0                    0
```

Syntax: **show statistics dos-attack**

To clear statistics about ICMP and TCP SYN packets dropped because burst thresholds were exceeded, enter the **clear statistics dos-attack** command.

```
Brocade#clear statistics dos-attack
```

Syntax: **clear statistics dos-attack**

DHCP

[Table 317](#) lists individual Brocade switches and the Dynamic Host Configuration Protocol (DHCP) packet inspection and tracking features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 317 Supported DHCP packet inspection and tracking features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Dynamic ARP inspection	Yes	Yes	Yes	Yes	Yes
DHCP snooping	Yes	Yes	Yes	Yes	Yes
DHCP relay agent information (DHCP Option 82)	Yes	Yes	Yes	Yes	Yes
IP source guard	Yes	Yes	Yes	Yes	Yes

Dynamic ARP inspection

For enhanced network security, you can configure the Brocade device to inspect and keep track of Dynamic Host Configuration Protocol (DHCP) assignments.

Dynamic ARP Inspection (DAI) enables the Brocade device to intercept and examine all ARP request and response packets in a subnet and discard those packets with invalid IP to MAC address bindings. DAI can prevent common man-in-the-middle (MiM) attacks such as ARP cache poisoning, and disallow mis-configuration of client IP addresses.

ARP poisoning

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. Before a host can talk to another host, it must map the IP address to a MAC address first. If the host does not have the mapping in its ARP table, it creates an ARP request to resolve the mapping. All computers on the subnet will receive and process the ARP requests, and the host whose IP address matches the IP address in the request will send an ARP reply.

An ARP poisoning attack can target hosts, switches, and routers connected to the Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. For instance, a malicious host can reply to an ARP request with its own MAC address, thereby causing other hosts on the same subnet to store this information in their ARP tables or replace the existing ARP entry. Furthermore, a host can send gratuitous replies without having received any ARP requests. A malicious host can also send out ARP packets claiming to have an IP address that actually belongs to another host (e.g. the default router). After the attack, all traffic from the device under attack flows through the attacker computer and then to the router, switch, or host.

About Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) allows only valid ARP requests and responses to be forwarded.

A Brocade device on which DAI is configured does the following:

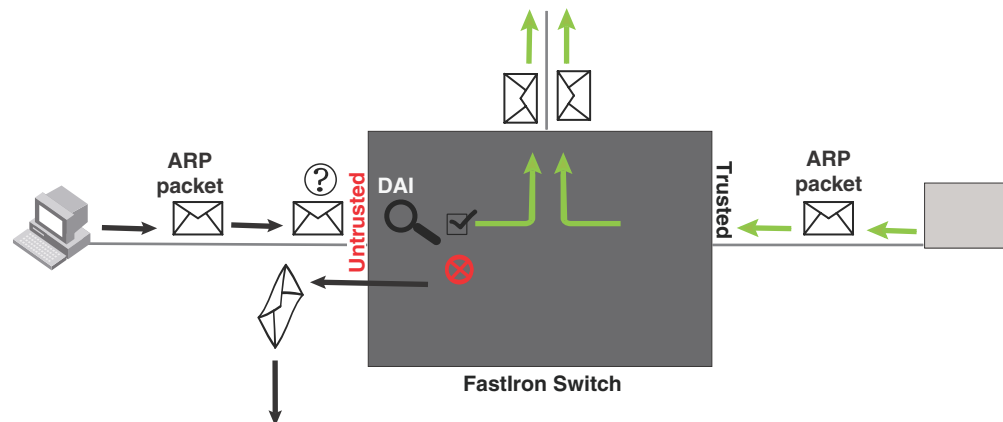
- Intercepts ARP packets received by the system CPU
- Inspects all ARP requests and responses received on untrusted ports
- Verifies that each of the intercepted packets has a valid IP-to-MAC address binding before updating the local ARP table, or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

When you enable DAI on a VLAN, by default, all member ports are untrusted. You must manually configure trusted ports. In a typical network configuration, ports connected to host ports are untrusted. You configure ports connected to other switches or routers as trusted.

DAI inspects ARP packets received on untrusted ports, as shown in [Figure 210](#). DAI carries out the inspection based on IP-to-MAC address bindings stored in a trusted binding database. For the Brocade device, the binding database is the ARP table, which supports DAI, DHCP snooping, and IP Source Guard. To inspect an ARP request packet, DAI checks the source IP and source MAC address against the ARP table. For an ARP reply packet, DAI checks the source IP, source MAC, destination IP, and destination MAC addresses. DAI forwards the valid packets and discards those with invalid IP-to-MAC address bindings.

When ARP packets reach a trusted port, DAI lets them through, as shown in [Figure 210](#).

FIGURE 210 Dynamic ARP inspection at work



ARP entries

DAI uses the IP/MAC mappings in the ARP table to validate ARP packets received on untrusted ports.

ARP entries in the ARP table derive from the following:

- **Dynamic ARP** – normal ARP learned from trusted ports.
- **Static ARP** – statically configured IP/MAC/port mapping.
- **Inspection ARP** – statically configured IP/MAC mapping, where the port is initially unspecified. The actual physical port mapping will be resolved and updated from validated ARP packets. Refer to [“Configuring an inspection ARP entry”](#) on page 1930.

- **DHCP-Snooping ARP** – information collected from snooping DHCP packets when DHCP snooping is enabled on VLANs.

The status of an ARP entry is either pending or valid:

- **Valid** – the mapping is valid, and the port is resolved. This is always the case for static ARP entries.
- **Pending** – for normal dynamic and inspection ARP entries before they are resolved, and the port mapped. Their status changes to valid when they are resolved, and the port mapped.

Refer to also [“System reboot and the binding database”](#) on page 1933.

Configuration notes and feature limitations for DAI

The following limits and restrictions apply when configuring DAI:

- To run Dynamic ARP Inspection, you must first enable support for ACL filtering based on VLAN membership or VE port membership. To do so, enter the following commands at the Global CONFIG Level of the CLI.

```
Brocade(config)#enable ACL-per-port-per-vlan
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
```

NOTE

You must save the configuration and reload the software to place the change into effect.

- Brocade recommends that you do not enable DAI on a trunk port.
- The maximum number of DHCP and static DAI entries depends on the maximum number of ARP table entries allowed on the device. A FastIron Layer 2 switch can have up to 4096 ARP entries and a FastIron Layer 3 switch can have up to 64,000 ARP entries. In a FastIron Layer 3 switch, you can use the **system-max ip-arp** command to change the maximum number of ARP entries for the device.

However, only up to 1024 DHCP entries can be saved to flash.

- ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled.
- On FastIron X Series devices, DAI is supported together with multi-device port authentication and dynamic ACLs. For details, refer to [“Support for dynamic ARP inspection with dynamic ACLs”](#) on page 1852.
- DAI is supported on a VLAN without a VE, or on a VE with or without an assigned IP address.

Dynamic ARP inspection configuration

Configuring DAI consists of the following steps.

1. Configure inspection ARP entries for hosts on untrusted ports. Refer to [“Configuring an inspection ARP entry”](#) on page 1930.
2. Enable DAI on a VLAN to inspect ARP packets. Refer to [“Enabling DAI on a VLAN”](#) on page 1930.
3. Configure the trust settings of the VLAN members. ARP packets received on *trusted* ports bypass the DAI validation process. ARP packets received on *untrusted* ports go through the DAI validation process. Refer to [“Enabling trust on a port”](#) on page 1931.
4. Enable DHCP snooping to populate the DHCP snooping IP-to-MAC binding database.

The following shows the default settings of DAI.

Feature	Default
Dynamic ARP Inspection	Disabled
Trust setting for ports	Untrusted

Configuring an inspection ARP entry

Static ARP and static inspection ARP entries need to be configured for hosts on untrusted ports. Otherwise, when DAI checks ARP packets from these hosts against entries in the ARP table, it will not find any entries for them, and the Brocade device will not allow and learn ARP from an untrusted host.

When the inspection ARP entry is resolved with the correct IP/MAC mapping, its status changes from pending to valid.

To configure an inspection ARP entry, enter a command such as the following.

```
Brocade(config)#arp 20.20.20.12 0001.0002.0003 inspection
```

This command defines an inspection ARP entry, mapping a device IP address 20.20.20.12 with its MAC address 0001.0002.0003. The ARP entry will be in **Pend** (pending) status until traffic with the matching IP-to-MAC is received on a port.

Syntax: [no] arp <ip-addr> <mac-addr> inspection

The <ip-addr> <mac-addr> parameter specifies a device IP address and MAC address pairing.

Enabling DAI on a VLAN

DAI is disabled by default. To enable DAI on an existing VLAN, enter the following command.

```
Brocade(config)#ip arp inspection vlan 2
```

The command enables DAI on VLAN 2. ARP packets from untrusted ports in VLAN 2 will undergo DAI inspection.

Syntax: [no] ip arp inspection vlan <vlan-number>

The <vlan-number> variable specifies the ID of a configured VLAN.

Enabling trust on a port

The default trust setting for a port is untrusted. For ports that are connected to host ports, leave their trust settings as untrusted.

To enable trust on a port, enter commands such as the following .

```
Brocade(config)#interface ethernet 1/4
Brocade(config-if-e10000-1/4)#arp inspection trust
```

The commands change the CLI to the interface configuration level of port 1/4 and set the trust setting of port 1/4 to trusted.

Syntax: [no] arp inspection trust

Displaying ARP inspection status and ports

To display the ARP inspection status for a VLAN and the trusted or untrusted port, enter the following command.

```
Brocade#show ip arp inspection vlan 2
IP ARP inspection VLAN 2: Disabled
  Trusted Ports :   ethe 1/4
  Untrusted Ports : ethe 2/1 to 2/3 ethe 4/1 to 4/24 ethe 6/1 to 6/4 ethe 8/1 to
                    8/4
```

Syntax: show ip arp inspection [vlan <vlan_id>]

The <vlan_id> variable specifies the ID of a configured VLAN.

Displaying the ARP table

To display the ARP table, enter the **show arp** command .

```
Brocade#show arp
Total number of ARP entries: 2, maximum capacity: 6000
No   IP Address      MAC Address      Type      Age      Port      Status
1    10.43.1.1         0004.80a0.4000   Dynamic   0        mgmt1     Valid
2    10.43.1.78        00e0.8160.6ab1   Dynamic   2        mgmt1     Valid
```

The command displays all ARP entries in the system. For field definitions, refer to [Table 184](#) on page 1071.

Syntax: show arp

DHCP snooping

Dynamic Host Configuration Protocol (DHCP) snooping enables the Brocade device to filter untrusted DHCP packets in a subnet. DHCP snooping can ward off MiM attacks, such as a malicious user posing as a DHCP server sending false DHCP server reply packets with the intention of misdirecting other users. DHCP snooping can also stop unauthorized DHCP servers and prevent errors due to user mis-configuration of DHCP servers.

Often DHCP snooping is used together with Dynamic ARP Inspection and IP Source Guard.

How DHCP snooping works

When enabled on a VLAN, DHCP snooping stands between untrusted ports (those connected to host ports) and trusted ports (those connected to DHCP servers). A VLAN with DHCP snooping enabled forwards DHCP request packets from clients and discards DHCP server reply packets on untrusted ports, and it forwards DHCP server reply packets on trusted ports to DHCP clients, as shown in the following figures

FIGURE 211 DHCP snooping at work - on an untrusted port

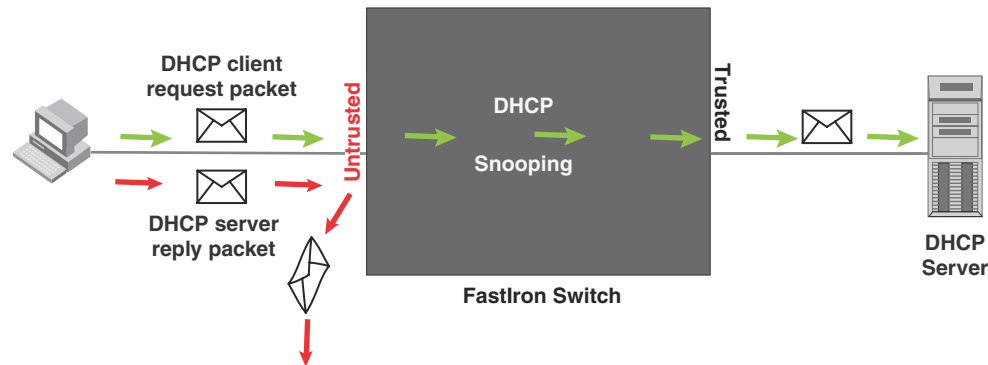
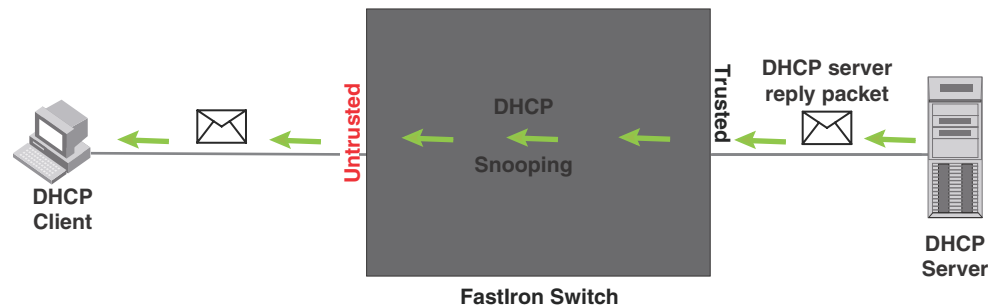


FIGURE 212 DHCP snooping at work - on a trusted port



DHCP binding database

When it forwards DHCP server reply packets on trusted ports, the Brocade device saves the client IP-to-MAC address binding information in the DHCP binding database. This is how the DHCP snooping binding table is populated. The information saved includes MAC address, IP address, lease time, VLAN number, and port number.

In the Brocade device, the DHCP binding database is integrated with the enhanced ARP table, which is used by Dynamic ARP Inspection. For more information, refer to [“ARP entries”](#) on page 1928.

The lease time will be refreshed when the client renews its IP address with the DHCP server; otherwise the Brocade device removes the entry when the lease time expires.

About client IP-to-MAC address mappings

Client IP addresses need not be on directly-connected networks, as long as the client MAC address is learned on the client port and the client port is in the same VLAN as the DHCP server port. In this case, the system will learn the client IP-to-MAC port mapping. Therefore, a VLAN with DHCP snooping enabled does not require a VE interface.

In earlier releases, in the Layer 3 software image, DHCP snooping does not learn the secure IP-to-MAC address mapping for a client, if the client port is not a virtual ethernet (VE) interface with an IP subnet address. In other words, the client IP address had to match one of the subnets of the client port in order for DHCP to learn the address mapping.

System reboot and the binding database

To allow DAI and DHCP snooping to work smoothly across a system reboot, the binding database is saved to a file in the system flash memory after an update to the binding database, with a 30 second delay. The flash file is written and read only if DHCP snooping is enabled.

Configuration notes and feature limitations for DHCP snooping

The following limits and restrictions apply to DHCP snooping:

- To run DHCP snooping, you must first enable support for ACL filtering based on VLAN membership or VE port membership. To do so, enter the following commands at the Global CONFIG Level of the CLI.

```
Brocade(config)#enable ACL-per-port-per-vlan
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
```

NOTE

You must save the configuration and reload the software to place the change into effect.

- DHCP snooping is supported on trunk ports (tagged and untagged) for trusted ports.
- DHCP snooping is not supported on trunk ports for untrusted ports.
- DHCP snooping is not supported together with DHCP Auto-configuration.
- A switch can have up to 256 ARP entries, therefore, DHCP entries are limited to 256. A router, however, can have 64,000 ARP entries, so a router can have up to 64,000 DHCP entries, of which only 1024 entries can be saved to flash on reboot.
- ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled.
- See also [“About client IP-to-MAC address mappings”](#) on page 1933.
- On FastIron X Series devices, DHCP snooping is supported together with multi-device port authentication and dynamic ACLs. For details, refer to [“Support for DHCP snooping with dynamic ACLs”](#) on page 1852.
- DHCP snooping supports DHCP relay agent information (DHCP Option 82). For details, refer to [“DHCP relay agent information”](#) on page 1936.

Configuring DHCP snooping

Configuring DHCP snooping consists of the following steps.

1. Enable DHCP snooping on a VLAN. Refer to [“Enabling DHCP snooping on a VLAN”](#) on page 1934.
2. For ports that are connected to a DHCP server, change their trust setting to trusted. Refer to [“Enabling trust on a port”](#) on page 1934.

The following shows the default settings of DHCP snooping.

Feature	Default
DHCP snooping	Disabled
Trust setting for ports	Untrusted

Enabling DHCP snooping on a VLAN

When DHCP snooping is enabled on a VLAN, DHCP packets are inspected.

DHCP snooping is disabled by default. This feature must be enabled on the client and the DHCP server VLANs. To enable DHCP snooping, enter the following global command for these VLANs.

```
Brocade(config)#ip dhcp snooping vlan 2
```

The command enables DHCP snooping on VLAN 2.

Syntax: [no] ip dhcp snooping vlan <vlan-number>

The <vlan-number> variable specifies the ID of a configured client or DHCP server VLAN.

Enabling trust on a port

The default trust setting for a port is untrusted. To enable trust on a port connected to a DHCP server, enter commands such as the following.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#dhcp snooping trust
```

Port 1/1 is connected to a DHCP server. The commands change the CLI to the interface configuration level of port 1/1 and set the trust setting of port 1/1 to trusted.

Syntax: [no] dhcp snooping trust

Disabling the learning of DHCP clients on a port

You can disable DHCP client learning on an individual port. To do so, enter commands such as the following.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#dhcp snooping client-learning disable
```

Syntax: [no] dhcp snooping client-learning disable

Use the **no** form of the command to re-enable DHCP client learning on a port once it has been disabled.

Clearing the DHCP binding database

You can clear the DHCP binding database using the CLI command **clear DHCP**. You can remove all entries in the database, or remove entries for a specific IP address only.

To remove all entries from the DHCP binding database, enter the **clear dhcp** command.

```
Brocade#clear dhcp
```

To clear entries for a specific IP address, enter a command such as the following.

```
Brocade#clear dhcp 10.10.102.4
```

Syntax: **clear dhcp** [*<ip-addr>*]

Displaying DHCP snooping status and ports

To display the DHCP snooping status for a VLAN and the trusted/untrusted port, use the **show ip dhcp snooping vlan** command.

```
Brocade#show ip dhcp snooping vlan 2
IP DHCP snooping VLAN 2: Enabled
```

Syntax: **show ip dhcp snooping** [vlan *<vlan-id>*]

Displaying the DHCP snooping binding database

To display the DHCP snooping binding database, use the **show ip dhcp snooping info** command.

```
Brocade#show ip dhcp snooping info
Dhcp snooping Info
Total learnt entries 1
SAVED DHCP ENTRIES IN FLASH
      IP Address      Mac Address      Port  vlan  lease
0      10.10.10.20      0001.0002.0003  6/13  1112  361
```

Syntax: **show ip dhcp snooping info**

Displaying DHCP binding entry and status

To display the DHCP binding entry and its current status, use the **show arp** command.

```
Brocade#show arp
Total number of ARP entries: 2, maximum capacity: 6000
No.   IP Address      MAC Address      Type    Age    Port    Status
1     10.43.1.1        00e0.0001.c320   Dynamic  0      mgmt1   Valid
2     10.43.1.199      00e0.0002.b263   Dynamic  7      mgmt1   Valid
```

Syntax: **show arp**

For field definitions, refer to [Table 184](#) on page 1071.

DHCP snooping configuration example

The following example configures VLAN 2 and VLAN 20, and changes the CLI to the global configuration level to enable DHCP snooping on the two VLANs. The commands are as follows.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#untagged ethe 1/3 to 1/4
Brocade(config-vlan-2)#router-interface ve 2
Brocade(config-vlan-2)#exit
Brocade(config)#ip dhcp snooping vlan 2

Brocade(config)#vlan 20
Brocade(config-vlan-20)#untagged ethe 1/1 to 1/2
Brocade(config-vlan-20)#router-interface ve 20
Brocade(config-vlan-20)#exit
Brocade(config)#ip dhcp snooping vlan 20
```

On VLAN 2, client ports 1/3 and 1/4 are untrusted by default all client ports are untrusted. Hence, only DHCP client request packets received on ports 1/3 and 1/4 are forwarded.

On VLAN 20, ports 1/1 and 1/2 are connected to a DHCP server. DHCP server ports are set to trusted .

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e10000-1/1)#dhcp snooping trust
Brocade(config-if-e10000-1/1)#exit
Brocade(config)#interface ethernet 1/2
Brocade(config-if-e10000-1/2)#dhcp snooping trust
Brocade(config-if-e10000-1/2)#exit
```

Hence, DHCP server reply packets received on ports 1/1 and 1/2 are forwarded, and client IP/MAC binding information is collected.

The example also sets the DHCP server address for the local relay agent.

```
Brocade(config)#interface ve 2
Brocade(config-vif-2)#ip address 20.20.20.1/24
Brocade(config-vif-2)#ip helper-address 1 30.30.30.4
Brocade(config-vif-2)#interface ve 20
Brocade(config-vif-20)#ip address 30.30.30.1/24
```

DHCP relay agent information

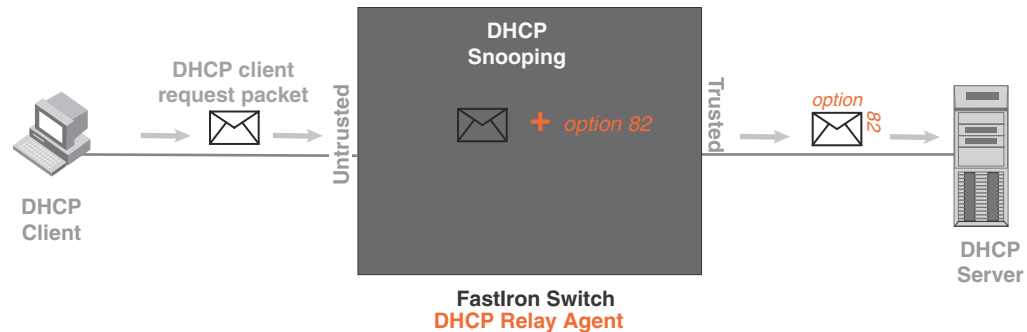
DHCP relay agent information, also known as DHCP option 82, enables a DHCP relay agent to insert information about a clients' identity into a DHCP client request being sent to a DHCP server.

When DHCP snooping is enabled on the FastIron switch, DHCP option 82 is automatically enabled. DHCP packets are processed as follows:

- Before relaying a DHCP discovery packet or DHCP request packet from a client to a DHCP server, the FastIron switch will add agent information to the packet.
- Before relaying a DHCP reply packet from a DHCP server to a client, the FastIron switch will remove relay agent information from the packet.

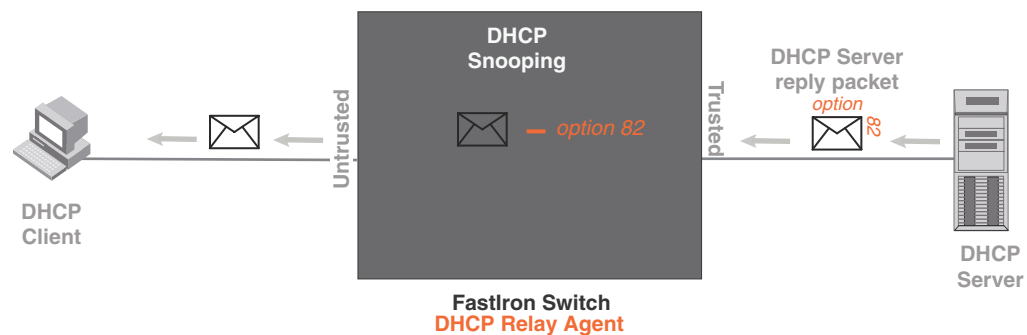
As illustrated in [Figure 213](#), the DHCP relay agent (the FastIron switch), inserts DHCP option 82 attributes when relaying a DHCP request packet to a DHCP server.

FIGURE 213 DHCP Option 82 attributes added to the DHCP packet



As illustrated in [Figure 214](#), the FastIron switch deletes DHCP option 82 attributes before forwarding a server reply packet back to a DHCP client.

FIGURE 214 DHCP Option 82 attributes removed from the DHCP packet



The DHCP option 82 insertion/deletion feature is available only when DHCP snooping is enabled for the client/server ports.

Configuration notes for DHCP option 82

- DHCP snooping and DHCP option 82 are supported on a per-VLAN basis.
- DHCP option 82 follows the same configuration rules and limitations as for DHCP snooping. For more information, refer to [“Configuration notes and feature limitations for DHCP snooping”](#) on page 1933.

DHCP Option 82 sub-options

The Brocade implementation of DHCP Option 82 supports the following sub-options:

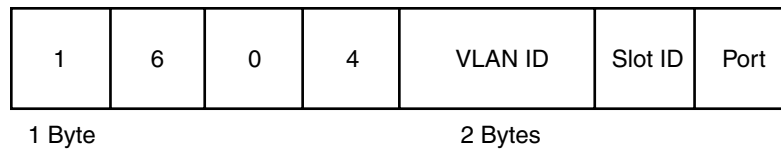
- Sub-Option 1 – Circuit ID
- Sub-Option 2 – Remote ID
- Sub-Option 6 – Subscriber ID

These sub-options are described in the following sections.

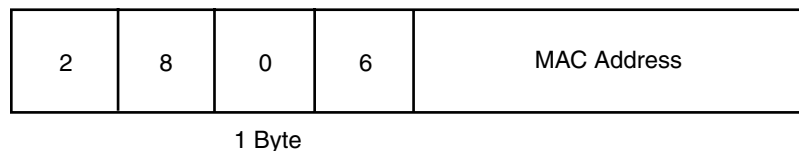
Sub-option 1 – circuit id

The Circuit ID (CID) identifies the circuit or port from which a DHCP client request was sent. The FastIron switch uses this information to relay DHCP responses back to the proper circuit, for example, the port number on which the DHCP client request packet was received.

Brocade FastIron devices support the **General CID packet format**. This simple format encodes the CID type, actual information length, VLAN ID, slot number, and port number. This format is compatible with the format used by other vendors' devices. [Figure 215](#) illustrates the general CID packet format.

FIGURE 215 General CID packet format***Sub-option 2 – Remote ID***

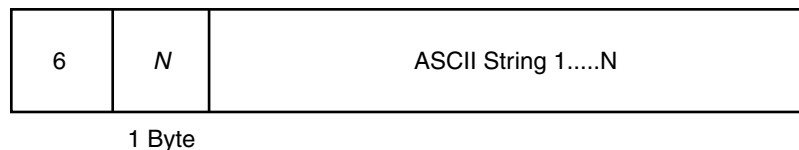
The Remote ID (RID) identifies the remote host end of the circuit (the relay agent). Brocade devices use the MAC address to identify itself as the relay agent. [Figure 216](#) illustrates the RID packet format.

FIGURE 216 RID packet format***Sub-option 6 - subscriber id***

The Subscriber ID (SID) is a unique identification number that enables an Internet Service Provider to:

- Identify a subscriber
- Assign specific attributes to that subscriber (for example, host IP address, subnet mask, and domain name server (DNS))
- Trigger accounting

[Figure 217](#) illustrates the SID packet format.

FIGURE 217 SID packet format

The second byte (*N* in [Figure 217](#)) is the length of the ASCII string that follows. The FastIron switch supports up to 50 ASCII characters.

DHCP option 82 configuration

When DHCP snooping is enabled on a VLAN, DHCP option 82 also is enabled by default. You do not need to perform any extra configuration steps to enable this feature. To enable DHCP snooping, refer to [“Enabling DHCP snooping on a VLAN”](#) on page 1934.

When processing DHCP packets, the FastIron switch applies the following default behavior when DHCP option 82 is enabled:

- Subjects all ports in the VLAN to DHCP option 82 processing
- Uses the general CID packet format
- Uses the standard RID packet format
- Replaces relay agent information received in DHCP packets with its own information
- Does not enable SID processing

When DHCP option 82 is enabled, you can optionally:

- Disable DHCP Option 82 processing on individual ports in the VLAN
- Configure the device to drop or keep the relay agent information in a DHCP packet instead of replacing it with its own information
- Enable SID processing

Disabling and re-enabling DHCP option 82 processing on an individual interface

By default, when DHCP option 82 is enabled on a VLAN, DHCP packets received on all member ports of the VLAN are subject to DHCP option 82 processing. You can optionally disable and later re-enable DHCP option 82 processing on one or more member ports of the VLAN. To do so, use the commands in this section.

To disable a particular port in a VLAN from adding relay agent information to DHCP packets, enter commands such as the following.

```
Brocade(config)#ip dhcp snooping vlan 1
Brocade(config)#interface ethernet 1/4
Brocade(config-if-e1000-1/4)#no dhcp snooping relay information
```

The first CLI command enables DHCP snooping and DHCP option 82 on VLAN 1. The second command changes the CLI configuration level to the Interface configuration level for port e 1/4. The last command disables DHCP option 82 on interface e 1/4, which is a member of VLAN 1.

To re-enable DHCP option 82 on an interface after it has been disabled, enter the following command at the Interface level of the CLI.

```
Brocade(config-if-e1000-1/4)#dhcp snooping relay information
```

Syntax: [no] dhcp snooping relay information

Use the show ip dhcp snooping vlan command to view the ports on which DHCP option 82 processing is disabled. For more information, refer to [“Viewing the ports on which DHCP option 82 is disabled”](#) on page 1941.

Changing the forwarding policy

When the Brocade device receives a DHCP message that contains relay agent information, by default, the device replaces the information with its own relay agent information. If desired, you can configure the device to keep the information instead of replacing it, or to drop (discard) messages that contain relay agent information. To do so, use the CLI commands in this section.

For example, to configure the device to *keep* the relay agent information contained in a DHCP message, enter the **ip dhcp relay information policy keep** command.

```
Brocade(config)#ip dhcp relay information policy keep
```

To configure the device to *drop* DHCP messages that contain relay agent information, enter the **ip dhcp relay information policy drop** command.

```
Brocade(config)#ip dhcp relay information policy drop
```

Syntax: **ip dhcp relay information policy** *<policy-type>*

<policy-type> can be one of the following:

- **drop** – Configures the device to discard messages containing relay agent information
- **keep** – Configures the device to keep the existing relay agent information
- **replace** – Configures the device to overwrite the relay agent information with the information in the Brocade configuration. This is the default behavior.

Use the **show ip dhcp relay information** command to view the forwarding policy configured on the switch. Refer to [“Viewing the circuit id, remote id, and forwarding policy”](#) on page 1941.

Enabling and disabling subscriber ID processing

You can configure a unique subscriber ID (SID) per port. Unlike the CID and RID sub-options, the SID sub-option is not automatically enabled when DHCP option 82 is enabled. To enable SID processing, enter commands such as the following.

```
Brocade(config)#ip dhcp snooping vlan 1
Brocade(config)#interface ethernet 1/4
Brocade(config-if-e1000-1/4)#dhcp snooping relay information subscriber-id Brcd01
```

The first CLI command enables DHCP snooping and DHCP option 82 on VLAN 1. The second command changes the CLI configuration level to the Interface configuration level for port e 1/4. The last command enables interface e 1/4 to insert the SID information in DHCP packets. In this case, the SID is **Brcd01**. All other ports in VLAN 1 on which SID is not enabled will send the standard relay agent information (CID and RID information) only.

Syntax: **[no] dhcp snooping relay information option subscriber-id** *<ASCII string>*

Enter up to 50 alphanumeric characters for *<ASCII string>*.

Use the **no** form of the command to disable SID processing once it is enabled.

Use the **show interfaces ethernet** command to view the subscriber ID configured on a port. Refer to [“Viewing the status of DHCP option 82 and the subscriber id”](#) on page 1942.

Viewing information about DHCP option 82 processing

Use the commands in this section to view information about DHCP option 82 processing.

Viewing the circuit id, remote id, and forwarding policy

Use the **show ip dhcp relay information** command to obtain information about the circuit ID, remote ID, and forwarding policy for DHCP option 82. The following shows an example output.

```
Brocade#show ip dhcp relay information
Relay Information: Format: Circuit-ID : vlan-mod-port
                    Remote-ID : mac
                    Policy : keep
```

Syntax: show ip dhcp relay information

TABLE 318 Output for the ip dhcp relay information command

Field	Description
Circuit-ID	The agent circuit ID format: <ul style="list-style-type: none"> vlan-mod-port – The default circuit ID format.
Remote-ID	The remote ID format. This field displays mac , which is the default remote ID format.
Policy	How the Brocade switch processes relay agent information it receives in DHCP messages: <ul style="list-style-type: none"> drop – drops the relay agent information keep – keeps the relay agent information replace – replaces the relay agent information with its own

Viewing the ports on which DHCP option 82 is disabled

Use the following command to refer which port in a DHCP snooping VLAN has DHCP Option 82 disabled.

```
Brocade#show ip dhcp snooping vlan 1
IP DHCP snooping VLAN 1: Enabled
  Trusted Ports :   ethe 3
  Untrusted Ports : ethe 1 to 2 ethe 4 to 24
  Relay Info. disabled Ports: ethe 10
```

Syntax: show ip dhcp snooping vlan <vlan-id>

TABLE 319 Output for the show ip dhcp snooping vlan command

Field	Description
IP DHCP snooping VLAN <vlan-id>	The DHCP snooping and DHCP option 82 status for a VLAN: <ul style="list-style-type: none"> Enabled Disabled
Trusted Ports	A list of trusted ports in the VLAN.
Untrusted Ports	A list of untrusted ports in the VLAN.
Relay Info. disabled Ports	Ports on which DHCP option 82 was disabled.

Viewing the status of DHCP option 82 and the subscriber id

Use the **show interfaces ethernet** command to obtain information about the status of DHCP option 82 and the configured subscriber ID, if applicable. In the example below, the text in **bold** type displays the information specific to DHCP option 82.

```
Brocade#show interfaces ethernet 3
GigabitEthernet3 is up, line protocol is up
  Hardware is GigabitEthernet, address is 00e0.5200.0002 (bia 00e0.5200.0002)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDI
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0
  Flow Control is config enabled, oper enabled, negotiation disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  IPG MII 96 bits-time, IPG GMII 96 bits-time
  IP MTU 1500 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 264 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runs, 0 giants
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions
    Relay Agent Information option: Enabled, Subscriber-ID: Brocade001
```

The above output shows that DHCP option 82 is **Enabled** on the device and the configured subscriber ID is **Brocade001**.

Syntax: **show interfaces ethernet** <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

IP source guard

You can use IP Source Guard together with Dynamic ARP Inspection on untrusted ports. Refer to [“DHCP snooping”](#) on page 1931 and [“Dynamic ARP inspection”](#) on page 1927.

The Brocade implementation of the IP Source Guard feature supports configuration on a port, on specific VLAN memberships on a port (Layer 2 devices only), and on specific ports on a virtual interface (VE) (Layer 3 devices only).

When IP Source Guard is first enabled, only DHCP packets are allowed and all other IP traffic is blocked. When the system learns a valid IP address, IP Source Guard then allows IP traffic. Only the traffic with valid source IP addresses are permitted. The system learns of a valid IP address from DHCP Snooping. When it learns a valid IP address, the system permits the learned source IP address.

When a new IP source entry binding on the port is created or deleted, the ACL will be recalculated and reapplied in hardware to reflect the change in IP source binding. By default, if IP Source Guard is enabled without any IP source binding on the port, an ACL that denies all IP traffic is loaded on the port.

Configuration notes and feature limitations for IP source guard

- To run IP Source Guard, you must first enable support for ACL filtering based on VLAN membership or VE port membership. To do so, enter the following commands at the Global CONFIG Level of the CLI.

```
Brocade(config)#enable ACL-per-port-per-vlan
Brocade(config)#write memory
Brocade(config)#exit
Brocade#reload
```

NOTE

You must save the configuration and reload the software to place the change into effect.

- Brocade FWS and FCX devices do not support IP Source Guard and dynamic ACLs on the same port.
- Brocade devices support IP Source Guard together with IPv4 ACLs (similar to ACLs for Dot1x), as long as both features are configured at the port-level or per-port-per-VLAN level. Brocade devices do not support IP Source Guard and IPv4 ACLs on the same port if one is configured at the port-level and the other is configured at the per-port-per-VLAN level.
- IP source guard and IPv6 ACLs are supported together on the same device, as long as they are not configured on the same port or virtual Interface.
- The following limitations apply when configuring IP Source Guard on Layer 3 devices:
 - You cannot enable IP Source Guard on a tagged port on a Layer 3 device. To enable IP Source Guard on a tagged port, enable it on a per-VE basis.
 - You cannot enable IP Source Guard on an untagged port with VE on a Layer 3 device. To enable IP Source Guard in this configuration, enable it on a per-VE basis.
 - There are no restrictions for Layer 2, either on the port or per-VLAN.
- You cannot enable IP Source Guard on a port that has any of the following features enabled:
 - MAC address filter
 - Rate limiting
 - Trunk port
 - 802.1x with ACLs
 - Multi-device port authentication with ACLs
- A port on which IP Source Guard is enabled limits the support of IP addresses, VLANs, and ACL rules per port. An IP Source Guard port supports a maximum of:

- 64 IP addresses
- 64 VLANs
- 64 rules per ACL
- The number of configured ACL rules affect the rate at which hardware resources are used when IP Source Guard is enabled. Use the **show access-list hw-usage on** command to enable hardware usage for an ACL, followed by a **show access-list <access-list-id>** command to determine the hardware usage for an ACL.

Example

```
Brocade#show access-list hw-usage on
Brocade#show access-list 100
Extended IP access list 100 (hw usage : 2)
deny ip any any (hw usage : 1)
```

To provide more hardware resource for IP Source Guard addresses, modify the ACL rules so that it uses less hardware resource.

- If you enable IP Source Guard in a network topology that has DHCP clients, you must also enable DHCP snooping. Otherwise, all IP traffic including DHCP packets will be blocked.
- When you enable IP Source Guard in a network topology that does not have DHCP clients, you must create an IP source binding for each client that will be allowed access to the network. Otherwise, data packets will be blocked. Refer to [“Defining static IP source bindings”](#) on page 1944.
- Source Guard Protection enables concurrent support with multi-device port authentication. For details, Refer to [“Enabling source guard protection”](#) on page 1864.
- IP Source Guard is supported on a VE with or without an assigned IP address.

Enabling IP source guard on a port

You can enable IP Source Guard on DHCP snooping untrusted ports. Refer to [“DHCP snooping”](#) on page 1931 for how to configure DHCP and DHCP untrusted ports.

By default, IP Source Guard is disabled. To enable IP Source Guard on a DHCP untrusted port, enter the following commands.

```
Brocade(config)#interface ethernet 1/4
Brocade(config-if-e10000-1/4)#source-guard enable
```

The commands change the CLI to the interface configuration level for port 1/4 and enable IP Source Guard on the port.

Syntax: [no] source-guard enable

Defining static IP source bindings

You can manually enter valid IP addresses in the binding database. To do so, enter a command such as the following.

```
Brocade(config)#ip source binding 10.10.10.1 e 2/4 vlan 4
```

Syntax: [no] ip source binding <ip-addr> ethernet [<slotnum> / <portnum>] [vlan <vlannum>]

For <ip-addr>, enter a valid IP address.

The `<slotnum>` parameter is required on chassis devices.

The `<portnum>` parameter is a valid port number.

The `[vlan <vlannum>]` parameter is optional. If you enter a VLAN number, the binding applies to that VLAN only. If you do not enter a VLAN number, the static binding applies to all VLANs associated with the port. Note that since static IP source bindings consume system resources, you should avoid unnecessary bindings.

Enabling IP source guard per-port-per-VLAN

To enable IP Source Guard per-port-per VLAN, enter commands such as the following.

```
Brocade(config)#vlan 12 name vlan12
Brocade(config-vlan-12)#untag ethernet 5 to 8
Brocade(config-vlan-12)#tag ethernet 23 to 24
Brocade(config-vlan-12)#exit
Brocade(config)#int e 23
Brocade(config-if-e1000-23)#per-vlan vlan12
Brocade(config-if-e1000-23-vlan-12)#source-guard enable
```

The commands in this example configure port-based VLAN 12, and add ports e 5 – 8 as untagged ports and ports e 23 – 24 as tagged ports to the VLAN. The last two commands enable IP Source Guard on port e 23, a member of VLAN 12.

Syntax: `[no] source-guard enable`

Enabling IP source guard on a VE

To enable IP Source Guard on a virtual interface, enter commands such as the following.

```
Brocade(config)#vlan 2
Brocade(config-vlan-2)#tag e1
Added tagged port(s) ethe 1 to port-vlan 2
Brocade(config-vlan-2)#router-int ve 2
Brocade(config-vlan-2)#int ve 2
Brocade(config-vif-2)#source-guard enable e 1
```

Syntax: `[no] source-guard enable`

Displaying learned IP addresses

To display the learned IP addresses for IP Source Guard ports, use the CLI commands **show ip source-guard ethernet**.

IP source guard

Rate Limiting and Rate Shaping on FastIron X Series and FCX and ICX Series Switches

[Table 320](#) lists the individual Brocade FastIron switches and the rate limiting and rate shaping features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 320 Supported rate limiting and rate shaping features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
Inbound rate limiting (Port-based rate limiting on inbound ports)	Yes	Yes	Yes	Yes	Yes
Outbound rate shaping	Yes	For details about inbound rate and outbound limiting on FastIron WS Series devices, see Chapter 50, "Rate Limiting on FastIron WS Series Switches,"	Yes	Yes	Yes
ACL-based rate limiting	Yes	Yes	Yes	Yes	Yes
	For details about ACL-based rate limiting, see Chapter 42, "Traffic Policies" .				

This chapter describes how to configure rate limiting and rate shaping on Brocade FastIron X Series, Brocade FCX Series, Brocade ICX 6610, Brocade ICX 6430, and Brocade ICX 6450 devices.

Rate limiting applies to inbound ports and rate shaping applies to outbound ports.

Rate limiting overview

Port-based fixed rate limiting is supported on inbound ports. This feature allows you to specify the maximum number of *bytes* a given port can receive. The port drops bytes that exceed the limit you specify. You can configure a Fixed rate limiting policy on a port inbound direction only. Fixed rate limiting applies to all traffic on the rate limited port.

Fixed rate limiting is at line rate and occurs in hardware. Refer to ["Rate limiting in hardware"](#) on page 1948.

When you specify the maximum number of bytes, you specify it in kilobits per second (kbps). The Fixed rate limiting policy applies to one-second intervals and allows the port to receive the number of bytes you specify in the policy, but drops additional bytes. Unused bandwidth is not carried over from one interval to the next.

NOTE

Brocade recommends that you do not use Fixed rate limiting on ports that receive route control traffic or Spanning Tree Protocol (STP) control traffic. If the port drops control packets due to the Fixed rate limiting policy, routing or STP can be disrupted.

Rate limiting in hardware

Each Brocade device supports line-rate rate limiting in hardware. The device creates entries in Content Addressable Memory (CAM) for the rate limiting policies. The CAM entries enable the device to perform the rate limiting in hardware instead of sending the traffic to the CPU. The device sends the first packet in a given traffic flow to the CPU, which creates a CAM entry for the traffic flow. A CAM entry consists of the source and destination addresses of the traffic. The device uses the CAM entry for rate limiting all the traffic within the same flow. A rate limiting CAM entry remains in the CAM for two minutes before aging out.

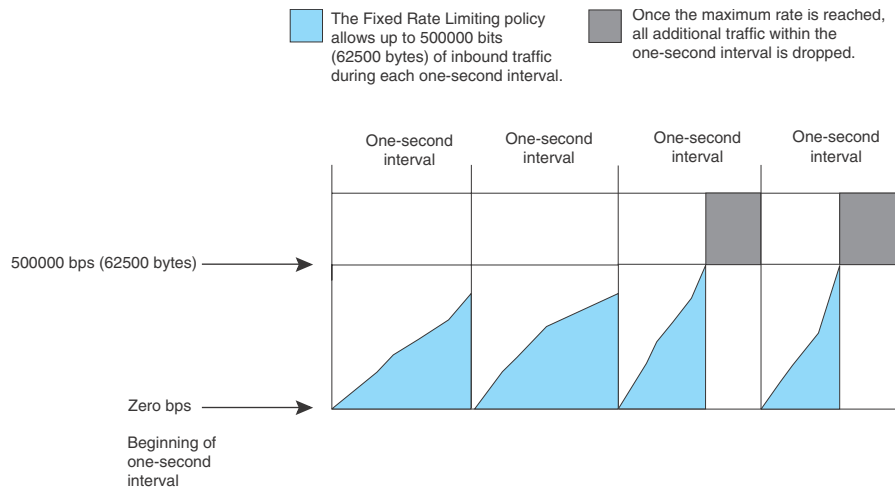
How Fixed rate limiting works

Fixed rate limiting counts the number of bytes that a port receives, in one second intervals. If the number exceeds the maximum number you specify when you configure the rate, the port drops all further inbound packets for the duration of the one-second interval.

Once the one-second interval is complete, the port clears the counter and re-enables traffic.

[Figure 218](#) shows an example of how Fixed rate limiting works. In this example, a Fixed rate limiting policy is applied to a port to limit the inbound traffic to 500000 bits (62500 bytes) a second. During the first two one-second intervals, the port receives less than 500000 bits in each interval. However, the port receives more than 500000 bits during the third and fourth one-second intervals, and consequently drops the excess traffic.

FIGURE 218 Fixed rate limiting

**NOTE**

The software counts the bytes by polling statistics counters for the port every 100 milliseconds, which provides 10 readings each second. Due to the polling interval, the Fixed Rate Limiting policy has an accuracy of within 10% of the port's line rate. It is therefore possible for the policy to sometimes allow more traffic than the limit you specify, but the extra traffic is never more than 10% of the port's line rate.

Configuration notes for rate limiting

- Rate limiting is available only on inbound ports.
- Port based Rate limiting is not supported on the SX-FI62XG and SX-FI42XG modules of the FastIron X Series devices.
- FastIron X Series devices do not support fixed rate limiting on tagged ports in the base Layer 3 and full Layer 3 images.
- The rate limit on IPv6 hardware takes several seconds to take effect at higher configured rate limit values. For example, if the configured rate limit is 750 Mbps, line-rate limiting could take up to 43 seconds to take effect.

Configuring a port-based rate limiting policy

To configure rate limiting on a port, enter commands such as the following.

```
Brocade(config)#interface ethernet 24
Brocade(config-if-e1000-24)#rate input fixed 500
```

These commands configure a fixed rate limiting policy that allows port 24 to receive a maximum of 500 kbits per second. If the port receives additional bytes during a given one-second interval, the port drops all inbound packets on the port until the next one-second interval starts.

Syntax: `[no] rate-limit input fixed <average-rate>`

For FastIron devices, the `<average-rate>` parameter specifies the maximum number of *kilobits* per second (kbps) the port can receive. The minimum rate that can be configured is 64 kbps.

Configuring an ACL-based rate limiting policy

IP ACL-based rate limiting of inbound traffic provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs. This feature is available in the Layer 2 and Layer 3 code.

To configure ACL-based rate limiting on a Brocade device, you create individual **traffic policies**, then reference the traffic policies in one or more ACL entries (also called clauses or statements). The traffic policies become effective on ports to which the ACLs are bound.

For configuration procedures for ACL-based rate limiting, refer to [Chapter 42, “Traffic Policies”](#).

Displaying the fixed rate limiting configuration

To display the fixed rate limiting configuration on the device, enter the **show rate-limit input** command.

```
Brocade# show rate-limit input
Total rate-limited interface count: 5.
  Port      Configured Input Rate      Actual Input Rate
  1/1              65000              65000
  1/2             195000             195000
  1/6              1950              1950
  5/2            230432            230000
  5/6            234113            234000
```

Syntax: show rate-limit input

The command lists the ports on which fixed rate limiting is configured, and provides the information listed in [Table 321](#) for each of the ports.

TABLE 321 CLI display of Fixed rate limiting information

Field	Description
Total rate-limited interface count	The total number of ports that are configured for Fixed rate limiting.
Port	The port number.
Configured Input Rate	The maximum rate requested for inbound traffic. The rate is measured in kilobits per second (kbps).
Actual Input Rate	The actual maximum rate provided by the hardware. The rate is measured in bps.

Rate shaping overview

Outbound Rate Shaping is a port-level feature that is used to shape the rate and control the bandwidth of outbound traffic on a port. This feature smooths out excess and bursty traffic to the configured maximum limit before it is sent out on a port. Packets are stored in available buffers and then forwarded at a rate no greater than the configured limit. This process provides for better control over the inbound traffic of neighboring devices.

The device has one global rate shaper for a port and one rate shaper for each port priority queue. Rate shaping is done on a single-token basis, where each token is defined to be 1 byte.

Configuration notes for rate shaping

The following rules apply when configuring outbound rate shapers:

- Outbound rate shapers can be configured *only* on physical ports, not on virtual or loopback ports.
- For trunk ports, the rate shaper must be configured on individual ports of a trunk using the **config-trunk-ind** command (trunk configuration level); you cannot configure a rate shaper for a trunk.
- This feature is supported on FastIron X Series and Brocade FCX Series devices only.
- When outbound rate shaping is enabled on a port on an IPv4 device, the port QoS queuing method (**qos mechanism**) will be strict mode. This applies to IPv4 devices only. On IPv6 devices, the QoS mechanism is whatever method is configured on the port, even when outbound rate shaping is enabled.
- You can configure a rate shaper for a port and for the individual priority queues of that port. However, if a port rate shaper is configured, that value overrides the rate shaper value of a priority queue if the priority queue rate shaper is greater than the rate shaper for the port.

The configured rate shaper values are rounded up to the nearest multiples of minimum values supported on the platform. [Table 322](#) shows the minimum and maximum values for output rate shaping on various devices. Values are in Kbps.

TABLE 322 Output rate shaping on FastIron devices

Device	Module	Minimum	Maximum
ICX 6610	1 Gbps ports	3	999750
ICX 6610	10 Gbps ports	3388	9996513
FCX	1 Gbps ports	89	999666
FCX	10 Gbps ports	3388	9996513
FSX	SX-FI48GPP module	51	999485
FSX	SX-FI-24GPP module	51	999485
FSX	SX-FI424P module	651	999936
FSX	SX-FI-2XG module	3	9999000
FSX	SX-FI-8XG module	3	9999000

Configuring outbound rate shaping for a port

To configure the maximum rate at which outbound traffic is sent out on a port, enter commands such as the following.

```
Brocade(config)#interface e 1/2
Brocade(config-if-e1000-2)#rate-limit output shaping 1300
```

The above commands affect FastIron X Series and Brocade FCX Series systems differently:

- On FastIron X Series devices, the configured 1300 Kbps outbound rate shaping on port 2 is rounded up to the nearest multiple of 651 Kbps, which is 1302 Kbps. This value is the actual limit on the port for outbound traffic.
- On Brocade FCX Series devices, the configured outbound rate shaper of 651 Kbps on port 1/15 is rounded to 616 Kbps. The configured 1300 Kbps limit on port 15 is rounded to 1232 Kbps..

Syntax: [no] **rate-limit output shaping** <value>

Configuring outbound rate shaping for a specific priority

To configure the maximum rate at which outbound traffic is sent out on a port priority queue, enter commands such as the following.

```
Brocade(config)#interface e 1/2
Brocade(config-if-e1000-2)#rate-limit output shaping 500 priority 7
```

The above commands affect FastIron X Series and Brocade FCX Series systems differently:

- On FastIron X Series devices, the configured 500 Kbps limit for outbound traffic on Priority queue 7 on port 2 is rounded up to the nearest multiple of 651 Kbps, which is 651 Kbps.
- On Brocade FCX Series devices, the configured 500 Kbps limit for outbound traffic on priority queue 7 on port 2 is rounded to a value that is programmable by the hardware, which is 440 Kbps.

Syntax: [no] **rate-limit output shaping** <value> **priority** <priority-queue>

Specify 0-7 for <priority-queue>

Configuring outbound rate shaping for a trunk port

This feature is supported on individual ports of a static trunk group and on LACP trunk ports. However, it is not supported on LACP trunk ports for FCX devices.

To configure the maximum rate at which outbound traffic is sent out on a trunk port, enter the following on each trunk port where outbound traffic will be shaped.

```
Brocade(config)#trunk e 1/13 to 1/16
Brocade(config-trunk-13-16)#config-trunk-ind
Brocade(config-trunk-13-16)#rate-limit output shaping ethe 1/15 651
Brocade(config-trunk-13-16)#rate-limit output shaping ethe 1/14 1300
```

The above commands configure an outbound rate shaper on port 1/14 and port 1/15. The commands affect FastIron X Series devices and Brocade FCX Series devices differently:

- On FastIron X Series devices, the configured outbound rate shaper (651 Kbps) on port 1/15 is the maximum rate of outbound traffic that is sent out on that port, since 651 Kbps is a multiple of 651 Kbps. The configured 1300 Kbps limit on port 14 is rounded up to 1302 Kbps.
- On Brocade FCX Series devices, the configured outbound rate shaper (651 Kbps) on port 1/15 is the rounded to 616 Kbps. The configured 1300 Kbps limit on port 14 is rounded to 1232 Kbps.

Syntax: [no] **rate-limit output shaping ethernet** <port> <value>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Displaying rate shaping configurations

To display the configured outbound rate shaper on a device, enter the **show rate-limit output-shaping** command.

```
Brocade#show rate-limit output-shaping
```

Outbound Rate Shaping Limits in Kbps:

Port	PortMax	Prio0	Prio1	Prio2	Prio3	Prio4	Prio5	Prio6	Prio7
1	-	-	-	-	-	-	-	-	651
2	1302	-	-	-	-	-	-	-	-
15	651	-	-	-	-	-	-	-	-

The display lists the ports on a device, the configured outbound rate shaper on a port and for a priority for a port.

Rate Limiting on FastIron WS Series Switches

[Table 323](#) lists the individual Brocade FastIron switches and the rate limiting features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 323Supported rate limiting features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	FCX	ICX 6610	ICX 6430 ICX 6450
Inbound rate limiting (Port-based rate limiting on inbound ports)	Yes For details, see Chapter 49 on page 1947 .	Yes	Yes For details, see Chapter 49 on page 1947 .	Yes For details, see Chapter 49 on page 1947 .	Yes For details, see Chapter 49 on page 1947	Yes For details, see Chapter 49 on page 1947
Outbound rate limiting (Port-based and port- and priority-based rate limiting on outbound ports)	No	Yes	No	No	No	No
ACL-based rate limiting	Yes	Yes	Yes	Yes	No	No

For details about ACL-based rate limiting, see [Chapter 42, "Traffic Policies"](#).

This chapter describes how to configure fixed rate limiting on inbound and outbound ports on FWS devices using the CLI.

Rate limiting overview

Fixed rate limiting allows you to specify the maximum number of bytes a given port can send or receive. The port drops bytes that exceed the limit you specify. You can configure a Fixed rate limiting policy on a port inbound or outbound direction. Fixed rate limiting applies to all traffic on the rate limited port.

When you specify the maximum number of bytes, you specify it in kilobits per second (Kbps). The Fixed rate limiting policy applies to one-second intervals and allows the port to send or receive the number of bytes you specify in the policy, but drops additional bytes. Unused bandwidth is not carried over from one interval to the next.

NOTE

Brocade recommends that you do not use Fixed rate limiting on ports that send or receive route control traffic or Spanning Tree Protocol (STP) control traffic. If the port drops control packets due to the Fixed rate limiting policy, routing or STP can be disrupted.

Rate limiting in hardware

Each device supports line-rate rate limiting in hardware. The device creates entries in Content Addressable Memory (CAM) for the rate limiting policies. The CAM entries enable the device to perform the rate limiting in hardware instead of sending the traffic to the CPU. The device sends the first packet in a given traffic flow to the CPU, which creates a CAM entry for the traffic flow. A CAM entry consists of the source and destination addresses of the traffic. The device uses the CAM entry for rate limiting all the traffic within the same flow. A rate limiting CAM entry remains in the CAM for two minutes before aging out.

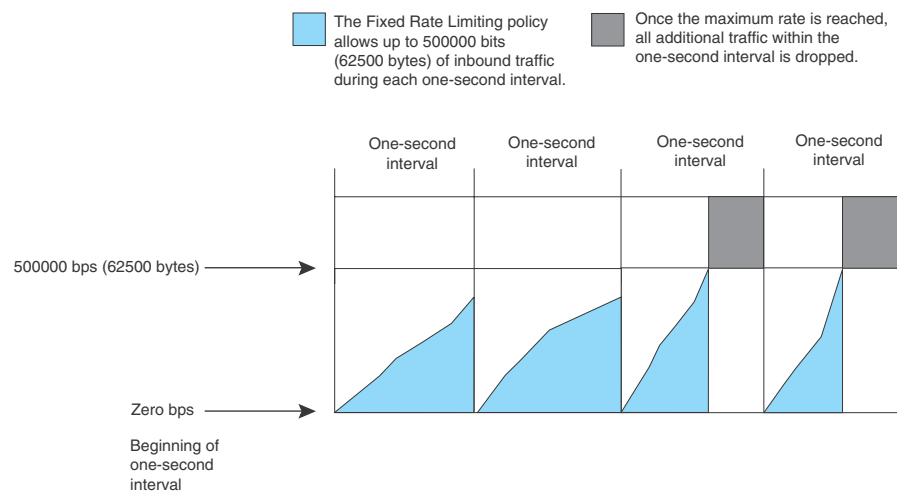
How fixed rate limiting works

Fixed rate limiting counts the number of bytes that a port either sends or receives, in one second intervals. The direction that the software monitors depends on the direction you specify when you configure the rate limit on the port. If the number of bytes exceeds the maximum number you specify when you configure the rate, the port drops all further packets for the rate-limited direction, for the duration of the one-second interval.

Once the one-second interval is complete, the port clears the counter and re-enables traffic.

Figure 219 shows an example of how Fixed rate limiting works. In this example, a Fixed rate limiting policy is applied to a port to limit the inbound traffic to 500000 bits (62500 bytes) a second. During the first two one-second intervals, the port receives less than 500000 bits in each interval. However, the port receives more than 500000 bits during the third and fourth one-second intervals, and consequently drops the excess traffic.

FIGURE 219 Fixed rate limiting



NOTE

The software counts the bytes by polling statistics counters for the port every 100 milliseconds, which provides 10 readings each second. Due to the polling interval, the Fixed Rate Limiting policy has an accuracy of within 10% of the port's line rate. It is therefore possible for the policy to sometimes allow more traffic than the limit you specify, but the extra traffic is never more than 10% of the port's line rate.

Fixed rate limiting on inbound port configuration

Inbound rate limiting allows you to specify the maximum number of Kbps a given port can receive.

Minimum and maximum inbound rate limits

[Table 324](#) lists the minimum and maximum inbound rate limits on GbE and 10-GbE ports

TABLE 324 Rates for inbound rate limiting

Port type	Minimum rate	Maximum rate
GbE	65 Kbps	1000000 Kbps
10-GbE	65 Kbps	10000000 Kbps

Configuration notes for fixed rate limiting

Inbound rate limiting is **supported** on:

- GbE ports
- 10-GbE ports
- Trunk ports

Inbound rate limiting is **not supported** on:

- Ports on which LACP is enabled
- Virtual interfaces
- Loopback interfaces

Fixed rate limiting is **supported** on:

- inbound ports only

Fixed rate limiting is **not supported** on:

- 10 Gbps Ethernet ports
- Tagged ports in the base Layer 3 and full Layer 3 images

Configuration syntax for fixed rate limiting

To configure inbound rate limiting on a port, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/2/1
Brocade(config-if-e10000-0/2/1)#rate-limit input fixed 1000000
Rate Limiting on Port 0/2/1 - Config: 1000000 Kbps, Actual: 1000000 Kbps
```

The above commands configure a fixed rate limiting policy that allows port 0/2/1, a 10-GbE port, to receive a maximum of 1000000 kilobits per second. If the port receives additional bits during a given one-second interval, the port drops all inbound packets on the port until the next one-second interval starts.

```
Brocade(config)#interface ethernet 0/1/10
Brocade(config-if-e1000-0/1/10)#rate-limit input fixed 1000
Rate Limiting on Port 0/1/10 - Config: 1000 Kbps, Actual: 1000 Kbps
```

The above commands configure a fixed rate limiting policy that allows port 0/1/10, a GbE port, to receive a maximum of 1000 kilobits per second. If the port receives additional bits during a given one-second interval, the port drops all inbound packets on the port until the next one-second interval starts.

Syntax: `[no] rate-limit input fixed <average-rate>`

The `<average-rate>` parameter specifies the maximum number of kilobits per second (Kbps) the port can receive. [Table 324](#) lists the minimum and maximum rates.

Fixed rate limiting on outbound port configuration

NOTE

This feature is not supported on FastIron X Series and Brocade FCX Series devices.

Outbound rate limiting allows you to specify the maximum number of kilobits a given port can transmit.

Brocade devices support the following types of outbound fixed rate limiting on Gbps and 10 Gbps Ethernet ports.

- **Port-based** – Limits the rate of outbound traffic on an individual physical port or trunk port, to a specified rate. Traffic that exceeds the maximum rate is dropped. Only one port-based outbound rate limiting policy can be applied to a port.
- **Port- and priority-based** – Limits the rate on an individual 802.1p priority queue on an individual physical port or trunk port. Traffic that exceeds the rate is dropped. Only one priority-based rate limiting policy can be specified per priority queue for a port. This means that a maximum of eight port- and priority-based policies can be configured on a port.

Minimum and maximum outbound rate limits

[Table 325](#) lists the minimum and maximum outbound rate limits on GbE and 10-GbE ports.

TABLE 325 Rates for outbound rate limiting

Port type	Minimum rate	Maximum rate	Granularity
GbE	65 Kbps	1000000 Kbps	65 Kbps
10-GbE	2500 Kbps	10000000 Kbps	2500 Kbps

Configuration notes for outbound rate limiting

- Outbound rate limiting is supported on:
 - GbE ports
 - 10-GbE ports
 - Trunk ports
- Outbound rate limiting is not supported on:
 - Ports on which LACP is enabled
 - Virtual interfaces
 - Loopback interfaces

- This feature is not supported on FastIron X Series and Brocade FCX Series devices.
- Because of the hardware architecture, the effect of outbound rate limiting differs on GbE ports compared to 10-GbE ports. For example, applying the same rate limiting value on GbE and 10-GbE ports will produce different results.
- You can configure both outbound port-base rate limiting and outbound port- and priority-based rate limiting on a single physical port or trunk port. However, if a priority-based limit for a given port is greater than the port-based rate limit, then the port-based rate limit will override the priority-based rate limit. Similarly, if the port-based rate limit is greater than the priority-based limit, then the priority-based rate limit will override the port-based rate limit.

Configuring port-based fixed rate limiting

To configure port-based fixed rate limiting on an outbound port, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/34
Brocade(config-if-e1000-0/1/34)#rate-limit output fixed 32
Outbound Rate Limiting on Port 0/1/34 Config: 32 Kbps, Actual: 65 Kbps
```

The above commands configure a fixed rate limiting policy that allows port 0/1/34 to transmit 32 Kbps. Since port 0/1/34 is a GbE port and the minimum rate is 65 Kbps (refer to [Table 325](#)), the system will adjust the configured rate of 32 Kbps to an actual rate to 65 Kbps. If the port transmits additional bits during a given one-second interval, the port will drop all outbound packets on the port until the next one-second interval starts.

```
Brocade(config)#interface ethernet 0/2/1
Brocade(config-if-e1000-0/2/1)#rate-limit output fixed 32
Outbound Rate Limiting on Port 0/2/1 Config: 32 Kbps, Actual: 2500 Kbps
```

The above commands configure a fixed rate limiting policy that allows port 0/2/1 to transmit 32 Kbps per second. Since port 0/2/1 is a 10-GbE port and the minimum rate is 2500 Kbps (refer to [Table 325](#)), the system will adjust the configured rate of 32 Kbps to an actual rate of 2500 Kbps. If the port transmits additional bits during a given one-second interval, the port will drop all outbound packets on the port until the next one-second interval starts.

Syntax: `[no] rate-limit output fixed <average-rate>`

The `<average-rate>` parameter specifies the average number of kilobits per second (Kbps) the port can send. [Table 325](#) lists the minimum and maximum rates for GbE and 10-GbE ports.

Configuring port- and priority-based rate limiting

Port- and priority-based rate limiting limits the rate on an individual 802.1p priority queue on an individual physical port or trunk port.

To configure port- and priority-based fixed rate limiting on an outbound port, enter commands such as the following.

```
Brocade(config)#interface ethernet 0/1/35
Brocade(config-if-e1000-0/1/35)#rate-limit output fixed 1000 priority 7
Outbound Rate Limiting on Port 0/1/35 for Priority 7
Config: 1000 Kbps, Actual: 975 Kbps
```

The above commands configure a fixed rate limiting policy that allows traffic with a priority of 7 on port 0/1/35 to transmit 1000 Kbps per second. The system rounds the configured rate to 975 Kbps. If the port transmits additional bits during a given one-second interval, the port will drop all outbound packets on the port until the next one-second interval starts.

ACL-based rate limiting policy configuration

Brocade devices support IP ACL-based rate limiting of inbound traffic. ACL-based rate limiting provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs.

To configure ACL-based rate limiting, you create individual **traffic policies**, then reference the traffic policies in one or more ACL entries (also called clauses or statements). The traffic policies become effective on ports to which the ACLs are bound.

For configuration procedures for ACL-based rate limiting, refer to [Chapter 42, “Traffic Policies”](#).

Displaying the fixed rate limiting configuration

You can display the fixed rate limiting configuration for inbound and outbound traffic.

Inbound ports

To display the fixed rate limiting configuration on inbound ports, enter the **show rate-limit input** command.

```
Brocade#show rate-limit input
Total rate-limited interface count: 11.
Port      Configured Input Rate      Actual Input Rate
1         1000000                    1000000
3         10000000                   10005000
7         10000000                   10000000
9         7500000                    7502000
11        8000000                    7999000
12        8000000                    7999000
13        8000000                    7999000
14        8000000                    7999000
15        8000000                    7999000
21        8000000                    8000000
25        7500000                    7502000
```

Syntax: show rate-limit input

The command lists the ports on which fixed rate limiting is configured, and provides the information listed in [Table 326](#) for each of the ports.

TABLE 326 CLI display of Fixed rate limiting information on inbound ports

Field	Description
Total rate-limited interface count	The total number of ports that are configured for Fixed rate limiting.
Port	The port number.
Configured Input Rate	The maximum rate requested for inbound traffic. The rate is measured in bits per second (bps).
Actual Input Rate	The actual maximum rate provided by the hardware. The rate is measured in bps.

Displaying fixed rate limiting configuration on outbound ports

To display the fixed rate limiting configuration on outbound ports, enter the **show rate-limit fixed output** command.

```
Brocade#show rate-limit fixed output
Outbound Rate Shaping Limits in Kbps:
Port      PortMax  Prio0  Prio1  Prio2  Prio3  Prio4  Prio5  Prio6  Prio7
  1        1000000
  3        1000000
  7        1000000
  9         750000
 11         800000
 12         800000
 13         800000
 14         800000
 15         800000
 21         800000
 25         750000
```

Syntax: show rate-limit fixed output

The command lists the ports on which fixed rate limiting is configured, and provides the information listed in [Table 326](#) for each of the ports.

TABLE 327 CLI display of Fixed rate limiting information on outbound ports

Field	Description
Port	The port number.
PortMax	The maximum rate requested for outbound traffic. The rate is measured in bits per second (bps).
Prio0 – Prio7	The port- and priority-based rate limit maximum provided by the hardware. The rate is measured in bps.

Displaying the fixed rate limiting configuration

Quality of Service

Table 328 lists the individual Brocade FastIron switches and the Quality of Service (QoS) features they support. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 328 Supported QoS features

Feature	FESX FSX 800 FSX 1600	FWS	FCX	ICX 6610	ICX 6430 ICX 6450
802.1p Quality of Service (QoS):	Yes	Yes	Yes	Yes	Yes
<ul style="list-style-type: none"> • Strict Priority (SP) • Weighted Round Robin (WRR) • Combined SP and WRR • 8 priority queues 					
802.1p priority override	Yes	Yes	Yes	Yes	Yes
802.1p marking	Yes	Yes	Yes	Yes	Yes
DiffServ support	Yes	Yes	Yes	Yes	Yes
DSCP-based QoS	Yes	Yes	Yes	Yes	Yes
QoS mappings	Yes	Yes	Yes	Yes	Yes
User-configurable scheduler profiles	No	No	Yes	Yes	Yes

QoS overview

Quality of Service (QoS) features are used to prioritize the use of bandwidth in a switch. When QoS features are enabled, traffic is classified as it arrives at the switch, and processed through on the basis of configured priorities. Traffic can be dropped, prioritized for guaranteed delivery, or subject to limited delivery options as configured by a number of different mechanisms.

This chapter describes how QoS is implemented and configured in FastIron devices.

Classification is the process of selecting packets on which to perform QoS, reading the QoS information, and assigning a priority to the packets. The classification process assigns a priority to packets as they enter the switch. These priorities can be determined on the basis of information contained within the packet or assigned to the packet as it arrives at the switch. Once a packet or traffic flow is classified, it is mapped to a forwarding priority queue.

Packets on Brocade devices are classified in up to eight traffic classes with values from 0 to 7. Packets with higher priority classifications are given a precedence for forwarding.

Processing of classified traffic

The *trust level* in effect on an interface determines the type of QoS information the device uses for performing QoS. The Brocade device establishes the trust level based on the configuration of various features and whether the traffic is switched or routed. The trust level can be one of the following:

- Ingress port default priority.
- Static MAC address.
- Layer 2 Class of Service (CoS) value – This is the 802.1p priority value in the Ethernet frame. It can be a value from 0 through 7. The 802.1p priority is also called the *Class of Service*.
- Layer 3 Differentiated Services Code Point (DSCP) – This is the value in the six most significant bits of the IP packet header 8-bit DSCP field. It can be a value from 0 through 63. These values are described in RFCs 2472 and 2475. The DSCP value is sometimes called the *DiffServ value*. The device automatically maps the DSCP value of a packet to a hardware forwarding queue. Refer to “[Viewing QoS settings](#)” on page 1992.
- ACL keyword – An ACL can also prioritize traffic and mark it before sending it along to the next hop. This is described under “[QoS options for IP ACLs](#)” on page 1740.

Given the variety of different criteria, there are many possibilities for traffic classification within a stream of network traffic. For this reason, the priority of packets must be resolved based on which criteria takes precedence. Precedence follows the schemes illustrated in [Figure 220](#) through [Figure 222](#).

Determining the trust level of a packet

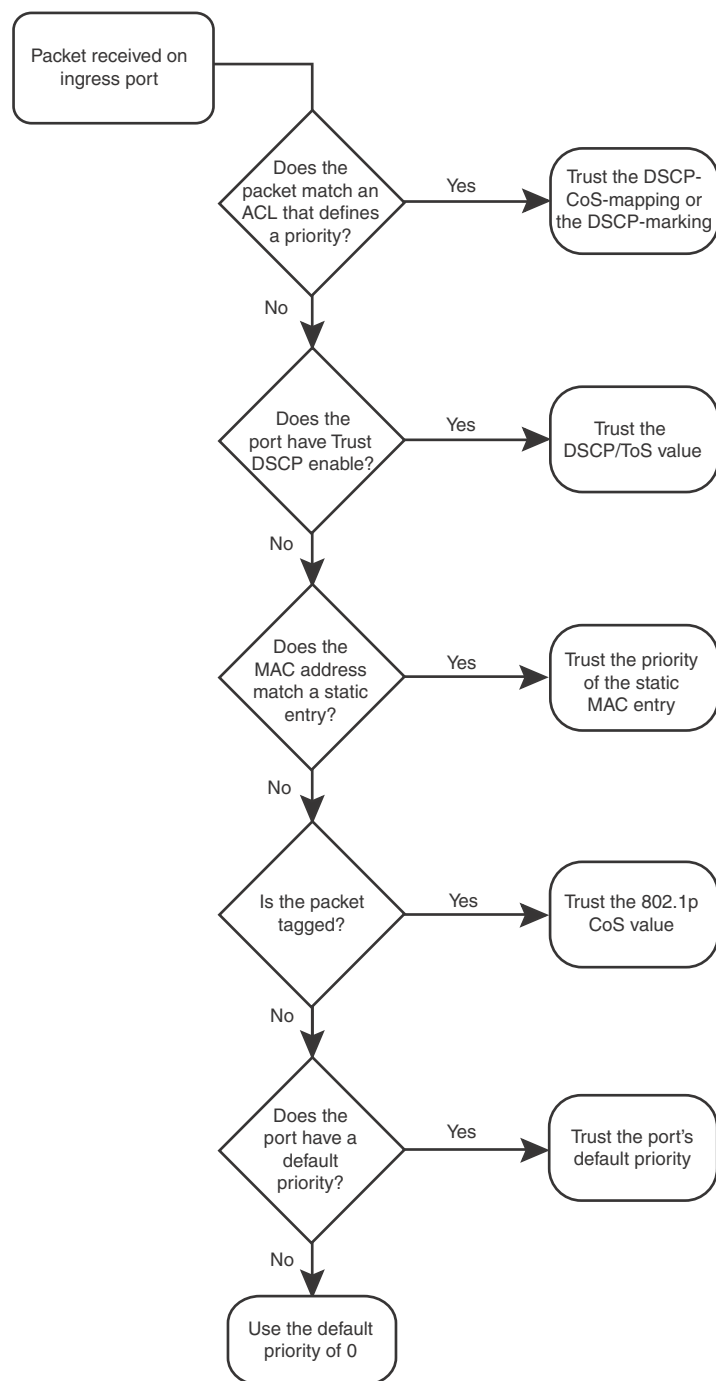
Packet trust level is determined differently on FastIron X Series devices than on FastIron WS and Brocade FCX and ICX series devices.

[Figure 220](#) illustrates how FastIron X Series devices determine the trust level of a packet.

NOTE

[Figure 220](#) is not applicable to the 48-port 10/100/1000 Mbps Ethernet POE interface module (SX-FI48GPP). To determine the trust level of a packet for the SX-FI48GPP interface module, refer to [Figure 221](#).

As shown in the flowchart, the first criteria considered is whether the packet matches on an ACL that defines a priority. Next, it checks if trust DSCP is enabled on the port. If this is not the case, the packet is next classified based on the static MAC address. If this is not true and the packet is tagged, the packet is classified with the 802.1p CoS value. If none of these is true, the packet is next classified based on the ingress port default priority or the default priority of zero (0).

FIGURE 220 Determining a packet trust level - FastIron X Series devices

[Figure 221](#) on page 1967 illustrates how the SX-FI48GPP interface module determines the trust level of a packet. The marking process for the SX-FI48GPP interface module is similar to the marking process for other FastIron SX modules. However, there are major differences between the SX-FI48GPP interface module and other FastIron SX modules.

- For the SX-FI48GPP interface module, static MAC priority takes higher precedence than VLAN priority. For other FastIron SX modules, VLAN priority takes higher precedence over static MAC priority.
- For other FastIron SX modules, the priority of the dynamically learned MAC address is inherited from the default port priority. For the SX-FI48GPP interface module, the priority of the dynamically learned MAC address is not inherited from the default port priority because it is not desirable to allow the port priority to take precedence over the VLAN priority. All dynamically learned MAC addresses are assigned a priority of 0 in the SX-FI48GPP interface module. Therefore, configuring a static MAC with a priority of 0 has no effect on QoS marking.

FIGURE 221 Determining a packet trust level - SX-FI48GPP, SX-FI-24GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI-8XG modules

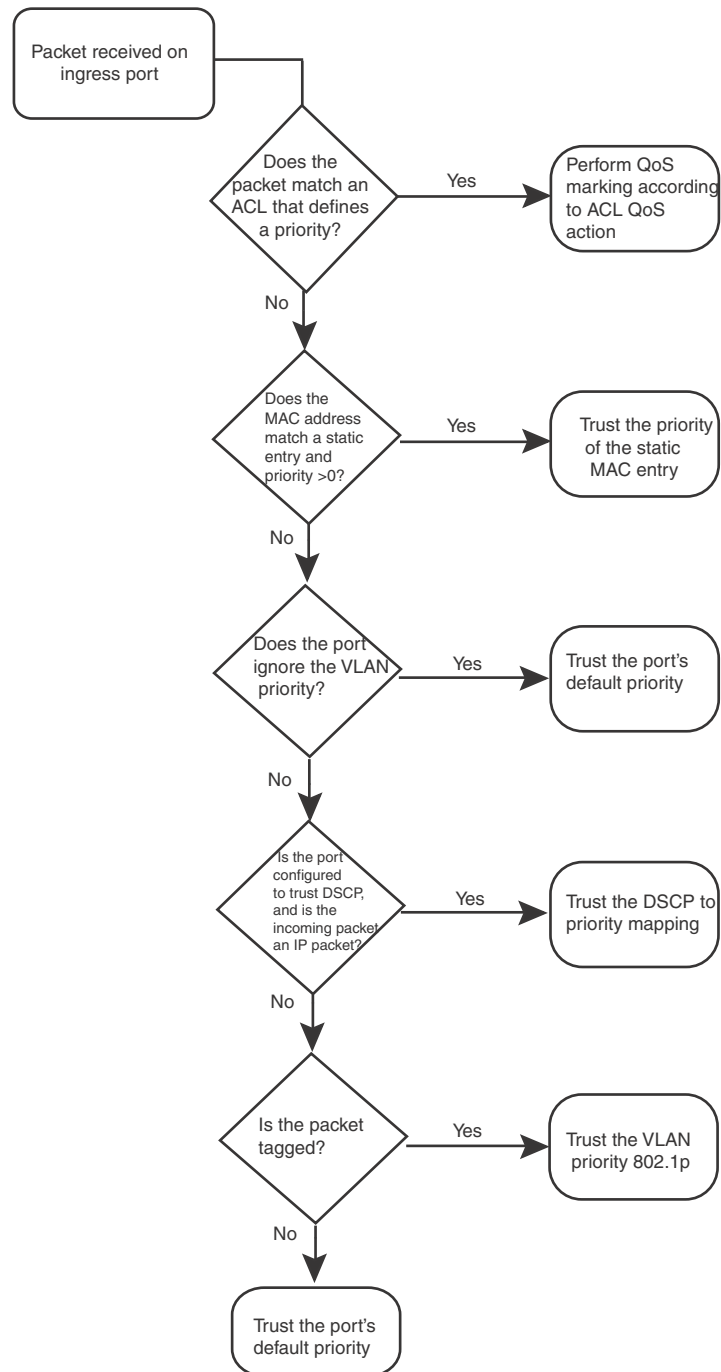
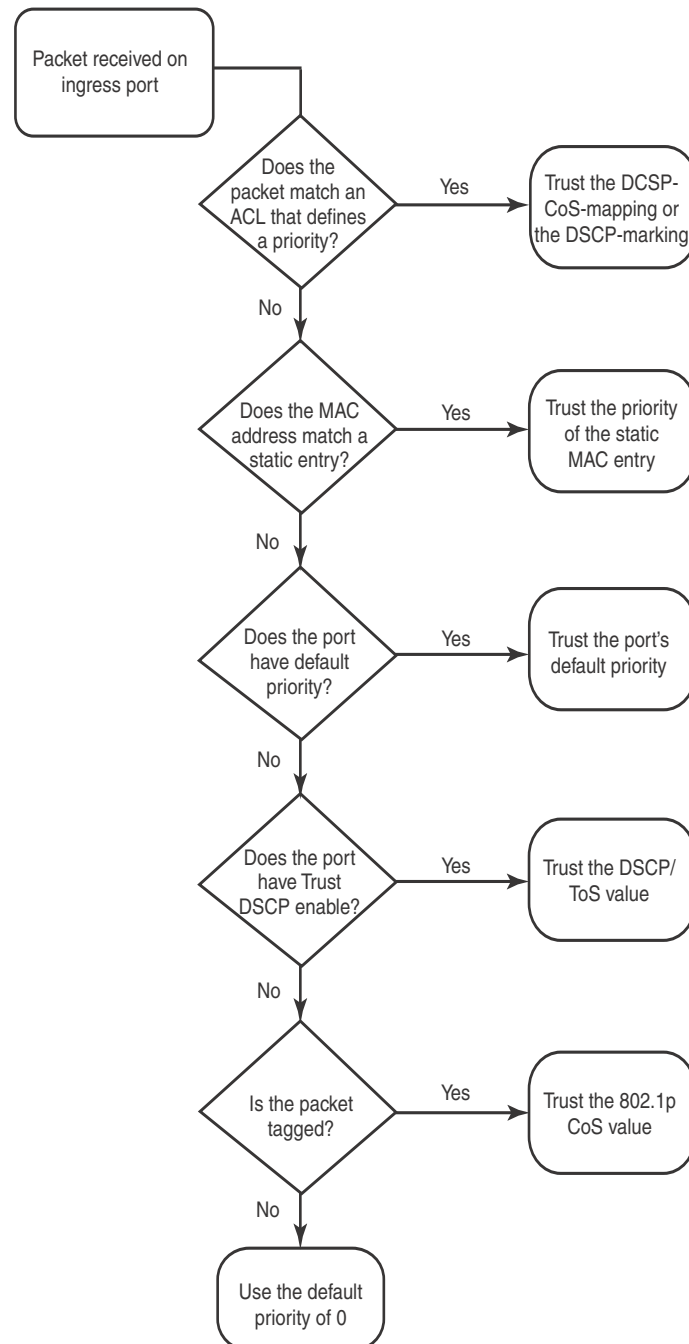


Figure 222 illustrates how FastIron WS and Brocade FCX and ICX series devices determine the trust level of a packet. As shown in the flowchart, the first criteria considered is whether the packet matches on an ACL that defines a priority. If this is not the case and the MAC address of the packet matches a static entry, the packet is classified with the priority of the static MAC entry. If neither of these is true, the packet is next classified with the ingress port default priority. then DSCP/ToS value, then 802.1p CoS value, and finally the default priority of zero (0).

FIGURE 222 Determining a packet trust level - FastIron WS and Brocade FCX Series and ICX devices



Once a packet is classified, it is mapped to a forwarding queue. For all products except the SX-FI48GPP interface module and ICX 6430 switch, there are eight queues designated from 0 through 7. The internal forwarding priority maps to one of these eight queues. For the SX-FI48GPP interface module and ICX 6430 switch, internal forwarding priority maps to four forwarding queues. The mapping between the internal priority and the forwarding queue cannot be changed.

Table 329 through Table 332 show the default QoS mappings for FCX platforms that are used if the trust level for CoS or DSCP is enabled. For information on the SX-FI48GPP interface module, refer to “Queues for the SX-FI48GPP interface module” on page 1972. For information on default QoS mappings for the ICX 6430 switch, refer to “Queues for the ICX 6430 switch” on page 1973.

TABLE 329 Default QoS mappings for FCX platforms, columns 0 to 15

DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
802.1p (CoS) value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
Internal forwarding priority	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Forwarding queue	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

TABLE 330 Default QoS mappings for FCX platforms, columns 16 to 31

DSCP value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
802.1p (CoS) value	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
DSCP value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Internal forwarding priority	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
Forwarding queue	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3

TABLE 331 Default QoS mappings for FCX platforms, columns 32 to 47

DSCP value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
802.1p (CoS) value	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
DSCP value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Internal forwarding priority	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
Forwarding queue	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5

TABLE 332 Default QoS mappings for FCX platforms, columns 48 to 63

DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
802.1p (CoS) value	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Internal forwarding priority	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
Forwarding queue	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7

Mapping between the DSCP value and forwarding queue cannot be changed. However, mapping between DSCP values and other properties can be changed as follows:

- DSCP to internal forwarding priority mapping – You can change the mapping between the DSCP value and the internal forwarding priority value from the default values shown in [Table 329](#) through [Table 332](#). This mapping is used for CoS marking and determining the internal priority when the trust level is DSCP. Refer to [“Changing the DSCP to internal forwarding priority mappings”](#) on page 1984.
- VLAN priority (802.1p) to hardware forwarding queue - You can change the mapping between the 802.1p value and hardware forwarding queue from the default value. Refer to [“Changing the VLAN priority 802.1p to hardware forwarding queue mappings”](#) on page 1985.

QoS for Brocade stackable devices

Brocade FastIron units in an IronStack support QoS. Units in a stack communicate the stack topology information and other proprietary control information through the stacking links. For more information about stacking links and IronStack technology, refer to [Chapter 7, “Brocade Stackable Devices”](#).

In addition to control information, the stacking links also carry user network data packets. In an IronStack topology, the priority of stacking-specific control packets is elevated above that of data path packets, preventing loss of control packets, and timed retries that affect performance. This prioritization also prevents stack topology changes that may occur if enough stack topology information packets are lost.

IronStack technology reserves one QoS profile to provide a higher priority for stack topology and control traffic.

QoS profile restrictions in an IronStack

In a stacking topology, because CoS level 7 is reserved for stacking, quality profiles for qosp7 cannot be configured. If an attempt is made to configure a profile for qosp7, the system ignores the configuration.

NOTE

This applies only when the device is operating in stacking mode. It does not apply to stand-alone devices.

QoS behavior for trusting Layer 2 (802.1p) in an IronStack

By default, Layer 2 trust is enabled. Because priority 7 is reserved for stacking control packets, any ingress data traffic with priority 7 is mapped to internal hardware queue 6. All other priorities are mapped to their corresponding queues.

QoS behavior for trusting Layer 3 (DSCP) in an IronStack

When the trust dscp mode is enabled, packets arriving with DSCP values 56 to 63 are mapped to internal hardware queue 6. All other DSCP values are mapped to their corresponding internal hardware queues.

QoS behavior on port priority and VLAN priority in an IronStack

Port priority has a higher precedence than the 802.1p priority examination. If port priority is set to 7, all incoming traffic is mapped to internal hardware queue 6.

When stacking is not enabled on a device, all priorities are mapped to their corresponding queues without restrictions.

QoS behavior for 802.1p marking in an IronStack

By default when stacking mode is enabled, 802.1p marking is enabled. Outgoing tagged traffic is marked with 802.1p in the VLAN tag based on the internal hardware queue into which ingress traffic was classified. As an exception, QoS priority 7 mapped to priority 6 will have the marking for hardware queue 7.

When stacking mode is disabled, outgoing traffic is not marked with 802.1p based on the internal hardware queue.

QoS queues

Brocade devices support the eight QoS queues (qosp0 through qosp7) listed in [Table 333](#).

TABLE 333 QoS queues

QoS priority level	QoS queue
0	qosp0 (lowest priority queue)
1	qosp1
2	qosp2
3	qosp3
4	qosp4
5	qosp5
6	qosp6
7	qosp7 (highest priority queue)

The queue names listed in [Table 333](#) are the default names. If desired, you can rename the queues as shown in [“Renaming the queues”](#) on page 1989.

Packets are classified and assigned to specific queues based on the criteria shown in [Figure 220](#), [Figure 221](#), and [Figure 222](#).

For FCX and ICX devices, ingress packets are classified into the eight priorities, which map to eight hardware queues or traffic classes (TCs) based on the priority. Exceptions to this model are the SX-FI48GPP and SX-FI-8XG interface modules and the ICX 6430 switch as explained in the following sections.

Queues for the SX-FI48GPP interface module

The SX-FI48GPP interface module consists of two separate hardware Network Processors (NPs). The front-end NP supports four hardware queues, and the back-end NP supports eight hardware queues. Ingress packets are classified into eight priorities mapped into four hardware queues. In the egress, traffic is destined to two adjacent network ports (for example, ports 1/1 and 1/2), and aggregated into one 1-GbE port in the back-end NP. The two network ports share the same hardware queues, and therefore they have the same buffer and descriptor limits and scheduling algorithm for transmission.

Ingress packets are classified into eight QoS priority levels at the front-end NP of the SX-FI48GPP module. The eight priorities are mapped into four hardware queues based on the priority queue configuration in [Table 334](#). QoS priority 7 is the highest priority, and QoS 0 is the lowest priority.

TABLE 334 Priority queues for the SX-FI48GPP

QoS priority level	Hardware queues (traffic classes)
0	0
1	0
2	1
3	1
4	2
5	2
6	3
7	3

QoS classification occurs in two iterations; initially in the front-end NP, followed by the back-end NP. The back-end NP has the same classification and marking capabilities of existing FastIron SX interface modules, but the front-end NP does not support ACL and static MAC priority. The front-end NP supports basic QoS features, such as port priority, QoS-ToS mapping, 802.1p to priority mapping, 802.1p override, and trust DSCP mode.

The default scheduling configuration for Weighted Round Robin (WRR), Hybrid WRR and Strict Priority (SP), and SP mode for the eight QoS priority queues mapped to the four hardware queues is described under [“Default scheduling configuration for the SX-FI48GPP module”](#) on page 1985.

Queues for the SX-FI-8XG interface module

The SX-FI-8XG interface module consists of two separate hardware Network Processors (NP). The front-end NP supports 8 hardware queues, and the back-end NP supports eight hardware queues. In the egress, traffic is destined to four adjacent ports (for example, ports 1/1 to 1/4), and aggregated into one 10GbE port in the back-end NP. The four network ports share the same hardware queues; therefore, they have the same buffer and descriptor limits and scheduling algorithm for transmission.

QoS classification occurs in two iterations; initially in the front-end NP, followed by the back-end NP. The back-end NP has the same classification and marking capabilities of existing FastIron SX interface modules, however, the front -end NP does not support ACL and static MAC priority. The front-end NP supports basic QoS features, such as port priority, qos-tos mapping, 802.1p to priority mapping, 802.1p override, and trust-dscp mode.

Queues for the ICX 6430 switch

For the ICX 6430 switch, ingress packets are classified into eight QoS priority levels. These are mapped internally to four hardware forwarding queues or traffic classes as shown in [Table 335](#). QoS priority 7 is the highest priority, and QoS 0 is the lowest QoS priority (qosp) level.

TABLE 335 Priority queues for the ICX 6430

QoS priority level	Hardware queues (Traffic classes)
0	0
1	0
2	1
3	1
4	1
5	2
6	2
7	3

For the ICX 6430 switch, internal forwarding priority maps to hardware forwarding queues 0 through 3. The mapping between the internal priority and hardware forwarding queue cannot be changed. [Table 336](#) through [Table 339](#) shows the default QoS mappings that are used if the trust level for CoS or DSCP is enabled. Mappings are the same for stand-alone and stacking systems.

TABLE 336 Default QoS mappings for ICX 6430, columns 0 to 15

DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
802.1p (CoS) value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15

TABLE 336 Default QoS mappings for ICX 6430, columns 0 to 15 (Continued)

DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
Internal forwarding priority	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Forwarding queue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE 337 Default QoS mappings for ICX 6430, columns 16 to 31

DSCP value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
802.1p (CoS) value	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
DSCP value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Internal forwarding priority	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
Forwarding queue	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE 338 Default QoS mappings for ICX 6430, columns 32 to 47

DSCP value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
802.1p (CoS) value	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
DSCP value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Internal forwarding priority	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
Forwarding queue	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2

TABLE 339 Default QoS mappings for ICX 6430, columns 48 to 63

DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
802.1p (CoS) value	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Internal forwarding priority	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
Forwarding queue	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3

Mapping between DSCP value and forwarding queue cannot be changed. However, mapping between DSCP values and other properties can be changed as follows:

- DSCP to internal forwarding priority mapping – You can change the mapping between the DSCP value and the internal forwarding priority value from the default values shown in [Table 336](#) through [Table 339](#). This mapping is used for CoS marking and determining the internal priority when the trust level is DSCP. Refer to [“Changing the DSCP to internal forwarding priority mappings”](#) on page 1984.
- VLAN priority (802.1p) to hardware forwarding queue - You can change the mapping between the 802.1p value and hardware forwarding queue from the default value. Refer to [“Changing the VLAN priority 802.1p to hardware forwarding queue mappings”](#) on page 1985

User-configurable scheduler profile

The user-configurable scheduler profile is a template that defines either the scheduling mechanism or scheduling profile (weights assigned to the queues) or both for the egress queues. A configured user-configurable scheduler profile for egress queues can be applied to any hardware device. The default QoS is applicable to the entire system. If the scheduler profile is configured using the `qos mech strict` command, all devices in the system will be configured with the strict priority. The user-configurable scheduler profile is applicable only to the specific devices, leaving the remaining devices running default QoS. On any device, the user-configurable scheduler profile has high priority over the default QoS. On any device, user-configurable scheduler profile has high priority over the default QoS. The user-configurable scheduler profile should be in line with default QoS commands in both stacking and stand-alone systems.

User-configurable scheduler profile configuration

Configuring a user-configurable scheduler profile involves, selecting a proper mechanism and appropriate weights for the traffic classes (TCs) corresponding to that mechanism. It is highly recommended that you let the system use the default scheduling mechanism unless user knows what parameters you intend to modify and for what reasons.

There are two ways of creating a user-configurable scheduler profile. The scheduler-profile can be created either by specifying a mechanism (WRR, Strict, or Mixed) or by specifying weights.

The user-configurable scheduler profile can be created by specifying a mechanism. There are three available mechanisms:

- Strict Priority (SP)
- Weighted Round Robin (WRR)
- Mixed (combination of SP and WRR)

Following is the command format for creating a profile while specifying a mechanism.

qos scheduler-profile <user_profile_name> **mechanism** <scheduling_mechanism>

Syntax: :

The <user_profile_name> variable is the name of the profile you are creating.

The <scheduling_mechanism> variable is SP, WRR or Mixed.

The user-configurable scheduler profile can be created by specifying weights from **qosp0** through **qosp7** as shown in the following command format.

qos scheduler-profile <user_profile_name> **profile** **qosp0** <w0> **qosp7** <w7>

Syntax:

The `<user_profile_name>` variable is the name of the profile you are creating.

Profile **qosp0** through **qosp7** are the default queue names.

The `<w0>` through `<w7>` variables are the assigned weights.

If you create a profile specifying only the weights (**qosp0** through **qosp7**) without specifying the mechanism, the default mechanism is used. The default mechanism for stacking systems is **Mixed**, and **WRR** for stand-alone systems.

If you change the profile mechanism, the weights also get changed according to the mechanism. The weights can be modified according to the following requirements:

- If the mechanism is changed to **WRR**, the default system weights get assigned
- If the mechanism is changed to **Mixed**, the default mix weights get assigned
- If the mechanism is changed to **Strict**, the weights are ignored and remain untouched.

Scheduler-profile modifications take effect dynamically on an active profile. The operational defaults for all scheduling types for stacking and stand-alone systems are listed in [Table 340](#).

Displaying the user-configurable scheduler profile configuration

To display the specified user-configurable scheduler profile configuration, use the **show scheduler-profile** `<user_profile_name>` command.

Syntax:

The `<user_profile_name>` variable is the name of the profile you are creating.

To display all the scheduler profiles configured in the runtime configuration for the system, use the **show scheduler-profile all** command.

FCX and ICX 6450 platforms

[Table 340](#) shows the default values for the scheduling type for stacking and stand-alone FCX and ICX 6450 platforms.

TABLE 340 Default values for scheduling type for stacking and stand-alone systems (for FCX and ICX 6450 platforms).

Stacking system (common for FCX platforms)

Traffic Class	SP	SP Jumbo	WRR	WRR Jumbo	Mixed	Mixed Jumbo
TC0	SP	SP	3	8	15	15
TC1	SP	SP	3	8	15	15
TC2	SP	SP	3	8	15	15
TC3	SP	SP	3	8	15	15
TC4	SP	SP	3	8	15	15
TC5	SP	SP	10	16	25	25
TC6	SP	SP	75	44	SP	SP
TC7	SP	SP	SP	SP	SP	SP

Stand-alone system (common for FCX platforms)

	SP	SP Jumbo	WRR	WRR Jumbo	Mixed	Mixed Jumbo
TC0	SP	SP	3	8	15	15
TC1	SP	SP	3	8	15	15
TC2	SP	SP	3	8	15	15
TC3	SP	SP	3	8	15	15
TC4	SP	SP	3	8	15	15
TC5	SP	SP	3	8	25	25
TC6	SP	SP	7	8	SP	SP
TC7	SP	SP	75	44	SP	SP

ICX 6430 platforms

[Table 340](#) shows the default values for scheduling type for stacking and stand-alone ICX 6430 platforms. The lowest weighted priority is for qosp0, while the highest is for qosp7.

Note that values are provided for QoS priority (QSP) levels. The weights applied to the traffic class (TC) are the sum of the weights of the QSP levels that map to that TC. For example, QSP0 and QSP1 map to TC0. If the weight for QSP0 is 6 and the weight for QSP1 is 6, then the weight for TC0 is 12. Refer to [Table 335](#) on page 1973 for QoS priority to traffic class mapping.

TABLE 341 Default values for scheduling type for stacking and stand-alone systems (for ICX 6430 platforms)

Stacking System for scheduling type for stacking and stand-alone systems (for ICX 6430 platforms)

QSP Level	SP	SP Jumbo	WRR	WRR Jumbo	Mixed	Mixed Jumbo
QSP0	SP	SP	3	8	15	15
QSP1	SP	SP	3	8	15	15
QSP2	SP	SP	3	8	15	15
QSP3	SP	SP	3	8	15	15
QSP4	SP	SP	3	8	40	40
QSP5	SP	SP	10	16	SP	SP
QSP6	SP	SP	75	44	SP	SP
QSP7	SP	SP	SP	SP	SP	SP

Stand-alone system for ICX 6430 platforms

	SP	SP Jumbo	WRR	WRR Jumbo	Mixed	Mixed Jumbo
QSP0	SP	SP	3	8	15	15
QSP1	SP	SP	3	8	15	15
QSP2	SP	SP	3	8	15	15
QSP3	SP	SP	3	8	15	15
QSP4	SP	SP	3	8	40	40
QSP5	SP	SP	3	8	SP	SP

Stand-alone system for ICX 6430 platforms (Continued)

QSP6	SP	SP	7	8	SP	SP
QSP7	SP	SP	75	44	SP	SP

QoS priorities-to-traffic assignment

By default, all traffic is in the best-effort queue (qosp0) and is honored on tagged ports on all FastIron switches. You can assign traffic to a higher queue based on the following:

- Incoming port (sometimes called the *ingress port*)
- Static MAC entry

When you change the priority, you specify a number from 0 through 7. The priority number specifies the IEEE 802.1 equivalent to one of the eight QoS queues on Brocade devices. The numbers correspond to the queues as shown in [Table 333](#).

Although it is possible for a packet to qualify for an adjusted QoS priority based on more than one of the criteria, the system always gives a packet the highest priority for which it qualifies. Thus, if a packet is entitled to the premium queue because of its IP source and destination addresses, but is entitled only to the high queue because of its incoming port, the system places the packet in the premium queue on the outgoing port.

Changing a port priority

To change the QoS priority of port 1/1 to the premium queue (qosp7), enter the following commands.

```
Brocade(config)#interface ethernet 1/1
Brocade(config-if-e1000-1/1)#priority 7
```

The device assigns priority 7 to untagged switched traffic received on port 1/1.

Use the following command to assign priority levels.

[no] priority <num>

Syntax:

The <num> variable can be from 0 through 7 and specifies the IEEE 802.1 equivalent to one of the eight QoS queues listed in [Table 333](#).

Assigning static MAC entries to priority queues

By default, all MAC entries are in the best-effort queue. When you configure a static MAC entry, you can assign the entry to a higher QoS level.

To configure a static MAC entry and assign the entry to the premium queue, enter commands such as the following.

```
Brocade(config)#vlan 9
Brocade(config-vlan-9)#static-mac-address 1145.1163.67FF ethernet 1/1 priority 7
Brocade(config-vlan-9)#write memory
```

Use the following command to configure a MAC entry and assign the entry to a priority queue.

[no] **static-mac-address** <mac-addr> **ethernet** <port> [priority <num>]

Syntax:

The <mac-addr> is the MAC address.

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

The **priority** <num> variable can be from 0 through 7 and specifies the IEEE 802.1 equivalent to one of the eight QoS queues.

NOTE

The location of the **static-mac-address** command in the CLI depends on whether you configure port-based VLANs on the device. If the device does not have more than one port-based VLAN (VLAN 1, which is the default VLAN containing all ports), the **static-mac-address** command is at the global CONFIG level of the CLI. If the device has more than one port-based VLAN, then the **static-mac-address** command is not available at the global CONFIG level. In this case, the command is available at the configuration level for each port-based VLAN.

Buffer allocation and threshold for QoS queues

By default, Brocade IronWare software allocates a certain number of buffers to the outbound transport queue for each port based on QoS priority. The buffers control the total number of packets permitted in the outbound queue for the port. If desired, you can increase or decrease the maximum number of outbound transmit buffers allocated to all QoS queues, or to specific QoS queues on a port or group of ports. For more information, refer to [“Dynamic buffer allocation for QoS priorities for FastIron X Series devices”](#) on page 584.

802.1p priority override

You can configure a port to ignore the 802.1p priority for traffic classification for an incoming packet. When this feature is enabled, packets will be classified as follows:

- If the packet matches an ACL that defines the priority, then ACL priority will be used.
- If the packet source or destination MAC address matches a configured static MAC address with priority, then static MAC priority will be used.
- If the ingress port has a configured priority, then port priority will be used.
- If the other situations do not apply, the configured or default port priority (0) will be used.

Note that the original 802.1p priority in the packet will be retained. This feature does not re-mark the 802.1p value.

Configuration notes and feature limitations

- 802.1p priority override is supported on physical ports and trunk ports. When applied to the primary port of a trunk group, the configuration applies to all members of the trunk group.
- This feature is not supported together with trust dscp.

Enabling 802.1p priority override

To enable 802.1p priority override, enter the following command at the interface level of the CLI.

```
Brocade(config-if-e1000-2)#priority ignore-8021p
```

Syntax: [no] priority ignore-8021p

Use the following command to show whether 802.1p priority override is enabled on a port.

```
Brocade#show run interface ethernet 1
interface ethernet 1
    priority ignore-8021p
```

Syntax: show run interface ethernet <port>

Specify the *port* variable in one of the following formats:

- FWS, FCX, and ICX stackable switches – *stack-unit/slotnum/portnum*
- FSX 800 and FSX 1600 chassis devices – *slotnum/portnum*
- ICX devices – *slotnum/portnum*
- FESX compact switches – *portnum*

Marking

Marking is the process of changing the packet QoS information (the 802.1p and DSCP information in a packet) for the next hop. For example, for traffic coming from a device that does not support Differentiated Services (DiffServ), you can change the packet IP precedence value into a DSCP value before forwarding the packet.

You can mark a packet's Layer 2 CoS value, its Layer 3 DSCP value, or both values. The Layer 2 CoS or DSCP value the device marks in the packet is the same value that results from mapping the packet QoS value into a Layer 2 CoS or DSCP value.

Marking is optional and is disabled by default. Marking is performed using ACLs. When marking is not used, the device still performs the mappings listed in [“QoS overview”](#) on page 1963 for scheduling the packet, but leaves the packet QoS values unchanged when the device forwards the packet.

For configuration syntax, rules, and examples of QoS marking, refer to [“QoS options for IP ACLs”](#) on page 1740.

DSCP-based QoS configuration

Brocade IronWare releases support basic DSCP-based QoS (also called Type of Service (ToS)-based QoS) as described in this chapter. However, the FastIron family of switches does not support other advanced DSCP-based QoS features as described in the *Enterprise Configuration and Management Guide*.

Brocade IronWare releases also support marking of the DSCP value. The software can read Layer 3 Quality of Service (QoS) information in an IP packet and select a forwarding queue for the packet based on the information. The software interprets the value in the six most significant bits of the IP packet header 8-bit ToS field as a DSCP value, and maps that value to an internal forwarding priority.

The internal forwarding priorities are mapped to one of the eight forwarding queues (qosp0 through qosp7) on the Brocade device. During a forwarding cycle, the device gives more preference to the higher-numbered queues, so that more packets are forwarded from these queues. For example, queue qosp7 receives the highest preference, while queue qosp0, the best-effort queue, receives the lowest preference.

Application notes for DSCP-based QoS

- DSCP-based QoS is not automatically honored for routed and switched traffic. The default is 802.1p to CoS mapping. To honor DSCP-based QoS, you must either use an ACL or enable trust DSCP. Refer to [“Using ACLs to honor DSCP-based QoS”](#) on page 1981.
- When DSCP marking is enabled, the device changes the contents of the inbound packet ToS field to match the DSCP-based QoS value. This differs from BigIron, which marks the outbound packet ToS field.

Using ACLs to honor DSCP-based QoS

This section shows how to configure Brocade devices to honor DSCP-based QoS for routed and switched traffic.

FastIron stackable devices

FastIron WS and Brocade FCX Series devices support DSCP-based QoS on a per-port basis. DSCP-based QoS is not automatically honored for switched traffic. The default is 802.1p to CoS mapping. To honor DSCP-based QoS, enter the following command at the interface level of the CLI.

[no] trust dscp

To disable the configuration, use the no form of the command.

When trust dscp is enabled, the interface honors the Layer 3 DSCP value. By default, the interface honors the Layer 2 CoS value.

NOTE

The **trust dscp** command is not supported with 802.1p priority override.

FastIron X Series devices

FastIron X Series devices require the use of an ACL to honor DSCP-based QoS for routed traffic in the Layer 3 image, or for switched traffic in the Layer 2 image. To enable DSCP-based QoS on these devices, apply an ACL entry such as the following.

```
Brocade(config)#access-list 101 permit ip any any dscp-cos-mapping
```

NOTE

Use the **bridged-routed** keyword in the ACL to honor DSCP for switched traffic in the Layer 3 image. Refer to [“Enabling ACL support for switched traffic in the router image”](#) on page 1733.

NOTE

The **access-list 101 permit ip any any dscp-cos-mapping** command is supported on the SX-FI48GPP interface module. For more information on QoS queues for the SX-FI48GPP interface module, refer to [“Queues for the SX-FI48GPP interface module”](#) on page 1972.

Trust DSCP for the SX-FI48GPP, SX-FI-24GPP, SX-FI-24HF, SX-FI-2XG, and SX-FI8XG modules

On the following modules, trust DSCP can be enabled on a per-port basis:

- SX-FI48GPP
- SX-FI-24GPP
- SX-FI-24HF
- SX-FI-2XG
- SX-FI-8XG

Each port on the supported modules corresponds to a front-end panel port. By default, trust VLAN priority is enabled.

NOTE

For all ports in the other FastIron SX modules, ACL should be used to implement the trust DSCP mode.

For example, to enable trust DSCP on interface ethernet 1/48 on the SX-FI48GPP module, enter the following command.

[no] trust dscp

To disable the configuration, use the no form of the command.

Configuring QoS mapping configuration

You can optionally change the following QoS mappings:

- DSCP to internal forwarding priority
- VLAN priority (802.1p) to hardware forwarding queue

The mappings are globally configurable and apply to all interfaces.

Default DSCP to internal forwarding priority mappings

The DSCP values are described in RFCs 2474 and 2475. [Table 342](#) lists the default mappings of DSCP values to internal forwarding priority values.

TABLE 342 Default DSCP to internal forwarding priority mappings

Internal forwarding priority	DSCP value
0 (lowest priority queue)	0 – 7
1	8 – 15
2	16 – 23
3	24 – 31
4	32 – 39
5	40 – 47
6	48 – 55
7 (highest priority queue)	56 – 63

Notice that DSCP values range from 0 through 63, whereas the internal forwarding priority values range from 0 through 7. Any DSCP value within a given range is mapped to the same internal forwarding priority value. For example, any DSCP value from 8 through 15 maps to priority 1.

After performing this mapping, the device maps the internal forwarding priority value to one of the hardware forwarding queues.

On FWS, FCX, and ICX devices, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from the FastIron X Series devices, which support seven priorities for user data instead of eight when sFlow is enabled. QoS queue 1 is reserved for sFlow and not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0. Note that the ICX 6430 does not support sFlow.

[Table 343](#) lists the default mappings of internal forwarding priority values to the hardware forwarding queues for the ICX 6430.

TABLE 343 Default mappings of internal forwarding priority values for the ICX 6430

Internal forwarding priority	Forwarding queues
0 (lowest priority queue)	qosp0
1	qosp0
2	qosp1
3	qosp1
4	qosp1
5	qosp2
6	qosp2
7 (highest priority queue)	qosp3

You can change the DSCP to internal forwarding mappings. You also can change the internal forwarding priority to hardware forwarding queue mappings.

Changing the DSCP to internal forwarding priority mappings

To change the DSCP to internal forwarding priority mappings for all the DSCP ranges, enter commands such as the following at the global CONFIG level of the CLI.

```
Brocade(config)#qos-tos map dscp-priority 0 2 3 4 to 1
Brocade(config)#qos-tos map dscp-priority 8 to 5
Brocade(config)#qos-tos map dscp-priority 16 to 4
Brocade(config)#qos-tos map dscp-priority 24 to 2
Brocade(config)#qos-tos map dscp-priority 32 to 0
Brocade(config)#qos-tos map dscp-priority 40 to 7
Brocade(config)#qos-tos map dscp-priority 48 to 3
Brocade(config)#qos-tos map dscp-priority 56 to 6
```

Use the following command to map priority levels to DSCP values.

[no] qos-tos map dscp-priority <dscp-value> [<dscp-value> ...] to <priority>

Syntax:

The <dscp-value> [<dscp-value> ...] variable specifies the DSCP value ranges you are remapping.

You can specify up to eight DSCP values in the same command, to map to the same forwarding priority.

The <priority> variable specifies the internal forwarding priority.

Following is an example of using this command.

```
qos-tos map dscp-priority 1 2 3 4 5 6 7 8 to 6
```

Following is output displayed from using the **show qos-tos** command as a result of issuing the preceding command.

```
Brocade#show qos-tos
```

Portions of table omitted for simplicity.

DSCP-Priority map: (dscp = d1d2)

	d2	0	1	2	3	4	5	6	7	8	9
d1											
0		1	0	1	1	1	0	0	0	5	1
1		6	1	1	1	1	1	4	2	2	2
2		2	2	2	2	2	3	3	3	3	3
3		3	3	0	4	4	4	4	4	4	4
4		7	5	5	5	5	5	5	5	3	6
5		6	6	6	6	6	6	6	7	7	7
6		7	7	7	7						

This output displays mappings in the DSCP to forwarding priority portion of the QoS information display. To read this part of the display, select the first part of the DSCP value from the d1 column and select the second part of the DSCP value from the d2 row. For example, to read the DSCP to forwarding priority mapping for DSCP value 24, select 2 from the d1 column and select 4 from the d2 row. The mappings that are changed by the example **qos-tos map dscp-priority** command are shown below in bold type.

Changing the VLAN priority 802.1p to hardware forwarding queue mappings

To map a VLAN priority to a different hardware forwarding queue, enter commands such as the following at the global CONFIG level of the CLI.

```
[no] qos tagged-priority <num> <queue>
```

Syntax:

The <num> variable can be from 0 through 7 and specifies the VLAN priority.

The <queue> variable specifies the hardware forwarding queue to which you are reassigning the priority.

The default queue names are as follows:

- qosp7
- qosp6
- qosp5
- qosp4
- qosp3
- qosp2
- qosp1
- qosp0

Following is an example of using this command.

```
Brocade(config)#qos tagged-priority 2 qosp0
```

Default scheduling configuration for the SX-FI48GPP module

The default scheduling configuration for Weighted Round Robin (WRR), Strict Priority (SP), and mixed WRR and SP mode for the eight QoS priority (qosp) queues mapped to the four hardware queues is described in [Table 344](#).

TABLE 344 Default configuration for 8 to 4 queues for the SX-FI48GPP module

Hardware Queue	Weighted Round Robin (WRR) mode	Mixed WRR and SP	Strict Priority (SP) mode
3	Weight 82%	Strict Priority	Strict Priority
2	Weight 6%	Weight 40%	Strict Priority
1	Weight 6%	Weight 30%	Strict Priority
0	Weight 6%	Weight 30%	Strict Priority

Note that [Table 344](#) includes values for default, non-jumbo mode WRR. The hardware queues are calculated using default qosp values from [Table 340](#) on page 1976 as follows:

- Front end queue 3= 75% (qosp7) + 7% (qosp6) = 82%

- Front end queue 2 = 3% (qosp4) + 3% (qosp5) = 6%
- Front end queue 1 = 3% (qosp2) + 3% (qosp3) = 6%
- Front end queue 0 = 3% (qosp0) + 3% (qosp1) = 6%

The hardware queues for mixed WRR and SP mode are calculated as follows:

- Front end queue 3 is Strict Priority as default values for qosp7 and qosp6 are SP
- Front end queue 2 = 25% (qosp4) + 15% (qosp5) = 40%
- Front end queue 1 = 15% (qosp2) + 15% (qosp3) = 30%
- Front end queue 0 = 15% (qosp0) + 15% (qosp1) = 30%

Default scheduling configuration for the ICX 6430

The default scheduling configuration for Weighted Round Robin (WRR), Strict Priority (SP), and mixed WRR and SP mode for the eight QoS priority (qosp) queues mapped to the four hardware queues for an ICX 6430 is described in [Table 345](#).

TABLE 345 Default configuration for 8 to 4 queues (stand-alone system)

Hardware queue	Weighted Round Robin (WRR) mode	Mixed WRR and SP	Strict Priority (SP) mode
3	Weight 75%	Strict Priority	Strict Priority
2	Weight 10%	Strict Priority	Strict Priority
1	Weight 9%	Weight 70%	Strict Priority
0	Weight 6%	Weight 30%	Strict Priority

[Table 345](#) includes values for default, non-jumbo mode WRR for a stand-alone system. The hardware queues are calculated using default qosp values from [Table 341](#) on page 1977 as follows.

- Queue 3 = 75% (qosp7)
- Queue 2 = 3% (qosp5) + 7% (qosp6) = 10%
- Queue 1 = 3% (qosp2) + 3% (qosp3) + 3% (qosp4) = 9%
- Queue 0 = 3% (qosp0) + 3% (qosp1) = 6%

The hardware queues for mixed WRR and SP mode are calculated as follows:

- Queue 3 is Strict Priority as the default value for qosp7 is SP
- Queue 2 is Strict Priority as default values for qosp5 and qosp6 are SP
- Queue 1 = 15% (qosp2) + 15% (qosp3) + 40% (qosp4) = 70%
- Queue 0 = 15% (qosp0) + 15% (qosp1) = 30%

NOTE

If any qosp value is SP, then the weight of the hardware queue is SP.

Scheduling QoS information

Scheduling is the process of mapping a packet to an internal forwarding queue based on its QoS information, and servicing the queues according to a mechanism.

Scheduling for the SX-FI48GPP module

The SX-FI48GPP module supports scheduling at the front-end and back-end NP. If egress congestion occurs at the front-end NP of the SX-FI48GPP module, scheduling is based on four queues instead of eight. For more information on default configuration for 8 to 4 queue mapping, refer to [Table 334](#) on page 1972. If egress congestion occurs at the back-end of the SX-FI48GPP module, then scheduling is based on eight queues. When SX-FI48GPP ports are running at a reduced speed (100 Mbps or 10 Mbps), egress congestion usually occurs at the front-end NP.

QoS queuing methods

The following QoS queuing methods are supported in all IronWare releases for the FastIron devices:

- **Weighted Round Robin (WRR)** – WRR ensures that all queues are serviced during each cycle. A WRR algorithm is used to rotate service among the eight queues on the FastIron devices. The rotation is based on the weights you assign to each queue. This method rotates service among the queues, forwarding a specific number of packets in one queue before moving on to the next one.

NOTE

In stacking mode, the qosp7 queue is reserved as Strict Priority (SP) under weighted queuing. Attempts to change the qosp7 setting will be ignored.

WRR is the default queuing method and uses a default set of queue weights.

The number of packets serviced during each visit to a queue depends on the percentages you configure for the queues. The software automatically converts the percentages you specify into weights for the queues.

NOTE

Queue cycles on the FastIron devices are based on bytes. These devices service a given number of bytes (based on weight) in each queue cycle. FES and BI/FI queue cycles are based on packets. The bytes-based scheme is more accurate than a packets-based scheme if packets vary greatly in size.

- **Strict Priority (SP)** – SP ensures service for high-priority traffic. The software assigns the maximum weights to each queue, to cause the queuing mechanism to serve as many packets in one queue as possible before moving to a lower queue. This method biases the queuing mechanism to favor the higher queues over the lower queues.

For example, strict queuing processes as many packets as possible in qosp3 before processing any packets in qosp2, then processes as many packets as possible in qosp2 before processing any packets in qosp1, and so on.

- **Hybrid WRR and SP** – This configurable queueing mechanism combines both the SP and WRR mechanisms. The combined method enables the Brocade device to give strict priority to delay-sensitive traffic such as VoIP traffic, and weighted round robin priority to other traffic types.

By default, when you select the combined SP and WRR queueing method, the Brocade device assigns strict priority to traffic in qosp7 and qosp6, and weighted round robin priority to traffic in qosp0 through qosp5. Thus, the Brocade device schedules traffic in queue 7 and queue 6 first, based on the strict priority queueing method. When there is no traffic in queue 7 and queue 6, the device schedules the other queues in round-robin fashion from the highest priority queue to the lowest priority queue.

NOTE

Brocade stackable devices that are operating as members of a stack reserve queue 7 for stacking functions. For more information, refer to [“QoS for Brocade stackable devices”](#) on page 1970.

By default, when you specify the combined SP and WRR queueing method, the system balances the traffic among the queues as shown in [Table 346](#). If desired, you can change the default bandwidth values as shown in [“Bandwidth allocations of the hybrid WRR and SP queues”](#) on page 1991.

TABLE 346 Default bandwidth for combined SP and WRR queueing methods

Queue	Default bandwidth
qosp7	Strict Priority (highest priority)
qosp6	Strict Priority
qosp5	25%
qosp4	15%
qosp3	15%
qosp2	15%
qosp1	15%
qosp0	15% (lowest priority)

Selecting the QoS queueing method

By default, Brocade devices use the WRR method of packet prioritization. To change the method to SP, enter the **qos mechanism strict** command at the global CONFIG level of the CLI.

```
Brocade(config)#qos mechanism strict
```

To change the method back to WRR, enter the **qos mechanism weighted** command.

```
Brocade(config)#qos mechanism weighted
```

Syntax: [no] qos mechanism strict | weighted

To change the queueing mechanism to the combined SP and WRR method, enter the **qos mechanism mixed-sp-wrr** command at the global CONFIG level of the CLI.

```
Brocade(config)#qos mechanism mixed-sp-wrr
```

Syntax: qos mechanism mixed-sp-wrr

Configuring the QoS queues

Each of the queues has the following configurable parameters:

- The queue name
- The minimum percentage of a port outbound bandwidth guaranteed to the queue

Renaming the queues

The default queue names are qosp7, qosp6, qosp5, qosp4, qosp3, qosp2, qosp1, and qosp0. You can change one or more of the names if desired.

To rename queue “qosp3” to “92-octane”, enter the following command.

```
Brocade(config)#qos name qosp3 92-octane
```

Syntax: `qos name <old-name> <new-name>`

The `<old-name>` variable specifies the name of the queue before the change.

The `<new-name>` variable specifies the new name of the queue. You can specify an alphanumeric string up to 32 characters long.

Changing the minimum bandwidth percentages of the WRR queues

If you are using the weighted round robin mechanism instead of the strict priority mechanism, you can change the weights for each queue by changing the minimum percentage of bandwidth you want each queue to guarantee for its traffic.

NOTE

On the SX-FI48GPP interface module, the bandwidth percentages for 8 to 4 queue mapping for WRR queues are different from other Brocade SX modules. For more information on 8 to 4 queue mapping on the SX-FI48GPP interface module, refer to [“Default scheduling configuration for the SX-FI48GPP module”](#) on page 1985.

By default, the eight QoS queues on FastIron devices receive the minimum guaranteed percentages of a port’s total bandwidth, as shown in [Table 347](#). Note that the defaults differ when jumbo frames are enabled.

TABLE 347 Default minimum bandwidth percentages on Brocade devices

Queue	Default minimum percentage of bandwidth	
	Without jumbo frames	With jumbo frames
qosp7	75%	44%
qosp6	7%	8%
qosp5	3%	8%
qosp4	3%	8%
qosp3	3%	8%
qosp2	3%	8%
qosp1	3%	8%
qosp0	3%	8%

When the queuing method is WRR, the software internally translates the percentages into weights. The weight associated with each queue controls how many packets are processed for the queue at a given stage of a cycle through the weighted round robin algorithm.

NOTE

Queue cycles on the FastIron devices are based on bytes. These devices service a given number of bytes (based on the weight) in each queue cycle. FES and BI/FI queue cycles are based on packets. The bytes-based scheme is more accurate than a packets-based scheme if packets vary greatly in size.

The bandwidth allocated to each queue is based on the relative weights of the queues. You can change the bandwidth percentages allocated to the queues by changing the queue weights.

There is no minimum bandwidth requirement for a given queue. For example, queue qosp3 is not required to have at least 50 percent of the bandwidth.

To change the bandwidth percentages for the queues, enter commands such as the following. Note that this example uses the default queue names.

```
Brocade(config)#qos profile qosp7 25 qosp6 15 qosp5 12 qosp4 12 qosp3 10 qosp2
10 qosp1 10 qosp0 6
Profile qosp7      : Priority7    bandwidth requested 25% calculated 25%
Profile qosp6      : Priority6    bandwidth requested 15% calculated 15%
Profile qosp5      : Priority5    bandwidth requested 12% calculated 12%
Profile qosp4      : Priority4    bandwidth requested 12% calculated 12%
Profile qosp3      : Priority3    bandwidth requested 10% calculated 10%
Profile qosp2      : Priority2    bandwidth requested 10% calculated 10%
Profile qosp1      : Priority1    bandwidth requested 10% calculated 10%
Profile qosp0      : Priority0    bandwidth requested 6%  calculated 6%
```

Syntax: [no] qos profile <queue> <percentage> <queue> <percentage> <queue> <percentage>
 <queue> <percentage> <queue> <percentage> <queue> <percentage> <queue>
 <percentage> <queue> <percentage>

Each <queue> variable specifies the name of a queue. You can specify the queues in any order on the command line, but you must specify each queue.

The <percentage> variable specifies a number for the percentage of the device outbound bandwidth that is allocated to the queue. Brocade QoS queues require a minimum bandwidth percentage of 3 percent for each priority. When jumbo frames are enabled, the minimum bandwidth requirement is 8 percent. If these minimum values are not met, QoS may not be accurate.

Configuration notes for changing the bandwidth

- The total of the percentages you enter must equal 100.
- FastIron devices do not adjust the bandwidth percentages you enter. BigIron QoS does adjust the bandwidth percentages to ensure that each queue has at least its required minimum bandwidth percentage.

On FWS, FCX, and ICX devices, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from the FastIron X Series devices, which support seven priorities for user data instead of eight when sFlow is enabled. QoS queue 1 is reserved for sFlow and not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0. Note that the ICX 6430 does not support sFlow.

Bandwidth allocations of the hybrid WRR and SP queues

NOTE

On the SX-FI48GPP interface module, the bandwidth percentages for 8 to 4 queue mapping for hybrid WRR and SP queues are different from other Brocade SX modules. For more information on 8 to 4 queue mapping on the SX-FI48GPP interface module, refer to [“Default scheduling configuration for the SX-FI48GPP module”](#) on page 1985.

To change the default bandwidth percentages for the queues when the device is configured to use the combined SP and WRR queuing mechanism, enter commands such as the following. Note that this example uses the default queue names.

```
Brocade(config)#qos profile qosp7 sp qosp6 sp qosp5 20 qosp4 16 qosp3 16 qosp2 16
qosp1 16 qosp0 16
```

Syntax: [no] qos profile <queue 7> sp <queue 6> sp | <percentage> <queue 5> <percentage> <queue 4> <percentage> <queue 3> <percentage> <queue 2> <percentage> <queue 1> <percentage> <queue 0> <percentage>

Each <queue> specifies the name of a queue, such as 7, 6, 5, 4, 3, 2, 1, and 0. You can specify the queues in any order on the command line, but you must specify each queue. Note that queue 7 supports Strict Priority (sp) only, queue 6 supports both SP and WRR queuing mechanisms (sp |), and queues 0 through 5 support the WRR queuing mechanism only.

NOTE

Brocade stackable devices that are operating as members of a stack reserve queue 7 for stacking functions.

The **sp** parameter configures strict priority as the queuing mechanism. Note that only queue 7 and queue 6 support this method.

The <percentage> variable configures WRR as the queuing mechanism and specifies the percentage of the device outbound bandwidth allocated to the queue. The queues require a minimum bandwidth percentage of 3 percent for each priority. When jumbo frames are enabled, the minimum bandwidth requirement is 8 percent. If these minimum values are not met, QoS may not be accurate.

NOTE

The percentages must add up to 100. The Brocade FastIron devices do not adjust the bandwidth percentages you enter. In contrast, the BigIron QoS does adjust the bandwidth percentages to ensure that each queue has at least its required minimum bandwidth percentage.

Viewing QoS settings

To display the QoS settings for all of the queues, enter the **show qos-profiles** command.

The following example shows the output on an FESX device.

```
Brocade#show qos-profiles all
bandwidth scheduling mechanism: weighted priority
Profile qosp7      : Priority7   bandwidth requested 25% calculated 25%
Profile qosp6      : Priority6   bandwidth requested 15% calculated 15%
Profile qosp5      : Priority5   bandwidth requested 12% calculated 12%
Profile qosp4      : Priority4   bandwidth requested 12% calculated 12%
Profile qosp3      : Priority3   bandwidth requested 10% calculated 10%
Profile qosp2      : Priority2   bandwidth requested 10% calculated 10%
Profile qosp1      : Priority1   bandwidth requested 10% calculated 10%
Profile qosp0      : Priority0   bandwidth requested 6%  calculated 6%
```

Syntax: **show qos-profiles all** | **<name>**

The **all** parameter displays the settings for all eight queues.

The **<name>** variable displays the settings for the specified queue.

Viewing DSCP-based QoS settings

To display configuration information for DSCP-based QoS, enter the **show qos-tos** command at any level of the CLI.

```
Brocade#show qos-tos
DSCP-->Traffic-Class map: (DSCP = d1d2: 00, 01...63)
      d2|  0   1   2   3   4   5   6   7   8   9
d1  ----+-----
0   |  0   0   0   0   0   0   0   0   1   1
1   |  1   1   1   1   1   1   2   2   2   2
2   |  2   2   2   2   3   3   3   3   3   3
3   |  3   3   4   4   4   4   4   4   4   4
4   |  5   5   5   5   5   5   5   5   6   6
5   |  6   6   6   6   6   6   7   7   7   7
6   |  7   7   7   7   7   7   7   7   7   7

Traffic-Class-->802.1p-Priority map (use to derive DSCP--802.1p-Priority):
Traffic | 802.1p
Class   | Priority
-----+-----
0       | 0
1       | 1
2       | 2
3       | 3
4       | 4
5       | 5
6       | 6
7       | 7
-----+-----
```

Syntax: `show qos-tos`

[Table 348](#) shows the output information for the `show qos-tos` command.

TABLE 348 DSCP-based QoS configuration information

Field	Description
DSCP to traffic class map	
d1 and d2	The DSCP to forwarding priority mappings that are currently in effect. NOTE: The example shows the default mappings. If you change the mappings, the command displays the changed mappings
Traffic class to 802.1p priority map	
Traffic Class and 802.1p Priority	The traffic class to 802.1p priority mappings that are currently in effect. NOTE: The example shows the default mappings. If you change the mappings, the command displays the changed mappings.

The **show qos-tos** command can also be used to display configuration information for 8 to 4 queue mapping. The following example displays an 8 to 4 queue mapping configuration.

```
Brocade#show qos-tos
DSCP-->Traffic-Class map: (DSCP = d1d2: 00, 01...63)
```

d2	0	1	2	3	4	5	6	7	8	9
d1										
0	0	0	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	2	3	3	3	3	3	3
3	3	3	4	4	4	4	4	4	4	4
4	5	5	5	5	5	5	5	5	6	6
5	6	6	6	6	6	6	7	7	7	7
6	7	7	7	7						

```
Traffic-Class-->802.1p-Priority map (use to derive DSCP--802.1p-Priority):
```

Traffic Class	802.1p Priority
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

```
8to4 queue mapping:
```

Priority	Hardware Queue
0	0
1	0
2	1
3	1
4	2
5	2
6	3
7	3

[Table 349](#) shows the output information for 8 to 4 queue mapping for the **show qos-tos** command.

TABLE 349 8 to 4 queue mapping configuration information

Field	Description
8 to 4 queue mapping	
Priority and Hardware Queue	The priority to hardware queues that are currently in effect for 8 to 4 queue mapping on the SX-FI48GPP interface module. QoS priority 7 is the highest priority, and QoS 0 is the lowest priority

Syslog messages

[Table 1](#) lists all of the Syslog messages. Note that some of the messages apply only to Layer 3 switches.

NOTE

This chapter does not list Syslog messages that can be displayed when a debug option is enabled.

The messages are listed by message level, in the following order, then by message type:

- Emergencies (none)
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

TABLE 1 Brocade Syslog messages

Message level	Message	Explanation
Alert	<num-modules> modules and 1 power supply, need more power supply!!	Indicates that the chassis needs more power supplies to run the modules in the chassis. The <num-modules> parameter indicates the number of modules in the chassis.
Alert	Fan <num>, <location>, failed	A fan has failed. The <num> is the fan number. The <location> describes where the failed fan is in the chassis.
Alert	MAC Authentication failed for <mac-address> on <portnum>	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, the VLAN returned in the RADIUS Access-Accept message did not refer to a valid VLAN or VLAN ID on the Brocade device. This is treated as an authentication failure.
Alert	MAC Authentication failed for <mac-address> on <portnum> (Invalid User)	RADIUS authentication failed for the specified <mac-address> on the specified <portnum> because the MAC address sent to the RADIUS server was not found in the RADIUS server users database.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Alert	MAC Authentication failed for <mac-address> on <portnum> (No VLAN Info received from RADIUS server)	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, dynamic VLAN assignment was enabled for the port, but the RADIUS Access-Accept message did not include VLAN information. This is treated as an authentication failure.
Alert	MAC Authentication failed for <mac-address> on <portnum> (Port is already in another radius given vlan)	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, the RADIUS Access-Accept message specified a VLAN ID, although the port had previously been moved to a different RADIUS-assigned VLAN. This is treated as an authentication failure.
Alert	MAC Authentication failed for <mac-address> on <portnum> (RADIUS given vlan does not exist)	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, the RADIUS Access-Accept message specified a VLAN that does not exist in the Brocade configuration. This is treated as an authentication failure.
Alert	MAC Authentication failed for <mac-address> on <portnum> (RADIUS given VLAN does not match with TAGGED vlan)	Multi-device port authentication failed for the <mac-address> on a tagged port because the packet with this MAC address as the source was tagged with a VLAN ID different from the RADIUS-supplied VLAN ID.
Alert	Management module at slot <slot-num> state changed from <module-state> to <module-state>.	Indicates a state change in a management module. The <slot-num> indicates the chassis slot containing the module. The <module-state> can be one of the following: <ul style="list-style-type: none"> • active • standby • crashed • coming-up • unknown
Alert	OSPF LSA Overflow, LSA Type = <lsa-type>	Indicates an LSA database overflow. The <lsa-type> parameter indicates the type of LSA that experienced the overflow condition. The LSA type is one of the following: <ul style="list-style-type: none"> • 1 – Router • 2 – Network • 3 – Summary • 4 – Summary • 5 – External
Alert	OSPF Memory Overflow	OSPF has run out of memory.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Alert	Power supply <num>, <location>, failed	A power supply has failed. The <num> is the power supply number. The <location> describes where the failed power supply is in the chassis.
Alert	System: Module in slot <slot-num> encountered PCI config read error: Bus <PCI-bus-number>, Dev <PCI-device-number>, Reg Offset <PCI-config-register-offset>.	The module encountered a hardware configuration read error.
Alert	System: Module in slot <slot-num> encountered PCI config write error: Bus <PCI-bus-number>, Dev <PCI-device-number>, Reg Offset <PCI-config-register-offset>.	The module encountered a hardware configuration write error.
Alert	System: Module in slot <slot-num> encountered PCI memory read error: Mem Addr <memory-address>	The module encountered a hardware memory read error. The <memory-address> is in hexadecimal format.
Alert	System: Module in slot <slot-num> encountered PCI memory write error: Mem Addr <memory-address>.	The module encountered a hardware memory write error. The <memory-address> is in hexadecimal format.
Alert	System: Module in slot <slot-num> encountered unrecoverable PCI bridge validation failure. Module will be deleted.	The module encountered an unrecoverable (hardware) bridge validation failure. The module will be disabled or powered down.
Alert	System: Module in slot <slot-num> encountered unrecoverable PCI config read failure. Module will be deleted.	The module encountered an unrecoverable hardware configuration read failure. The module will be disabled or powered down.
Alert	System: Module in slot <slot-num> encountered unrecoverable PCI config write failure. Module will be deleted.	The module encountered an unrecoverable hardware configuration write failure. The module will be disabled or powered down.
Alert	System: Module in slot <slot-num> encountered unrecoverable PCI device validation failure. Module will be deleted.	The module encountered an unrecoverable (hardware) device validation failure. The module will be disabled or powered down.
Alert	System: Module in slot <slot-num> encountered unrecoverable PCI memory read failure. Module will be deleted.	The module encountered an unrecoverable hardware memory read failure. The module will be disabled or powered down.
Alert	System: Module in slot <slot-num> encountered unrecoverable PCI memory write failure. Module will be deleted.	The module encountered an unrecoverable hardware memory write failure. The module will be disabled or powered down.
Alert	System: No Free Tcam Entry available. System will be unstable	In FWS devices, the limit for the TCAM routing entries has been reached. You must reboot the device.
Alert	System: Temperature is over shutdown level, system is going to be reset in <num> seconds	The chassis temperature has risen above shutdown level. The system will be shut down in the amount of time indicated.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Alert	Temperature <degrees> C degrees, warning level <warn-degrees> C degrees, shutdown level <shutdown-degrees> C degrees	Indicates an over temperature condition on the active module. The <degrees> value indicates the temperature of the module. The <warn-degrees> value is the warning threshold temperature configured for the module. The <shutdown-degrees> value is the shutdown temperature configured for the module.
Critical	Authentication shut down <portnum> due to DOS attack	Denial of Service (DoS) attack protection was enabled for multi-device port authentication on the specified <portnum>, and the per-second rate of RADIUS authentication attempts for the port exceeded the configured limit. The Brocade device considers this to be a DoS attack and disables the port.
Debug	BGP4: Not enough memory available to run BGP4	The device could not start the BGP4 routing protocol because there is not enough memory available.
Debug	DOT1X: Not enough memory	There is not enough system memory for 802.1X authentication to take place. Contact Brocade Technical Support.
Error	No of prefixes received from BGP peer <ip-addr> exceeds maximum prefix-limit...shutdown	The Layer 3 switch has received more than the specified maximum number of prefixes from the neighbor, and the Layer 3 switch is therefore shutting down its BGP4 session with the neighbor.
Informational	IPv6: IPv6 protocol disabled on the device from <session-id>	IPv6 protocol was disabled on the device during the specified session.
Informational	IPv6: IPv6 protocol enabled on the device from <session-id>	IPv6 protocol was enabled on the device during the specified session.
Informational	MAC Filter applied to port <port-id> by <username> from <session-id> (filter id=<filter-ids>)	Indicates a MAC address filter was applied to the specified port by the specified user during the specified session. <session-id> can be console, telnet, ssh, web, or snmp. <filter-ids> is a list of the MAC address filters that were applied.
Informational	MAC Filter removed from port <port-id> by <username> from <session-id> (filter id=<filter-ids>)	Indicates a MAC address filter was removed from the specified port by the specified user during the specified session. <session-id> can be console, telnet, ssh, web, or snmp. <filter-ids> is a list of the MAC address filters that were removed.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	Security: Password has been changed for user <username> from <session-id>	Password of the specified user has been changed during the specified session ID or type. <session-id> can be console, telnet, ssh, web, or snmp.
Informational	<device-name> : Logical link on interface ethernet <slot#/port#> is down.	The specified ports were logically brought down while singleton was configured on the port.
Informational	<device-name>: Logical link on interface ethernet <slot#/port#> is up.	The specified ports were logically brought up while singleton was configured on the port.
Informational	<user-name> login to PRIVILEGED mode	A user has logged into the Privileged EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> login to USER EXEC mode	A user has logged into the USER EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> logout from PRIVILEGED mode	A user has logged out of Privileged EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> logout from USER EXEC mode	A user has logged out of the USER EXEC mode of the CLI. The <user-name> is the user name.
Informational	ACL <ACL id> added deleted modified from console telnet ssh web snmp session	A user created, modified, deleted, or applied an ACL through a Web, SNMP, console, SSH, or Telnet session.
Informational	Bridge is new root, vlan <vlan-id>, root ID <root-id>	A Spanning Tree Protocol (STP) topology change has occurred, resulting in the Brocade device becoming the root bridge. The <vlan-id> is the ID of the VLAN in which the STP topology change occurred. The <root-id> is the STP bridge root ID.
Informational	Bridge root changed, vlan <vlan-id>, new root ID <string>, root interface <portnum>	A Spanning Tree Protocol (STP) topology change has occurred. The <vlan-id> is the ID of the VLAN in which the STP topology change occurred. The <root-id> is the STP bridge root ID. The <portnum> is the number of the port connected to the new root bridge.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	Bridge topology change, vlan <vlan-id>, interface <portnum>, changed state to <stp-state>	<p>A Spanning Tree Protocol (STP) topology change has occurred on a port.</p> <p>The <vlan-id> is the ID of the VLAN in which the STP topology change occurred.</p> <p>The <portnum> is the port number.</p> <p>The <stp-state> is the new STP state and can be one of the following:</p> <ul style="list-style-type: none"> • disabled • blocking • listening • learning • forwarding • unknown
Informational	Cold start	The device has been powered on.
Informational	DHCP : snooping on untrusted port <portnum>, type <number>, drop	The device has indicated that the DHCP client receives DHCP server reply packets on untrusted ports, and packets are dropped.
Informational	DOT1X : port <portnum> - mac <mac address> Cannot apply an ACL or MAC filter on a port member of a VE (virtual interface)	The RADIUS server returned an IP ACL or MAC address filter, but the port is a member of a virtual interface (VE).
Informational	DOT1X : port <portnum> - mac <mac address> cannot remove inbound ACL	An error occurred while removing the inbound ACL.
Informational	DOT1X : port <portnum> - mac <mac address> Downloading a MAC filter, but MAC filter have no effect on router port	The RADIUS server returned an MAC address filter, but the <portnum> is a router port (it has one or more IP addresses).
Informational	DOT1X : port <portnum> - mac <mac address> Downloading an IP ACL, but IP ACL have no effect on a switch port	The RADIUS server returned an IP ACL, but the <portnum> is a switch port (no IP address).
Informational	DOT1X : port <portnum> - mac <mac address> Error - could not add all MAC filters	The Brocade device was unable to implement the MAC address filters returned by the RADIUS server.
Informational	DOT1X : port <portnum> - mac <mac address> Invalid MAC filter ID - this ID doesn't exist	The MAC address filter ID returned by the RADIUS server does not exist in the Brocade configuration.
Informational	DOT1X : port <portnum> - mac <mac address> Invalid MAC filter ID - this ID is user defined and cannot be used	The port was assigned a MAC address filter ID that had been dynamically created by another user.
Informational	DOT1X : port <portnum> - mac <mac address> is unauthorized because system resource is not enough or the invalid information to set the dynamic assigned IP ACLs or MAC address filters	802.1X authentication failed for the Client with the specified <mac address> on the specified <portnum> either due to insufficient system resources on the device, or due to invalid IP ACL or MAC address filter information returned by the RADIUS server.
Informational	DOT1X : port <portnum> - mac <mac address> Port is already bound with MAC filter	The RADIUS server returned a MAC address filter, but a MAC address filter had already been applied to the port.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	DOT1X : port <portnum> - mac <mac address> This device doesn't support ACL with MAC Filtering on the same port	The RADIUS server returned a MAC address filter while an IP ACL was applied to the port, or returned an IP ACL while a MAC address filter was applied to the port.
Informational	DOT1X Port <portnum> is unauthorized because system resource is not enough or the invalid information to set the dynamic assigned IP ACLs or MAC address filters	802.1X authentication could not take place on the port. This happened because strict security mode was enabled and one of the following occurred: <ul style="list-style-type: none"> Insufficient system resources were available on the device to apply an IP ACL or MAC address filter to the port Invalid information was received from the RADIUS server (for example, the Filter-ID attribute did not refer to an existing IP ACL or MAC address filter)
Informational	DOT1X: Port <portnum> currently used vlan-id changes to <vlan-id> due to dot1x-RADIUS vlan assignment	A user has completed 802.1X authentication. The profile received from the RADIUS server specifies a VLAN ID for the user. The port to which the user is connected has been moved to the VLAN indicated by <vlan-id>.
Informational	DOT1X: Port <portnum> currently used vlan-id is set back to port default vlan-id <vlan-id>	The user connected to <portnum> has disconnected, causing the port to be moved back into its default VLAN, <vlan-id>.
Informational	DOT1X: Port <portnum>, AuthControlledPortStatus change: authorized	The status of the interface controlled port has changed from unauthorized to authorized.
Informational	DOT1X: Port <portnum>, AuthControlledPortStatus change: unauthorized	The status of the interface controlled port has changed from authorized to unauthorized.
Informational	Enable super port-config read-only password deleted added modified from console telnet ssh web snmp OR Line password deleted added modified from console telnet ssh web snmp	A user created, re-configured, or deleted an Enable or Line password through the Web, SNMP, console, SSH, or Telnet session.
Informational	ERR_DISABLE: Interface ethernet <port-number>, err-disable recovery timeout	Errdisable recovery timer expired and the port has been reenabled.
Informational	ERR_DISABLE: Interface ethernet 16, err-disable recovery timeout	If the wait time (port is down and is waiting to come up) expires and the port is brought up the following message is displayed.
Informational	ERR_DISABLE: Link flaps on port ethernet 16 exceeded threshold; port in err-disable state	The threshold for the number of times that a port link toggles from "up" to "down" and "down" to "up" has been exceeded.
Informational	Interface <portnum>, line protocol down	The line protocol on a port has gone down. The <portnum> is the port number.
Informational	Interface <portnum>, line protocol up	The line protocol on a port has come up. The <portnum> is the port number.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	Interface <portnum>, state down	A port has gone down. The <portnum> is the port number.
Informational	Interface <portnum>, state up	A port has come up. The <portnum> is the port number.
Informational	MAC Based Vlan Disabled on port <port id>	A MAC Based VLAN has been disabled on a port
Informational	MAC Based Vlan Enabled on port <port id>	A MAC Based VLAN has been enabled on a port.
Informational	MAC Filter added deleted modified from console telnet ssh web snmp session filter id = <MAC filter ID>, src mac = <Source MAC address> any, dst mac = <Destination MAC address> any	A user created, modified, deleted, or applied this MAC address filter through the Web, SNMP, console, SSH, or Telnet session.
Informational	MSTP: BPDU-guard interface ethernet <port-number> detect (Received BPDU), putting into err-disable state.	BPDU guard violation occurred in MSTP.
Informational	OPTICAL MONITORING: port <port-number> is not capable.	The optical transceiver is qualified by Brocade, but the transceiver does not support digital optical performance monitoring.
Informational	Port <p> priority changed to <n>	A port priority has changed.
Informational	Port <portnum>, srcip-security max-ipaddr-per-int reached.Last IP=<ipaddr>	The address limit specified by the srcip-security max-ipaddr-per-interface command has been reached for the port.
Informational	Port <portnum>, srcip-security max-ipaddr-per-int reached.Last IP=<ipaddr>	The address limit specified by the srcip-security max-ipaddr-per-interface command has been reached for the port.
Informational	Security: console login by <username> to USER PRIVILEGE EXEC mode	The specified user logged into the device console into the specified EXEC mode.
Informational	Security: console logout by <username>	The specified user logged out of the device console.
Informational	Security: telnet SSH login by <username> from src IP <ip-address>, src MAC <mac-address> to USER PRIVILEGE EXEC mode	The specified user logged into the device using Telnet or SSH from either or both the specified IP address and MAC address. The user logged into the specified EXEC mode.
Informational	Security: telnet SSH logout by <username> from src IP <ip-address>, src MAC <mac-address> to USER PRIVILEGE EXEC mode	The specified user logged out of the device. The user was using Telnet or SSH to access the device from either or both the specified IP address and MAC address. The user logged out of the specified EXEC mode.
Informational	SNMP read-only community read-write community contact location user group view engineld trap [host] [<value-str>] deleted added modified from console telnet ssh web snmp session	A user made SNMP configuration changes through the Web, SNMP, console, SSH, or Telnet session. [<value-str>] does not appear in the message if SNMP community or engineld is specified.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	SNMP Auth. failure, intruder IP: <ip-addr>	A user has tried to open a management session with the device using an invalid SNMP community string. The <ip-addr> is the IP address of the host that sent the invalid community string.
Informational	SSH telnet server enabled disabled from console telnet ssh web snmp session [by user <username>]	A user enabled or disabled an SSH or Telnet session, or changed the SSH enable/disable configuration through the Web, SNMP, console, SSH, or Telnet session.
Informational	startup-config was changed or startup-config was changed by <user-name>	A configuration change was saved to the startup-config file. The <user-name> is the user ID, if they entered a user ID to log in.
Informational	STP: Root Guard Port <port-number>, VLAN <vlan-ID> consistent (Timeout).	Root guard unblocks a port.
Informational	STP: Root Guard Port <port-number>, VLAN <vlan-ID> inconsistent (Received superior BPDU).	Root guard blocked a port.
Informational	STP: VLAN <vlan id> BPDU-Guard on Port <port id> triggered (Received BPDU), putting into err-disable state	The BPDU guard feature has detected an incoming BPDU on {vlan-id, port-id}
Informational	STP: VLAN <vlan id> Root-Protect Port <port id>, Consistent (Timeout)	The root protect feature goes back to the consistent state.
Informational	STP: VLAN <vlan id> Root-Protect Port <port id>, Inconsistent (Received superior BPDU)	The root protect feature has detected a superior BPDU and goes into the inconsistent state on {vlan-id, port-id}.
Informational	STP: VLAN <vlan-id> BPDU-guard port <port-number> detect (Received BPDU), putting into err-disable state	STP placed a port into an errdisable state for BPDU guard.
Informational	STP: VLAN 1 BPDU-guard port <port-number> detect (Received BPDU), putting into err-disable state.	BPDU guard violation in occurred in STP or RSTP.
Informational	Syslog server <IP-address> deleted added modified from console telnet ssh web snmp OR Syslog operation enabled disabled from console telnet ssh web snmp	A user made Syslog configuration changes to the specified Syslog server address, or enabled or disabled a Syslog operation through the Web, SNMP, console, SSH, or Telnet session.
Informational	SYSTEM: Optic is not Brocade-qualified (<port-number>)	Brocade does not support the optical transceiver.
Informational	System: Fan <fan id> (from left when facing right side), ok	The fan status has changed from fail to normal.
Informational	System: Fan speed changed automatically to <fan speed>	The system automatically changed the fan speed to the speed specified in this message.
Informational	System: No free TCAM entry. System will be unstable	There are no TCAM entries available.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	System: Static Mac entry with Mac Address <mac-address> is added from the <unit>/<slot>/<port> to <unit>/<slot>/<port> on VLANs <vlan-id> to <vlan-id>	A MAC address is added to a range of interfaces, which are members of the specified VLAN range.
Informational	System: Static Mac entry with Mac Address <mac-address> is added to ethe <unit>/<slot>/<port> to <unit>/<slot>/<port> on <vlan-id>	A MAC address is added to a range of interfaces, which are members of the specified VLAN.
Informational	System: Static Mac entry with Mac Address <mac-address> is added to portnumber <unit>/<slot>/<port> on VLAN <vlan-id>	A MAC address is added to an interface and the interface is a member of the specified VLAN.
Informational	System: Static Mac entry with Mac Address <mac-address> is deleted from ethe <unit>/<slot>/<port> to <unit>/<slot>/<port> on <vlan-id>	A MAC address is deleted from a range of interfaces, which are members of the specified VLAN.
Informational	System: Static Mac entry with Mac Address <mac-address> is deleted from ethe <unit>/<slot>/<port> to <unit>/<slot>/<port> on VLANs <vlan-id> to <vlan-id>	A MAC address is deleted from a range of interfaces, which are members of the specified VLAN range.
Informational	System: Static Mac entry with Mac Address <mac-address> is deleted from portnumber <unit>/<slot>/<port> on <vlan-id>	A MAC address is deleted from an interface and the interface is a member of the specified VLAN.
Informational	System: Static Mac entry with Mac Address <mac-address> is deleted from portnumber <unit>/<slot>/<port> on VLANs <vlan-id> to <vlan-id>	A MAC address is deleted from an interface and the interface is a member of the specified VLAN range.
Informational	telnet SSH web access [by <username>] from src IP <source ip address>, src MAC <source MAC address> rejected, <n> attempts	There were failed web, SSH, or Telnet login access attempts from the specified source IP and MAC address. <ul style="list-style-type: none"> [by <user> <username>] does not appear if telnet or SSH clients are specified. <n> is the number of times this SNMP trap occurred in the last five minutes, or other configured number of minutes.
Informational	Trunk group (<ports>) created by 802.3ad link-aggregation module.	802.3ad link aggregation is configured on the device, and the feature has dynamically created a trunk group (aggregate link). The <ports> is a list of the ports that were aggregated to make the trunk group.
Informational	user <username> added deleted modified from console telnet ssh web snmp	A user created, modified, or deleted a local user account through the Web, SNMP, console, SSH, or Telnet session.
Informational	vlan <vlan id> added deleted modified from console telnet ssh web snmp session	A user created, modified, or deleted a VLAN through the Web, SNMP, console, SSH, or Telnet session.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	Warm start	The system software (flash code) has been reloaded.
Informational	Stack: Stack unit <unit#> has been deleted to the stack system	The specified unit has been deleted from the stacking system.
Informational	Stack unit <unitNumber> has been elected as ACTIVE unit of the stack system	The specified unit in a stack has been elected as the Master unit for the stacking system.
Informational	Stack: Stack unit <unit#> has been added to the stack system	The specified unit has been added to the stacking system.
Informational	System: Management MAC address changed to <mac_address>	The management MAC address of a stacking system has been changed
Informational	System: Stack unit <unit#> Fan <fan#> (<description>), failed	The operational status of a fan in the specified unit in a stack changed from normal to failure.
Informational	System: Stack unit <unit#> Power supply <power-supply#> is down	The operational status of a power supply of the specified unit in a stack changed from normal to failure.
Informational	System: Stack unit <unit#> Power supply <power-supply#> is up	The operational status of a power supply of the specified unit in a stack changed from failure to normal.
Informational	System: Stack unit <unit#> Fan <fan#> (<description>), ok	The operational status of a fan in the specified unit in a stack changed from failure to normal.
Informational	System: Stack unit <unitNumber> Temperature <actual-temp> C degrees, warning level <warning-temp> C degrees, shutdown level <shutdown-temp> C degrees	The actual temperature reading for a unit in a stack is above the warning temperature threshold.
Informational	vlan <vlan-id> Bridge is RootBridge <mac-address> (MgmtPriChg)	802.1W changed the current bridge to be the root bridge of the given topology due to administrative change in bridge priority.
Informational	vlan <vlan-id> Bridge is RootBridge <mac-address> (MsgAgeExpiry)	The message age expired on the Root port so 802.1W changed the current bridge to be the root bridge of the topology.
Informational	vlan <vlan-id> interface <portnum> Bridge TC Event (DOT1wTransition)	802.1W recognized a topology change event in the bridge. The topology change event is the forwarding action that started on a non-edge Designated port or Root port.
Informational	vlan <vlan-id> interface <portnum> STP state -> <state> (DOT1wTransition)	802.1W changed the state of a port to a new state: forwarding, learning, blocking. If the port changes to blocking, the bridge port is in discarding state.
Informational	vlan <vlan-id> New RootBridge <mac-address> RootPort <portnum> (BpduRcvd)	802.1W selected a new root bridge as a result of the BPDUs received on a bridge port.
Informational	vlan <vlan-id> New RootPort <portnum> (RootSelection)	802.1W changed the port role to Root port, using the root selection computation.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	ACL exceed max DMA L4 cam resource, using flow based ACL instead	The port does not have enough Layer 4 CAM entries for the ACL. To correct this condition, allocate more Layer 4 CAM entries. To allocate more Layer 4 CAM entries, enter the following command at the CLI configuration level for the interface: ip access-group max-l4-cam <num>
Notification	ACL insufficient L4 cam resource, using flow based ACL instead	The port does not have a large enough CAM partition for the ACLs
Notification	ACL insufficient L4 session resource, using flow based ACL instead	The device does not have enough Layer 4 session entries. To correct this condition, allocate more memory for sessions. To allocate more memory, enter the following command at the global CONFIG level of the CLI interface: system-max session-limit <num>
Notification	ACL port fragment packet inspect rate <rate> exceeded on port <portnum>	The fragment rate allowed on an individual interface has been exceeded. The <rate> indicates the maximum rate allowed. The <portnum> indicates the port. This message can occur if fragment throttling is enabled.
Notification	ACL system fragment packet inspect rate <rate> exceeded	The fragment rate allowed on the device has been exceeded. The <rate> indicates the maximum rate allowed. This message can occur if fragment throttling is enabled.
Notification	Authentication Disabled on <portnum>	The multi-device port authentication feature was disabled on the on the specified <portnum>.
Notification	Authentication Enabled on <portnum>	The multi-device port authentication feature was enabled on the on the specified <portnum>.
Notification	BGP Peer <ip-addr> DOWN (IDLE)	Indicates that a BGP4 neighbor has gone down. The <ip-addr> is the IP address of the neighbor BGP4 interface with the Brocade device.
Notification	BGP Peer <ip-addr> UP (ESTABLISHED)	Indicates that a BGP4 neighbor has come up. The <ip-addr> is the IP address of the neighbor BGP4 interface with the Brocade device.
Notification	DHCP : snooping on untrusted port <portnum>, type <number>, drop	Indicates that the DHCP client receives DHCP server reply packets on untrusted ports, and packets are dropped.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	DOT1X issues software but not physical port down indication of Port <portnum> to other software applications	The device has indicated that the specified is no longer authorized, but the actual port may still be active.
Notification	DOT1X issues software but not physical port up indication of Port <portnum> to other software applications	The device has indicated that the specified port has been authenticated, but the actual port may not be active.
Notification	DOT1X: Port <port_id> Mac <mac_address> -user <user_id> - RADIUS timeout for authentication	The RADIUS session has timed out for this 802.1x port.
Notification	ISIS L1 ADJACENCY DOWN <system-id> on circuit <circuit-id>	The Layer 3 switch adjacency with this Level-1 IS has gone down. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.
Notification	ISIS L1 ADJACENCY UP <system-id> on circuit <circuit-id>	The Layer 3 switch adjacency with this Level-1 IS has come up. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.
Notification	ISIS L2 ADJACENCY DOWN <system-id> on circuit <circuit-id>	The Layer 3 switch adjacency with this Level-2 IS has gone down. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.
Notification	ISIS L2 ADJACENCY UP <system-id> on circuit <circuit-id>	The Layer 3 switch adjacency with this Level-2 IS has come up. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.
Notification	Local ICMP exceeds <burst-max> burst packets, stopping for <lockup> seconds!!	The number of ICMP packets exceeds the <burst-max> threshold set by the ip icmp burst command. The Brocade device may be the victim of a Denial of Service (DoS) attack. All ICMP packets will be dropped for the number of seconds specified by the <lockup> value. When the lockup period expires, the packet counter is reset and measurement is restarted.
Notification	Local TCP exceeds <burst-max> burst packets, stopping for <lockup> seconds!!	The number of TCP SYN packets exceeds the <burst-max> threshold set by the ip tcp burst command. The Brocade device may be the victim of a TCP SYN DoS attack. All TCP SYN packets will be dropped for the number of seconds specified by the <lockup> value. When the lockup period expires, the packet counter is reset and measurement is restarted.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	Local TCP exceeds <num> burst packets, stopping for <num> seconds!!	<p>Threshold parameters for local TCP traffic on the device have been configured, and the maximum burst size for TCP packets has been exceeded.</p> <p>The first <num> is the maximum burst size (maximum number of packets allowed). The second <num> is the number of seconds during which additional TCP packets will be blocked on the device.</p> <p>NOTE: This message can occur in response to an attempted TCP SYN attack.</p>
Notification	MAC Authentication RADIUS timeout for <mac_address> on port <port_id>	The RADIUS session has timed out for the MAC address for this port.
Notification	MAC Authentication succeeded for <mac-address> on <portnum>	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>.
Notification	Module was inserted to slot <slot-num>	<p>Indicates that a module was inserted into a chassis slot.</p> <p>The <slot-num> is the number of the chassis slot into which the module was inserted.</p>
Notification	Module was removed from slot <slot-num>	<p>Indicates that a module was removed from a chassis slot.</p> <p>The <slot-num> is the number of the chassis slot from which the module was removed.</p>
Notification	OSPF interface state changed, rid <router-id>, intf addr <ip-addr>, state <ospf-state>	<p>Indicates that the state of an OSPF interface has changed.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the interface IP address.</p> <p>The <ospf-state> indicates the state to which the interface has changed and can be one of the following:</p> <ul style="list-style-type: none"> • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF intf authen failure, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	<p>Indicates that an OSPF interface authentication failure has occurred. The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <src-ip-addr> is the IP address of the interface from which the Brocade device received the authentication failure.</p> <p>The <error-type> can be one of the following:</p> <ul style="list-style-type: none"> • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF intf config error, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	<p>Indicates that an OSPF interface configuration error has occurred.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <src-ip-addr> is the IP address of the interface from which the Brocade device received the error packet.</p> <p>The <error-type> can be one of the following:</p> <ul style="list-style-type: none"> • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown
Notification	OSPF intf rcvd bad pkt, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, pkt type <pkt-type>	<p>Indicates that an OSPF interface received a bad packet.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <src-ip-addr> is the IP address of the interface from which the Brocade device received the authentication failure.</p> <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF intf rcvd bad pkt: Bad Checksum, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	<p>The device received an OSPF packet that had an invalid checksum.</p> <p>The rid <ip-addr> is the Brocade router ID.</p> <p>The intf addr <ip-addr> is the IP address of the Brocade interface that received the packet.</p> <p>The pkt size <num> is the number of bytes in the packet.</p> <p>The checksum <num> is the checksum value for the packet.</p> <p>The pkt src addr <ip-addr> is the IP address of the neighbor that sent the packet.</p> <p>The pkt type <type> is the OSPF packet type and can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state acknowledgement • unknown (indicates an invalid packet type)
Notification	OSPF intf rcvd bad pkt: Bad Packet type, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	<p>The device received an OSPF packet with an invalid type.</p> <p>The parameters are the same as for the Bad Checksum message. The pkt type <type> value is "unknown", indicating that the packet type is invalid.</p>
Notification	OSPF intf rcvd bad pkt: Invalid packet size, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	<p>The device received an OSPF packet with an invalid packet size.</p> <p>The parameters are the same as for the Bad Checksum message.</p>
Notification	OSPF intf rcvd bad pkt: Unable to find associated neighbor, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	<p>The neighbor IP address in the packet is not in the list of OSPF neighbors in the Brocade device.</p> <p>The parameters are the same as for the Bad Checksum message.</p>

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF intf retransmit, rid <router-id>, intf addr <ip-addr>, nbr rid <nbr-router-id>, pkt type is <pkt-type>, LSA type <lsa-type>, LSA id <lsa-id>, LSA rid <lsa-router-id>	<p>An OSPF interface on the Brocade device has retransmitted a Link State Advertisement (LSA).</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <nbr-router-id> is the router ID of the neighbor router.</p> <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown <p>The <lsa-type> is the type of LSA.</p> <p>The <lsa-id> is the LSA ID.</p> <p>The <lsa-router-id> is the LSA router ID.</p>
Notification	OSPF LSDB approaching overflow, rid <router-id>, limit <num>	<p>The software is close to an LSDB condition.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <num> is the number of LSAs.</p>
Notification	OSPF LSDB overflow, rid <router-id>, limit <num>	<p>A Link State Database Overflow (LSDB) condition has occurred.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <num> is the number of LSAs.</p>
Notification	OSPF max age LSA, rid <router-id>, area <area-id>, LSA type <lsa-type>, LSA id <lsa-id>, LSA rid <lsa-router-id>	<p>An LSA has reached its maximum age.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <area-id> is the OSPF area.</p> <p>The <lsa-type> is the type of LSA.</p> <p>The <lsa-id> is the LSA ID.</p> <p>The <lsa-router-id> is the LSA router ID.</p>

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF nbr state changed, rid <router-id>, nbr addr <ip-addr>, nbr rid <nbr-router-id>, state <ospf-state>	<p>Indicates that the state of an OSPF neighbor has changed.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the neighbor.</p> <p>The <nbr-router-id> is the router ID of the neighbor.</p> <p>The <ospf-state> indicates the state to which the interface has changed and can be one of the following:</p> <ul style="list-style-type: none"> • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown
Notification	OSPF originate LSA, rid <router-id>, area <area-id>, LSA type <lsa-type>, LSA id <lsa-id>, LSA router id <lsa-router-id>	<p>An OSPF interface has originated an LSA.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <area-id> is the OSPF area.</p> <p>The <lsa-type> is the type of LSA.</p> <p>The <lsa-id> is the LSA ID.</p> <p>The <lsa-router-id> is the LSA router ID.</p>

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF virtual intf authen failure, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	<p>Indicates that an OSPF virtual routing interface authentication failure has occurred.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <src-ip-addr> is the IP address of the interface from which the Brocade device received the authentication failure.</p> <p>The <error-type> can be one of the following:</p> <ul style="list-style-type: none"> • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF virtual intf config error, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	<p>Indicates that an OSPF virtual routing interface configuration error has occurred. The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <src-ip-addr> is the IP address of the interface from which the Brocade device received the error packet.</p> <p>The <error-type> can be one of the following:</p> <ul style="list-style-type: none"> • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown
Notification	OSPF virtual intf rcvd bad pkt, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, pkt type <pkt-type>	<p>Indicates that an OSPF interface received a bad packet.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <src-ip-addr> is the IP address of the interface from which the Brocade device received the authentication failure.</p> <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF virtual intf retransmit, rid <router-id>, intf addr <ip-addr>, nbr rid <nbr-router-id>, pkt type is <pkt-type>, LSA type <lsa-type>, LSA id <lsa-id>, LSA rid <lsa-router-id>	<p>An OSPF interface on the Brocade device has retransmitted a Link State Advertisement (LSA).</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the interface on the Brocade device.</p> <p>The <nbr-router-id> is the router ID of the neighbor router.</p> <p>The <packet-type> can be one of the following:</p> <ul style="list-style-type: none"> • hello • database description • link state request • link state update • link state ack • unknown <p>The <lsa-type> is the type of LSA.</p> <p>The <lsa-id> is the LSA ID.</p> <p>The <lsa-router-id> is the LSA router ID.</p>
Notification	OSPF virtual intf state changed, rid <router-id>, area <area-id>, nbr <ip-addr>, state <ospf-state>	<p>Indicates that the state of an OSPF virtual routing interface has changed.</p> <p>The <router-id> is the router ID of the router the interface is on.</p> <p>The <area-id> is the area the interface is in.</p> <p>The <ip-addr> is the IP address of the OSPF neighbor.</p> <p>The <ospf-state> indicates the state to which the interface has changed and can be one of the following:</p> <ul style="list-style-type: none"> • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	OSPF virtual nbr state changed, rid <router-id>, nbr addr <ip-addr>, nbr rid <nbr-router-id>, state <ospf-state>	<p>Indicates that the state of an OSPF virtual neighbor has changed.</p> <p>The <router-id> is the router ID of the Brocade device.</p> <p>The <ip-addr> is the IP address of the neighbor.</p> <p>The <nbr-router-id> is the router ID of the neighbor.</p> <p>The <ospf-state> indicates the state to which the interface has changed and can be one of the following:</p> <ul style="list-style-type: none"> • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown
Notification	Transit ICMP in interface <portnum> exceeds <num> burst packets, stopping for <num> seconds!!	<p>Threshold parameters for ICMP transit (through) traffic have been configured on an interface, and the maximum burst size for ICMP packets on the interface has been exceeded.</p> <p>The <portnum> is the port number.</p> <p>The first <num> is the maximum burst size (maximum number of packets allowed).</p> <p>The second <num> is the number of seconds during which additional ICMP packets will be blocked on the interface.</p> <p>NOTE: This message can occur in response to an attempted Smurf attack.</p>
Notification	Transit TCP in interface <portnum> exceeds <num> burst packets, stopping for <num> seconds!	<p>Threshold parameters for TCP transit (through) traffic have been configured on an interface, and the maximum burst size for TCP packets on the interface has been exceeded.</p> <p>The <portnum> is the port number.</p> <p>The first <num> is the maximum burst size (maximum number of packets allowed).</p> <p>The second <num> is the number of seconds during which additional TCP packets will be blocked on the interface.</p> <p>NOTE: This message can occur in response to an attempted TCP SYN attack.</p>

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Notification	VRRP intf state changed, intf <portnum>, vrid <virtual-router-id>, state <vrrp-state> VRRP (IPv6) intf state changed, intf <portnum>, vrid <virtual-router-id>, state <vrrp-state>	A state change has occurred in a Virtual Router Redundancy Protocol (VRRP) or VRRP-E IPv4 or IPv6 interface. The <portnum> is the port or interface where VRRP or VRRP-E is configured. The <virtual-router-id> is the virtual router ID (VRID) configured on the interface. The <vrrp-state> can be one of the following: <ul style="list-style-type: none"> • init • master • backup • unknown
Warning	DOT1X security violation at port <portnum>, malicious mac address detected: <mac-address>	A security violation was encountered at the specified port number.
Warning	Dup IP <ip-addr> detected, sent from MAC <mac-addr> interface <portnum>	Indicates that the Brocade device received a packet from another device on the network with an IP address that is also configured on the Brocade device. The <ip-addr> is the duplicate IP address. The <mac-addr> is the MAC address of the device with the duplicate IP address. The <portnum> is the Brocade port that received the packet with the duplicate IP address. The address is the packet source IP address.
Warning	IGMP/MLD no hardware vidx, broadcast to the entire vlan. rated limited number	IGMP or MLD snooping has run out of hardware application VLANs. There are 4096 application VLANs per device. Traffic streams for snooping entries without an application VLAN are switched to the entire VLAN and to the CPU to be dropped. This message is rate-limited to appear a maximum of once every 10 minutes. The rate-limited number shows the number on non-printed warnings.
Warning	IGMP/MLD: <vlanId>(<portId>) is V1 but rcvd V2 from nbr <ipAddr>	Port has received a query with a MLD version that does not match the port MLD version. This message is rated-limited to appear a maximum of once every 10 hours.
Warning	Latched low RX Power TX Power TX Bias Current Supply Voltage Temperature warning alarm warning, port <port-number>	The optical transceiver on the given port has risen above or fallen below the alarm or warning threshold.

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Warning	list <ACL-num> denied <ip-proto> <src-ip-addr> (<src-tcp/udp-port>) (Ethernet <portnum> <mac-addr>) -> <dst-ip-addr> (<dst-tcp/udp-port>), 1 event(s)	<p>Indicates that an Access Control List (ACL) denied (dropped) packets.</p> <p>The <ACL-num> indicates the ACL number. Numbers 1 – 99 indicate standard ACLs. Numbers 100 – 199 indicate extended ACLs.</p> <p>The <ip-proto> indicates the IP protocol of the denied packets.</p> <p>The <src-ip-addr> is the source IP address of the denied packets.</p> <p>The <src-tcp/udp-port> is the source TCP or UDP port, if applicable, of the denied packets.</p> <p>The <portnum> indicates the port number on which the packet was denied.</p> <p>The <mac-addr> indicates the source MAC address of the denied packets.</p> <p>The <dst-ip-addr> indicates the destination IP address of the denied packets.</p> <p>The <dst-tcp/udp-port> indicates the destination TCP or UDP port number, if applicable, of the denied packets.</p>
Warning	Locked address violation at interface e<portnum>, address <mac-address>	<p>Indicates that a port on which you have configured a lock-address filter received a packet that was dropped because the packet source MAC address did not match an address learned by the port before the lock took effect.</p> <p>The e<portnum> is the port number.</p> <p>The <mac-address> is the MAC address that was denied by the address lock.</p> <p>Assuming that you configured the port to learn only the addresses that have valid access to the port, this message indicates a security violation.</p>
Warning	mac filter group denied packets on port <portnum> src macaddr <mac-addr>, <num> packets	<p>Indicates that a MAC address filtergroup configured on a port has denied packets.</p> <p>The <portnum> is the port on which the packets were denied.</p> <p>The <mac-addr> is the source MAC address of the denied packets.</p> <p>The <num> indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.</p>
Warning	multicast no software resource: resource-name, rate limited number	<p>IGMP or MLD snooping has run out of software resources. This message is rate-limited to appear a maximum of once every 10 minutes. The rate-limited number shows the number of non-printed warnings.</p>

TABLE 1 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Warning	No global IP! cannot send IGMP msg.	The device is configured for ip multicast active but there is no configured IP address and the device cannot send out IGMP queries.
Warning	No of prefixes received from BGP peer <ip-addr> exceeds warning limit <num>	<p>The Layer 3 switch has received more than the allowed percentage of prefixes from the neighbor.</p> <p>The <ip-addr> is the IP address of the neighbor.</p> <p>The <num> is the number of prefixes that matches the percentage you specified. For example, if you specified a threshold of 100 prefixes and 75 percent as the warning threshold, this message is generated if the Layer 3 switch receives a 76th prefix from the neighbor.</p>
Warning	NTP server <ip-addr> failed to respond	<p>Indicates that a Simple Network Time Protocol (SNTP) server did not respond to the device query for the current time.</p> <p>The <ip-addr> indicates the IP address of the SNTP server.</p>
Warning	rip filter list <list-num> <direction> V1 V2 denied <ip-addr>, <num> packets	<p>Indicates that a RIP route filter denied (dropped) packets.</p> <p>The <list-num> is the ID of the filter list.</p> <p>The <direction> indicates whether the filter was applied to incoming packets or outgoing packets. The value can be one of the following:</p> <ul style="list-style-type: none"> • in • out <p>The V1 or V2 value specifies the RIP version (RIPv1 or RIPv2).</p> <p>The <ip-addr> indicates the network number in the denied updates.</p> <p>The <num> indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.</p>
Warning	Temperature is over warning level.	The chassis temperature has risen above the warning level.

NIAP-CCEVS Certification

Some Brocade devices have passed the Common Criteria (CC) certification testing. This testing is sponsored by the National Information Assurance Partnership (NIAP) - Common Criteria Evaluation and Validation Scheme (CCEVS). For more information regarding the NIAP-CCEVS certification process refer to the following link: <http://www.niap-ccevs.org/>.

In an effort to maintain a proper level of security as it relates to access to network infrastructure resources, Brocade recommends that all Brocade hardware be installed within a secure location that is accessible by approved personnel only.

NIAP-CCEVS certified Brocade equipment and Ironware releases

The following Brocade devices have been NIAP-CCEVS certified. The following IronWare software release must be used to remain compliant with this certification:

TABLE 2 NIAP-CCEVS certified Brocade equipment and IronWare software releases

Brocade product	Brocade IronWare software version	Discussed in
NetIron XMR Family	3.8.00a	<i>NetIron Series Configuration Guide</i>
NetIron MLX Family	3.8.00a	<i>NetIron Series Configuration Guide</i>
BigIron RX Family	2.5.00b	<i>BigIron RX Series Configuration Guide</i>
FastIron SuperX/SX Family	4.1.00	<i>FastIron and Turbolron 24X Configuration Guide</i>
FastIron Edge X Family	4.1.00	<i>FastIron and Turbolron 24X Configuration Guide</i>
FastIron GS/LS Family	4.2.00a	<i>FastIron and Turbolron 24X Configuration Guide</i>
FastIron Edge Switch Family	4.0.00a	<i>Switch and Router Security Guide</i>

TABLE 2 NIAP-CCEVS certified Brocade equipment and IronWare software releases

Brocade product	Brocade IronWare software version	Discussed in
ServerIron JetCore Family	11.0.00a	<i>ServerIron TrafficWorks Graphical User Interface</i> <i>ServerIron TrafficWorks Server Load Balancing Guide</i> <i>ServerIron TrafficWorks Advanced Server Load Balancing Guide</i> <i>ServerIron TrafficWorks Global Server Load Balancing Guide</i> <i>ServerIron TrafficWorks Security Guide</i> <i>ServerIron TrafficWorks Administration Guide</i> <i>ServerIron TrafficWorks Switching and Routing Guide</i> <i>ServerIron Firewall Load Balancing Guide</i>
ServerIron ADX Family	12.0.00	<i>ServerIron ADX TrafficWorks Graphical User Interface</i> <i>ServerIron ADX TrafficWorks Server Load Balancing Guide</i> <i>ServerIron ADX TrafficWorks Advanced Server Load Balancing Guide</i> <i>ServerIron ADX TrafficWorks Global Server Load Balancing Guide</i> <i>ServerIron ADX TrafficWorks Security Guide</i> <i>ServerIron ADX TrafficWorks Administration Guide</i> <i>ServerIron ADX TrafficWorks Switching and Routing Guide</i> <i>ServerIron ADX Firewall Load Balancing Guide</i>

Web-Management access to NIAP-CCEVS certified Brocade equipment

All Brocade networking devices that are to remain in compliancy with the NIAP-CCEVS certification must disable all remote access through the integrated Web management graphical user interface (GUI). In accordance with NIAP-CCEVS this functionality is considered a security risk and must be disabled.

Please refer to the *Brocade Configuration Guides* associated with each product listed in the table “[NIAP-CCEVS certified Brocade equipment and IronWare software releases](#)”, for detailed instructions on how to disable the Web Management Interface feature.

Local user password changes

Please note that if existing usernames and passwords have been configured on a Brocade Device with specific privilege levels (super-user, read-only, port-config), and if you attempt to change a user's password by executing the following syntax:

```
Brocade-Device(config)# user brcdreadonly password value
```

The privilege level of this particular user will be changed from its current value to "super-user". The “super-user” level username and password combination provides full access to the Brocade command line interface (CLI). To prevent this from occurring, use the following syntax:

```
Brocade-Device(config)# user fdryreadonly privilege value password value
```


Local user password changes

Index

Numerics

- 100BaseTX configuration, 57
 - 31-bit subnet mask, 968
 - 802.1Q-in-Q tagging
 - CLI syntax, 808
 - configuration, 804, 806
 - configuration rules, 805
 - configuring profiles, 808
 - enabling, 805
 - 802.1x port security
 - accounting, 1799
 - accounting attributes for RADIUS, 1819
 - accounting configuration, 1818
 - allowing access to multiple hosts, 1815
 - and sFlow, 1798
 - applying IP ACLs and MAC address filters, 1806
 - authenticating multiple hosts, 1795
 - authentication with dynamic VLAN assignment, 1835
 - clearing statistics, 1825
 - communication between the devices, 1791
 - configuration, 1799
 - configuring an authentication method, 1800
 - configuring per-user IP ACLs or MAC address filters, 1809
 - configuring re-authentication, 1811
 - device roles in a configuration, 1790
 - disabling strict security mode, 1807
 - displaying dynamically-assigned VLAN information, 1825
 - displaying information, 1821
 - displaying MAC address and IP ACL information, 1826
 - displaying multiple-host authentication information, 1829
 - displaying statistics, 1824
 - displaying the status of strict security mode, 1828
 - dynamic VLAN assignment, 1802
 - dynamically applying ACLs or MAC address filters, 1808
 - enabling, 1810
 - enabling accounting, 1819
 - hub configuration, 1834
 - IETF RFC support, 1789
 - initializing, 1814
 - MAC address filtering, 1817
 - message exchange during authentication, 1793
 - multi-device authentication and security on the same port, 1836
 - overview, 1790
 - sample configurations, 1833
 - saving dynamic VLAN assignments to the running-config file, 1805
 - setting RADIUS parameters, 1800
 - setting the EAP frame retransmissions, 1814
 - setting the IP MTU size, 1794
 - setting the port control, 1810
 - setting the quiet period, 1812
 - specifying a timeout for retransmission of messages, 1814
 - specifying the RADIUS timeout action, 1801
 - specifying the wait interval, 1812
 - support for RADIUS, 1795
- ## A
- AAA operations for TACACS/TACACS+, 143
 - AAA security for commands pasted into the running-config file, 145
 - access methods
 - disabling SNMP access, 124
 - disabling TFTP access, 124
 - disabling Web management access, 123
 - disabling Web management access by HP ProCurve Manager, 123
 - access policies, ACL and IP, 955
 - access restrictions, remote, 115

ACL

- adding a comment to an entry, 1727
 - adding a comment to an IPv6 entry, 1769
 - and IP access policies, 955
 - applying an IPv4 ACL to a subset of ports (Layer 3), 1735
 - applying an IPv4 ACL to VLAN members (Layer 2), 1734
 - applying egress to CPU traffic, 1725
 - applying IPv6 to a trunk group, 1769
 - applying to a virtual interface in a VLAN, 1728
 - comment text management, 1726
 - configuration example, 1711
 - configuration example for extended named, 1725
 - configuration examples for extended, 1718
 - configuration notes for filtering, 1734
 - configuration tasks for logging, 1730
 - configuring for ARP filtering, 1737
 - configuring IPv6, 1758
 - configuring standard ACLs, 1709
 - configuring the route map, 1750
 - creating IPv6, 1761
 - default and implicit IPv6 action, 1759
 - deleting a comment from an entry, 1727
 - deleting a comment from an IPv6 entry, 1770
 - deny | permit, 1257, 1761
 - displaying ACL information, 1746
 - displaying filters for ARP, 1738
 - displaying IPv6, 1770
 - displaying log entries, 1731
 - DSCP matching, 1744
 - enabling and viewing hardware usage statistics, 1745
 - enabling filtering based on VE port membership, 1734
 - enabling filtering based on VLAN membership, 1734
 - enabling IPv6 on an interface, 1768
 - enabling statistics, 1783
 - enabling strict control of fragmented packet filtering, 1732
 - enabling support for switched traffic, 1733
 - example logging configuration, 1730
 - extended named configuration, 1720
 - extended number configuration, 1714
 - filtering ARP packets, 1736
 - filtering on IP precedence and ToS values, 1739
 - hardware-based configuration considerations, 1708
 - how hardware-based ACLs work, 1707
 - IDs and entries, 1706
 - IPv6 configuration notes, 1757
 - IPv6 overview, 1755
 - IPv6 traffic filtering criteria, 1756
 - ipv6 traffic-filter in, 1769
 - logging, 1729
 - numbering and naming, 1706
 - overview, 1705
 - policy-based routing (PBR), 1747
 - preserving user input for TCP/UDP port numbers, 1725
 - QoS options, 1740
 - remark, 1726
 - standard named configuration, 1711
 - statistics, 1745
 - support for IPv6 logging, 1770
 - supported features on inbound traffic, 1703
 - supported features on outbound traffic, 1704
 - TCP flags and edge port security, 1739
 - troubleshooting, 1747
 - types, 1705
 - using as input to the OSPF distribution list, 1255
 - using to change the forwarding queue, 1743
 - using to control multicast features, 1588, 1745
 - using to limit PIM RP candidate advertisement, 1590
 - using to map the DSCP value, 1741
 - viewing comments, 1727
- ### ACL Log
- acl-logging, 1731
 - logging-enable, 1731
- ### ACL-based inbound mirroring, 931
- ### ACL-based mirroring, configuring for ACLs, 935
- ### ACL-based rate limiting, 1744
- clearing counters, 1785
 - configuring, 1777, 1778
 - inspecting the 802.1p bit, 1781
 - specifying action to be taken for packets that are over the limit, 1781
 - statistics, 1782
 - support for, 1777
 - using traffic policies, 1776
 - viewing counters, 1784

Address Resolution Protocol (ARP)

- changing the aging period, 983
- configuration, 981
- configuring forwarding parameters, 986
- creating static entries, 985
- enabling on an interface, 984
- enabling the proxy, 983, 984
- enabling the proxy globally, 984
- how it works, 981
- rate limiting
 - ARP packets, 982
- static entry support, 986
- address-lock filters, 574
- aggregate links, displaying and determining status, 731
- aggregated VLAN
 - configuring, 799
 - verifying configuration, 801
- alarm interval, setting, 500
- alarm status values, 504
- ARP
 - cache and static table, 950
 - clearing the filter count, 1738
 - configuring an inspection entry, 1930
 - displaying entries, 951, 1070, 1081
- ARP ping
 - setting the wait time, 1021
- authentication
 - entering privileged EXEC mode, 150
- authorization
 - configuring command authorization, 153

B

- banner configuration, 37
- banner, setting a privileged EXEC CLI level, 39
- boot code synchronization, 77
- boot image
 - configuring, 1022
- boot preference, displaying, 81
- BootP
 - changing the IP address used for requests, 1012
 - changing the number of hops, 1013
 - configuration, 1011
 - configuring the reply source address, 1012
 - relay parameters, 1011

Border Gateway Protocol 4 (BGP4)

- adding a loopback interface, 1353
- adding a peer group, 1361
- adding BGP4 neighbors, 1354
- advertising the default route, 1371
- aggregated routes advertised to BGP4 neighbors, 1384
- applying a peer group to a neighbor, 1364
- AS-Path comparison, 1376
- AS-path filtering, 1395
- basic configuration tasks, 1352
- changing administrative distances, 1374
- changing next-hop update timer (optional), 1365
- changing the default local preference, 1370
- changing the keep alive and hold time (optional), 1365
- changing the maximum number of paths for load sharing, 1366
- changing the metric used for route redistribution, 1371
- changing the router ID, 1352
- clearing and resetting BGP4 routes in the IP route table, 1458
- clearing diagnostic buffers, 1460
- clearing route flap dampening statistics, 1421, 1459
- clearing traffic counters, 1458
- closing or resetting a neighbor session, 1457
- configuration and activation, 1348
- configuration notes for BGP4 autonomous systems, 1381
- configuration steps for BGP null 0 routing, 1387
- configuring a BGP confederation, 1382
- configuring a peer group, 1362
- configuring cooperative BGP4 route filtering, 1411
- configuring graceful restart, 1385
- configuring multi-exit discriminators (MEDs), 1377
- configuring timers for graceful restart (optional), 1385
- customizing load sharing, 1368
- defining a community ACL, 1400
- defining a community filter, 1399
- defining an AS-path ACL, 1396
- defining an AS-path filter, 1396
- defining IP prefix lists, 1401
- defining neighbor distribute lists, 1402
- defining route maps, 1403
- disabling or re-enabling re-advertisement of routes to neighbors, 1393
- displaying all routes received from neighbor, 1454
- displaying and clearing route flap dampening

- statistics, 1419
- displaying cooperative filtering information, 1413
- displaying CPU utilization statistics, 1425
- displaying dynamic refresh information, 1456
- displaying filtered routes, 1453
- displaying graceful restart neighbor information, 1451
- displaying information, 1421
- displaying information for a specific route, 1444
- displaying peer group information, 1439
- displaying recursive route lookups, 1372
- displaying route flap dampening statistics, 1449
- displaying route information for a neighbor, 1436
- displaying route-attribute entries, 1447
- displaying routes BGP4 has placed in route table, 1448
- displaying routes whose destinations are unreachable, 1443
- displaying summary BGP4 information, 1422
- displaying summary neighbor information, 1427
- displaying summary route information, 1440
- displaying the active configuration, 1425
- displaying the active route map configuration, 1450
- displaying the best BGP4 routes, 1442, 1443
- displaying the BGP4 route table, 1441
- dynamically requesting a route refresh from a
 - neighbor, 1454
- enabling fast external failover, 1366
- enabling next-hop recursion, 1371
- enabling on the router, 1352
- encryption of MD5 authentication keys, 1359
- entering the route map into the software, 1404
- example of autonomous systems, 1342
- filtering communities, 1399
- generating traps, 1421
- graceful restart, 1347
- KEEPALIVE messages from BGP4 routers, 1346
- match examples using ACLs, 1406
- MED favoring, 1377
- memory considerations, 1351
- message types, 1345
- modifying redistribution parameters, 1391
- NOTIFICATION messages from BGP4 routers, 1347
- null0 routing, 1386
- OPEN messages exchanged with BGP4 routers, 1345
- overview, 1342
- parameter changes, 1350
- parameters, 1349
- peer group configuration rules, 1361
- redistributing connected routes, 1391
- redistributing IBGP routes into RIP and OSPF, 1393
- redistributing OSPF external routes, 1392
- redistributing RIP routes, 1392
- redistributing static routes, 1393
- relationship between the BGP4 route table and IP route table, 1343
- removing route dampening from a neighbor route,

- 1418
- removing route dampening from a route, 1418
- removing route flap dampening, 1459
- requiring the first AS to be the neighbor AS, 1376
- resetting a neighbor session, 1451
- route flap dampening configuration, 1414
- route reflection parameter configuration, 1378
- route reflector configuration, 1380
- router ID comparison, 1376
- selecting a path for a route, 1344
- setting parameters in the routes, 1408
- setting the local AS number, 1353
- show commands for BGP null 0 routing, 1389
- shutting down a session with a BGP4 neighbor, 1364
- special characters for regular expressions, 1397
- specific IP address filtering, 1394
- specifying a list of networks to advertise, 1368
- specifying a route map name, 1369
- specifying the match conditions, 1405
- UPDATE messages from BGP4 routers, 1346
- updating route information, 1451
- updating using soft reconfiguration, 1452
- using a route map to configure route flap dampening, 1416, 1417
- using a table map to set the rag value, 1411
- using regular expressions to filter, 1397
- using the IP default route as a valid next hop, 1370
- Bridge Protocol Data Units (BPDUs)
 - displaying the guard status, 1164
 - enabling protection by port, 1163
 - overview, 1163
 - re-enabling ports, 1164
 - status example, 1165
- broadcast limiting command syntax, 33, 34
- broadcast, multicast, and unknown traffic
 - limiting, 32
 - viewing, 36
- Brocade FCX
 - changing default stacking port configurations, 261
 - configuring default ports, 261
 - hitless supported services and protocols, 330
 - IronStack configuration, 258
 - using secure-setup to build an IronStack, 263
- Brocade FCX-S and FCXS-F devices, configuring stacking ports, 259
- Brocade ICX
 - configuring an IronStack, 265
 - configuring trunked stacking ports, 265
- Brocade IronStack
 - accessing, 273
 - active controller, 326
 - active controller and standby controller elections, 327
 - active controller and standby controller resets, 328
 - bootup role, 327
 - configuration mismatch for stack units, 320
 - configuration notes, 249
 - configuring default stacking port to function as a data

- port, 264
- confirming software versions, 282
- construction methods, 249
- copying the flash image, 283
- device roles and elections, 326
- displaying chassis information, 291
- displaying flash information, 289
- displaying IPC statistics for a specified unit, 301
- displaying memory information, 290
- displaying session statistics for stack units, 297
- displaying software version information, 307
- displaying stack flash information, 296
- displaying stack information, 293
- displaying stack module information, 292
- displaying stack neighbors, 303
- displaying stack port information, 304
- displaying stack resource information, 293
- displaying stacking port interface information, 308
- displaying stacking port statistics, 309
- enabling the stacking mode, 280
- hitless stacking, 329
- image mismatches, 319
- installing a new unit using secure-setup, 311
- installing using static configuration, 311
- list of CLI commands, 279
- MAC address, 275
- managing, 273
- managing partitioning, 284
- MIB support, 285
- moving a unit to another stack, 313
- overview, 236
- persistent MAC addresses, 286
- recovering an earlier stack configuration version, 288
- removing a unit, 312
- removing an active controller from a stack, 313
- removing MAC address entries, 277
- renumbering stack units, 313
- replacing a unit, 312
- software requirements, 248
- stack mismatches, 318
- stack topology, 283
- stackable models, 236
- stacking images, 281
- standby controller, 327
- standby controller role in hitless stacking, 335
- startup configuration files and stacking flash, 325
- supported stack topologies, 326
- Syslog, SNMP, and traps for stack units, 314
- terminology, 237
- three-member IronStack in a ring topology using the automatic setup process, 254
- three-member stack in a ring topology using secure-

- setup, 249, 250
- three-member stack in a ring topology using the manual configuration process, 257
- topologies, 239
- troubleshooting
 - image copy images, 318
 - recovering from a stack unit mismatch, 323
 - secure-setup, 324
 - stacking upgrade, 317
 - unsuccessful stack build, 315
- unconfiguring, 287
- unit identification, 277
- unit priority, 277
- viewing a configuration, 270
- Brocade Ironstack
 - features, 236
 - merging stacks, 285
 - troubleshooting
 - memory allocation failure, 322
- Brocade X series
 - removing buffer allocation limits, 587
- buffer allocation limits, 602
- buffer limits, changing, 52
- buffer profiles
 - configuration, 587
 - configuring for FCX, FWS, and ICX devices, 590
 - configuring on FCX and ICX devices, 593
 - displaying the configuration, 589, 598
 - for VoIP on FastIron stackable devices, 603
 - sample configuration, 592
- buffer sharing
 - configuring, 599
 - displaying information, 601
 - levels, 600
- byte-based limiting command syntax, 35

C

- cable statistics, 499
- cabling requirements for PoE, 665
- CDP
 - clearing information, 442, 445
 - clearing statistics, 442
 - displaying entries, 445
 - displaying information, 443
 - displaying neighbors, 444
 - displaying packet statistics, 442
 - displaying statistics, 445
 - enabling interception of packets globally, 443
 - enabling interception of packets on an interface, 443

- Cisco Discovery Protocol (CDP) overview, 443
- cluster client automatic configuration
 - setting up for MCT, 841
 - with MCT, 835
- command
 - 100-tx, 57
 - aaa accounting dot1x, 1819
 - aaa accounting exec default start-stop radius | tacacs+ | none, 154
 - aaa authentication dot1x default, 1800
 - aaa authentication enable, 149
 - aaa authentication enable | login default, 168
 - aaa authentication enable implicit-user, 150
 - aaa authentication login privilege-mode, 150
 - aaa authentication snmp-server | web-server | enable | login default, 176
 - aaa authorization commands, 170
 - aaa authorization commands default tacacs+ | radius

| none, 153
 access-list, 1709, 1714, 1726, 1743, 1749, 1778
 accounting, 1901
 ACL-logging, 1731
 address-filter, 1394
 add-vlan, 792
 advertise backup, 648, 1676
 age, 1840
 aggregate-address, 1384, 1418
 aggregated-vlan, 800
 alias, 11
 all-client, 116
 appletalk-cable-vlan, 784
 area, 1239, 1290, 1311
 arp, 985, 1187
 as-path filter, 1396
 as-path-filter, 1397, 1398
 as-path-ignore, 1376
 attempt-max-num, 1903
 auth-fail-action restricted-vlan, 1815
 auth-fail-max-attempts, 1816
 auth-fail-vlanid, 1815
 auth-mode none, 1900
 auth-mode passcode static, 1896
 auto-cost reference-bandwidth, 1252, 1294, 1305
 autosave, 1841
 banner exec_mode, 39
 banner incoming, 39
 banner motd, 37
 bgp-redistribute-internal, 1393
 block-applicant all, 889
 block-learning all, 890
 boot system, 81
 bootfile, 1022
 bootp-relay-max-hops, 1013
 bridged-routed, 1733
 broadcast limit, 34, 36
 bsr-candidate ethernet, 1535
 buffer-profile port-region, 595
 buffer-sharing-full, 587, 602
 cdp run, 670
 clear, 526
 clear access-list accounting traffic-policy, 1785
 clear ACL-on-arp, 1738
 clear auth-mac-table, 1866
 clear dhcp, 1935
 clear dot1x statistics, 1825
 clear fdp counters, 442, 446
 clear fdp table, 442, 445
 clear igmp traffic, 1604
 clear ip bgp damping, 1418, 1459
 clear ip bgp flap-statistics, 1421, 1459
 clear ip bgp neighbor, 1413, 1452
 clear ip bgp neighbor all, 1455, 1460
 clear ip bgp routes, 1458
 clear ip bgp traffic, 1458
 clear ip dhcp-server binding, 1020
 clear ip msdp peer, 1578
 clear ip msdp sa-cache, 1579
 clear ip msdp statistics, 1579
 clear ip multicast counters, 1489, 1514
 clear ip multicast mcache, 1489, 1514
 clear ip multicast vlan, 1489, 1514
 clear ip ospf, 1271, 1272
 clear ip ospf neighbor, 1270
 clear ip ospf redistribution, 1271
 clear ip ospf topology, 1271
 clear ip pim traffic, 1557
 clear ip route, 1077
 clear ip tunnel, 1064
 clear ip vrrp-stat, 1694
 clear ipsec statistics, 1314
 clear ipv6 cache, 395
 clear ipv6 mld-snooping counters, 1647
 clear ipv6 mld-snooping mcache, 1647
 clear ipv6 mld-snooping vlan, 1648
 clear ipv6 neighbor, 395
 clear ipv6 rip routes, 1221
 clear ipv6 route, 396
 clear ipv6 traffic, 396
 clear ipv6 tunnel, 378
 clear link-aggregate, 734
 clear link-keepalive statistics, 690
 clear lldp neighbors, 495
 clear LLDP statistics, 490
 clear lldp statistics, 495
 clear logging, 512, 518
 clear mac, 847
 clear mac cluster, 847, 848
 clear mac vlan, 848
 clear mac-address, 277, 563
 clear pim counters, 1557
 clear pim rp-map, 1537
 clear port security, 1843
 clear statistics, 531, 552
 clear statistics dos-attack, 1926
 clear stp-protect-statistics, 1093
 clear table-mac-vlan, 921
 clear web connection, 157, 173
 client-auto-detect, 842
 client-to-client-reflection, 1381
 clock set, 30
 clock summer-time, 31
 clock timezone gmt, 31

clock timezone us, 32
 community-filter, 1399
 compare-routerid, 1377
 confederation identifier, 1383
 confederation peers, 1383
 config-trunk-ind, 712
 console, 274
 console timeout, 115
 copy flash console, 77, 79
 copy flash tftp, 87
 copy running-config, 87
 copy running-config tftp, 84
 copy startup-config tftp, 84
 copy tftp flash, 88
 copy tftp running-config, 86
 copy tftp startup-config, 84
 crypto key client generate | zeroize dsa, 196
 crypto key client generate | zeroize rsa, 196
 crypto key generate | zeroize rsa, 182
 crypto-ssl certificate, 122
 crypto-ssl certificate generate, 138
 cycle time, 1903
 dampening, 1415
 database-overflow-interval, 1268
 dead-interval, 648, 1676
 default-gateway, 121
 default-information-originate, 1263, 1302, 1371
 default-local-preference, 1370
 default-metric, 1206, 1297, 1371
 default-timers, 892
 default-vlan-d, 766
 deny redistribute, 1253
 deploy, 1022
 dhcp snooping client-learning disable, 1934
 dhcp snooping trust, 1934
 dhcp-default-router, 1022
 dhcp-server pool, 1021
 disable, 46
 disable ethernet, 712, 714
 disable multicast-to-cpu, 1592
 disable-hw-ip-checksum-check, 1084
 disable-pim, 1526, 1534
 distance, 1204, 1375
 distance external, 1265
 distribute-list, 1256
 distribute-list prefix-list, 1220, 1299
 distribute-list route-map, 1301
 dns-filter, 1905
 dns-server, 1022
 domain-name, 1022
 dont-advertise-connected, 1208
 dot1x auth-fail-action restrict-vlan, 1816
 dot1x auth-timeout-action succes, 1801
 dot1x initialize ethernet, 1814
 dot1x-enable, 1810
 dual-mode, 819
 dynamic, 784, 785
 enable, 46
 enable aaa console, 170
 enable ethernet, 714
 enable port-config-password, 126
 enable snmp ve-statistics, 22
 enable super-user-password, 126, 151
 enable telnet password, 151
 enable user disable-on-login-failure, 132
 end, 84, 86
 erase flash primary, 92
 erase flash secondary, 92
 erase startup-config, 92
 errdisable recovery interval, 67, 1168
 excluded-address, 1023
 fast port-span, 1104
 fast uplink-span, 1107
 fast uplink-span ethernet, 1108
 fdp advertise ipv4 | ipv6, 438
 fdp holdtime, 439
 fdp timer, 439
 flexible-10g-ports upper, 209
 flow-control, 47
 global-filter-strict-security, 1808
 graceful-restart, 1269, 1385
 graceful-restart restart-time, 1270
 graceful-restart restart-timer, 1385
 graft-retransmit-timer, 1528
 group-router-interface, 794
 gvrp-base-vlan-id, 888
 gvrp-enable, 889
 gvrp-max-leaveall-timer, 889
 hardware-drop, 1579
 hash-chain-length, 1492, 1632
 hello-interval, 648, 1675
 hello-time, 627
 hello-timer, 1527
 hitless-reload, 107
 hold-down-interval, 649
 hostname, 19
 inactivity-timer, 1528
 inline power, 669
 inline power budget, 673
 inline power power-limit, 671
 inline power priority, 674
 interface loopback, 1354
 interface tunnel, 377
 ip access-group, 1709, 1711, 1714

ip access-group frag deny, 1732
 ip access-list, 1733
 ip access-list extended, 1257, 1721
 ip access-list standard, 1256, 1711
 ip address, 365, 1037, 1054
 ip arp-age, 983
 ip as-path access-list, 1396
 ip community-list extended, 1400
 ip community-list standard, 1400
 ip default-gateway, 1037
 ip default-network, 1000
 ip dhcp relay information policy keep, 1940
 ip dhcp snooping vlan, 1934
 ip dhcp-client lease, 1036
 ip dhcp-server, 1021
 ip dhcp-server mgmt, 1020
 ip dhcp-server relay-agent-echo enable, 1021
 ip directed-broadcast, 987, 1922
 ip dns server-address, 971, 1038
 ip forward-protocol udp, 1010
 ip helper-use-responder-ip, 1012
 ip icmp burst-normal, 1923
 ip icmp echo broadcast-request, 989
 ip icmp redirects, 991
 ip igmp group-membership-time, 1520, 1599
 ip igmp max-response-time, 1520, 1600
 ip igmp query-interval, 1599
 ip igmp static-group, 1521
 ip igmp tracking, 1599
 ip igmp version, 1597, 1598
 ip irdp, 1006
 ip load-sharing, 1004
 ip local-proxy-arp, 985
 ip mtu, 975, 1795
 ip multicast, 1470, 1498
 ip multicast age-interval, 1472, 1499
 ip multicast leave-wait-time, 1473, 1500
 ip multicast max-response-time, 1472
 ip multicast mcache-age, 1473, 1500
 ip multicast query-interval, 1472, 1499
 ip multicast report-control, 1472, 1500
 ip multicast verbose-off, 1473, 1500
 ip multicast version, 1469, 1499
 ip multicast-routing, 1519
 ip ospf auth-change-wait-time, 1245
 ip ospf database-filter all out, 1246
 ip ospf network non-broadcast, 1246
 ip ospf network point-to-point, 1269
 ip pim, 1526
 ip pim border, 1535
 ip pism-snooping, 1500
 ip pim-sparse, 1058, 1535
 ip prefix-list, 1401
 ip preserve-ACL-user-input-format, 1725
 ip proxy-arp, 984
 ip proxy-arp enable, 984
 ip radius source-interface ethernet, 979
 ip redirect, 991
 ip rip filter-group, 1210
 ip rip learn-default, 1196
 ip rip poison-reverse, 1196, 1209
 ip route, 994, 995, 1054, 1186, 1259
 ip router-id, 977, 1353
 ip show-portname, 517
 ip show-service-number-in-log, 517
 ip snmp source-interface ethernet, 980
 ip source-route, 988
 ip ssh client, 116
 ip ssh key-authentication yes | no, 186
 ip ssh password-authentication no | yes, 186
 ip ssh permit-empty-passwd no | yes, 187
 ip ssh pub-key-file tftp, 185
 ip ssh source-interface ethernet, 980
 ip ssl certificate-date-file tftp, 138
 ip ssl private-key-file tftp, 138
 ip tacacs source-interface ethernet, 978
 ip tcp burst-normal, 1924
 ip tftp source-interface ethernet, 979
 ip ttl, 987, 1039
 ip use-ACL-on-arp, 1737
 ip vrrp auth-type no-auth | simple-text-auth, 1671
 ip vrrp-extended vrid, 1701
 ip-multicast-disable, 1498
 ip-proto, 784
 ip-subnet, 771, 785
 ipv6 access-list, 366, 1761
 ipv6 address, 361, 363, 366, 378
 ipv6 dns domain-name, 370, 415
 ipv6 dns server-address, 370, 415
 ipv6 enable, 361, 364, 377
 ipv6 hop-limit, 393
 ipv6 icmp error-interval, 383
 ipv6 icmp source-route, 393
 ipv6 mld-snooping, 1636
 ipv6 mld-snooping age-interval, 1637
 ipv6 mld-snooping leave-wait-time, 1638, 1641
 ipv6 mld-snooping mcache-age, 1638
 ipv6 mld-snooping query-interval, 1637
 ipv6 mld-snooping report-control, 1638
 ipv6 mld-snooping verbose-off, 1638
 ipv6 nd dad attempt, 387
 ipv6 nd managed-config-flag, 390
 ipv6 nd ns-interval, 387
 ipv6 nd other-config-flag, 390

- ipv6 nd prefix-advertisement, 389
- ipv6 nd ra-hop-limit, 388
- ipv6 nd ra-interval, 388
- ipv6 nd ra-lifetime, 388
- ipv6 nd reachable-time, 391
- ipv6 nd suppress-ra, 390
- ipv6 neighbor, 392
- ipv6 ospf area, 1291, 1306
- ipv6 ospf authentication ipsec disable, 1313
- ipv6 ospf authentication ipsec spi, 1310
- ipv6 rip default-information, 1218
- ipv6 rip enable, 1217
- ipv6 rip metric-offset, 1219
- ipv6 router ospf, 1289
- ipv6 router rip, 1216
- ipv6 traffic-filter, 1769
- ipv6 unicast-routing, 365, 386, 1669
- ipx-network, 771
- ipx-proto, 784
- keepalive, 1056
- key-rollover-interval, 1310, 1313
- kill console, 140
- learn-default, 1208
- lease, 1022
- legacy-inline-power, 669
- link-aggregate active, 724
- link-aggregate configure, 724, 730
- link-aggregate configure singleton, 735
- link-aggregate off, 725
- link-keepalive ethernet, 687
- link-keepalive interval, 688
- link-keepalive retries, 688
- lldp advertise mac-phy-config-status ports, 472
- lldp advertise management-address ipv4, 467
- lldp advertise max-frame-size ports, 473
- lldp advertise max-frame-size ports ethernet, 473
- lldp advertise port-description ports ethernet, 467
- lldp advertise port-vlan-id ports ethernet, 471
- lldp advertise system-capabilities ports ethernet, 468
- lldp advertise system-description ports ethernet, 469
- lldp advertise vlan-name vlan, 470
- lldp enable ports, 459
- lldp enable ports ethernet, 460
- lldp enable receive ports, 461
- lldp enable snmp notifications ports ethernet, 463
- lldp enable transmit ports, 460
- lldp enable transmit ports ethernet, 461
- lldp max-neighbors-per-port, 462
- lldp max-total-neighbors, 462
- lldp med location-id civic-address, 479
- lldp med location-id coordinate-based, 477
- lldp med location-id ecs-elin, 482
- lldp med network-policy application, 484
- lldp snmp-notification-interval, 463
- lldp transmit-delay, 464
- lldp transmit-hold, 465
- lldp transmit-interval, 464
- local-as, 1383
- lock-address ethernet, 574
- logging buffered, 515
- logging console, 509
- logging enable config-changed, 83
- logging enable user-login, 24
- logging facility, 516
- logging host, 514
- logging on, 514
- logging persistence, 518
- logging-enable, 1731
- log-status-change, 1307
- longpreamble, 264
- loop-detection, 66
- loop-detection-interval, 67
- mac filter, 569, 570
- mac filter log-enable, 572
- mac filter-group log-enable, 570
- mac-age-time, 560
- mac-authentication auth-fail-action, 1856
- mac-authentication disable-aging, 1867
- mac-authentication disable-ingress-filtering, 1860
- mac-authentication dos-protection mac-limit, 1864
- mac-authentication enable, 1855
- mac-authentication mac-filter, 1857
- mac-authentication max-age, 1868
- mac-authentication password-override, 1870
- mac-learn disable, 560
- mac-session-aging no-aging permitted-mac-only, 1816
- mac-vlan-permit, 747
- master, 626
- master-vlan, 609, 1155
- match community, 1406, 1408
- match ip address, 1406, 1750
- match ip address prefix-list, 1406
- match ip next-hop, 1407
- match ip next-hop prefix-list, 1407
- match ip route-source, 1407
- maximum-paths, 1368
- maxreq, 1814
- mdi-mdix, 46
- med-missing-as-worst, 1378
- member-group, 609, 1155
- member-vlan, 609, 1155
- mesh-group, 1567
- message-interval, 1538
- metric-type, 1265, 1297

metro-ring, 626
mirror-port ethernet, 929
mld-snooping disable-mld-snoop, 1639
mld-snooping fast-convergence, 1642
mld-snooping fast-leave-, 1641
mld-snooping proxy-off, 1640
mld-snooping router-port, 1640
mld-snooping static-group, 1640
mld-snooping tracking, 1641
mld-snooping version, 1639
monitor ethernet, 929
mrinfo, 1595
mroute, 1593
msdp-peer, 1562
mstp admin-edge-port ethernet, 1177
mstp admin-pt2pt-mac ethernet, 1177
mstp disable ethernet, 1178
mstp edge-port-auto-detect, 1177
mstp force-migration-check ethernet, 1178
mstp instance, 1175
mstp scope all, 1172
mstp start, 1178
mtu-exceed, 974
multicast active | passive, 1501
multicast fast-convergence, 1475, 1504
multicast fast-leave, 1504
multicast fast-leave-v2, 1475
multicast limit, 33, 34, 36
multicast pimsm-snooping, 1476, 1501
multicast proxy-off, 1474, 1503
multicast router-port ethernet, 1473, 1503
multicast static-group, 1502
multicast tracking, 1475, 1504
multipath ebgp, 1368
name, 626
nbr-timeout, 1527
ncopy, 89
ncopy running-config tftp, 86
neighbor, 1208, 1354, 1363, 1364, 1380, 1402,

1452

netbios-name-server, 1023
network, 1023, 1369
next-bootstrap-server, 1023
next-hop-enable-default, 1370
next-hop-recursion, 1374
no ip icmp unreachable, 990
no mstp instance, 1173
non-preempt-mode, 650
offset-list, 1204
optical-monitor, 500
originator-id, 1563
owner priority, 1684
permit redistribute, 1253
phy-fifo-depth, 53
ping, 95, 370
ping ipv6, 415
poison-local-routes, 1221
poison-reverse, 1221
port-down-auth-mac-cleanup, 1905
port-name, 41, 712
prefix-list, 1299
preforwarding-time, 627
priority, 278
priority ignore-802.1p, 1980
privilege level, 127
protected-link-group, 693
prune-timer, 1527
prune-wait, 1528
pvlan mapping, 814
pvlan pvlan-trunk, 814
pvlan type community, 814, 815
qd-share-level, 599
qos profile, 1990, 1991
qos scheduler-profile, 1975
qos tagged-priority, 1985
qos-tos map dscp-priority, 1984
radius-server host, 163, 165, 1800
radius-server host ipv6, 167
radius-server retransmit, 166
rarp, 1008
rate-limit output shaping, 1953
rate-limit output shaping ethernet, 1954
readvertise, 1393
re-authentication, 1811
redistribute, 1219
redistribute connected, 1391
redistribute ospf match external, 1392
redistribute rip, 1392
redistribute static, 1393
redistribution, 1195
redistribution bgp, 1254

relative-utilization, 553
reload, 283
reload after, 93
reload at, 92
reload cancel, 93
remove-vlan, 792
reserved-vlan-map, 767
restart-ports, 655
restart-vsrp-port, 655
ring-interface ethernet, 627
rmon alarm, 535
rmon history, 534
route-map, 1404, 1750
route-map allowInternalRoutes, 1301
router bgp, 1348, 1416
router msdp, 1562
router ospf, 1197, 1268
router pim, 1525
router rip, 1193, 1202
router vrrp, 1664, 1700
router vrrp-extended, 1668, 1669, 1701
router vsrp, 643
router-interface ve, 790
router-interface-group, 794
rp-address, 1539, 1589
rp-candidate add, 1536
rp-candidate delete, 1536
rp-candidate ethernet, 1591
sa-filter in, 1564
sa-filter originate, 1564
save-current-values, 647
scale-timer, 644
scp, 222
secure-login, 1901
secure-mac-address, 1841
servertimeout, 1814
set interface null0, 1750
set ip next hop, 1750
set ip next-hop peer-address, 1410
set metric-type internal, 1410
sflow agent-ip, 547
sflow enable, 545, 546
sflow export cpu-traffic, 549
sflow export system-info, 548
sflow forwarding, 546
sflow forwarding, 546
sflow max-packet-size, 548
sflow polling-interval, 541
sflow sample, 543, 544
sflow source-port, 545
sflow version 2 | 5, 547
short-path-forwarding, 860

show cable-diag tdr, 499
 show chassis, 276, 291
 show clock, 32
 show flash, 289
 show flash stack, 290
 show interfaces management, 2
 show ip pim dense, 1530
 show license, 229
 show logging, 23
 show media, 57, 501
 show memory, 290
 show module, 292
 show run interface, 36
 show running-config interface management, 2
 show snmp associations, 25
 show snmp server-mode, 29
 show snmp status, 27
 show stack, 287, 293
 show stack connection, 266, 268
 show stack resource, 293
 show statistics management, 3
 show symmetric-flow-control, 52
 show users, 135
 show version, 228, 232
 show web connection, 173
 show who, 7
 slow-start, 1679
 snmp-client, 116
 snmp-client ipv6, 367
 snmp-server community, 80, 91, 423
 snmp-server contact, 19
 snmp-server enable traps holddown-time, 21
 snmp-server engineid local, 426
 snmp-server group, 427, 431
 snmp-server host, 20
 snmp-server host ipv6, 367
 snmp-server location, 19
 snmp-server pw-check, 80, 91
 snmp-server trap ospf, 1267
 snmp-server trap-source ethernet, 981
 snmp-server user, 428
 snmp-server view, 430
 snmp broadcast client, 30
 snmp broadcast server, 30
 snmp poll-interval, 25
 snmp server, 24
 snmp server ipv6, 367
 snmp server-mode, 28
 snmp sync, 31
 spanning-tree, 558, 769, 1091
 spanning-tree 802-1w, 1139
 spanning-tree ethernet, 1092
 spanning-tree root-protect, 1166
 spanning-tree single, 1152
 speed-duplex, 41, 45
 ssh, 197
 stack disable, 281
 stack enable, 280
 stack mac, 276
 stack persistent-mac-timer, 286
 stack secure-setup, 313, 324
 stack switch-over, 343
 stack unconfigure, 287
 stack unconfigure rollback, 288
 stack unit, 262
 stack-port, 262
 static-mac-address, 562, 1978
 stp-bpdu-guard, 1163
 stp-group, 1154
 summary-address, 1241, 1262, 1298
 supptimeout, 1814
 symmetric-flow-control, 51
 symmetric-flow-control set, 52
 system max hw-ip-mcast-mll, 1190
 system max hw-ip-next-hop, 1190
 system max hw-logical-interface, 1190
 system-max dvmrp-max-int-group, 1518
 system-max dvmrp-mcache, 1518
 system-max gre-tunnels, 394, 1055
 system-max hw-traffic-conditioner, 1776
 system-max igmp-snoop-mcache, 1498
 system-max ip-route, 394, 584
 system-max ip-static-arp, 986
 system-max ip-subnet-port, 584
 system-max mld-snoop-mcache, 1636
 system-max pim-max-int-group, 1518
 system-max rmon-entries, 531
 system-max view, 429
 system-max vlan, 795
 system-max-hw-ip-route-tcam, 1188
 tacacs-server key, 148
 tacacs-server retransmit, 148
 tacacs-server timeout, 148
 tagged ethernet, 568, 800
 tag-profile, 808
 tag-type, 800, 805
 telnet, 368, 418
 telnet login-retries, 118
 telnet login-timeout, 118
 telnet server enable vlan, 119
 telnet timeout, 118
 telnet-client, 116
 terminal monitor, 509
 tftp client enable vlan, 120

- tftp-server, 1024
- threshold, 715
- timeout restrict-fw-period, 1818
- timeout tx-period, 1813
- timers keep-alive, 1365
- timers lsa-group-pacing, 1266, 1305
- timers spf, 1264, 1303
- topology-group, 609
- traceroute, 96, 368, 419, 972, 1038
- traceroute ipv6, 368
- traffic-policy, 1778, 1781
- transmit-counter profiles, 527
- trunk deploy, 707
- trunk ethernet, 707, 714
- trunk hash-options include-layer2, 706
- trust-port ethernet, 1902
- tunnel destination, 377
- tunnel loopback, 1053
- tunnel mode gre ip, 1053
- tunnel mode ipv6ip, 377
- tunnel path-mtu-discovery age-timer, 1057
- tunnel path-mtu-discovery disable, 1057
- tunnel source, 377, 1051
- unknown-unicast limit, 34, 36
- untagged ethernet, 784, 785
- update-time, 1366
- use-local-management-mac, 40
- use-vrrp-path, 651, 1209
- vendor-class, 1024
- violation restrict, 1842
- violation shutdown, 1842
- virtual-link-if-address interface ethernet, 1292
- vlan, 567, 745, 784, 785, 800
- vlan-group, 793
- voice-vlan, 61
- vrrp-extended vrid, 1669
- vsrp auth-type no-auth | simple-text-auth, 645
- vsrp vrid, 643
- vsrp-aware vrid, 645, 651
- web access-group ipv6, 369
- web client ipv6, 369
- webauth, 1891
- web-client, 116
- web-management, 14, 123
- web-management enable vlan, 119
- web-management http | https, 137
- write memory, 14, 20

- command line interface
 - banner configuration, 37
 - creating an alias for a command, 11
 - logging in, 4
 - nomenclature on Chassis-based models, 6
 - nomenclature on FESX compact devices, 6
 - nomenclature on stackable devices, 7
 - security levels, 4

command output

- egress queue statistics, 531
- ipv6 mld-snooping mcache, 1644
- ipv6 mld-snooping resource, 1645
- IPv6 tunnel interface information, 379
- sFlow information, 551
- show 802.1w, 1141
- show aaa, 156, 172
- show access-list, 1728
- show ARP, 1071
- show arp, 1081
- show auth-mac-address, 1871
- show default values, 583
- show dot1x, 1821, 1829
- show dot1x config, 1823, 1830
- show dot1x mac-session, 1831
- show dot1x statistics, 1824
- show fdp neighbor, 440
- show gvrp, 894
- show gvrp statistics, 900
- show inline power, 677
- show inline power detail, 682
- show interface tunnel, 1060
- show interfaces stack-ports, 308
- show ip, 1066, 1080
- show ip access-list, 1728
- show ip bgp attribute-entries, 1448
- show ip bgp flap-statistics, 1420, 1449
- show ip bgp neighbors, 1427, 1430, 1437
- show ip bgp route, 1444
- show ip bgp routes detail, 1446
- show ip bgp routes summary, 1440
- show ip bgp summary, 1422
- show ip cache, 1073
- show ip dhcp relay information, 1941
- show ip dhcp-server address pools, 1025
- show ip dhcp-server binding, 1024
- show ip dhcp-server flash, 1026
- show ip dhcp-server summary, 1027
- show ip igmp group, 1601
- show ip igmp interface, 1603
- show ip igmp traffic, 1604
- show ip interface, 1069
- show ip msdp peer, 1575
- show ip msdp sa-cache, 1578
- show ip msdp summary, 1574
- show ip multicast error, 1478
- show ip multicast group, 1479, 1506
- show ip multicast mcache, 1480, 1507
- show ip multicast pimsm-snooping, 1488
- show ip multicast resource, 1481, 1508
- show ip multicast traffic, 1481, 1509

- show ip multicast vlan, 1477, 1510
- show ip ospf area, 1275
- show ip ospf database external-link-state, 1282
- show ip ospf interface, 1278
- show ip ospf neighbor, 1276
- show ip ospf routes, 1279
- show ip ospf trap, 1285
- show ip pim bsr, 1548
- show ip pim dense, 1530
- show ip pim group, 1547
- show ip pim neighbor, 1553
- show ip pim resource, 1550
- show ip pim rp-hash, 1552
- show ip pim rp-set, 1552
- show ip pim sparse, 1546
- show ip pim traffic, 1557
- show ip pimsm-snooping, 1487
- show ip rip, 1211
- show ip route summary, 1076
- show ip ssh config, 189
- show ip static-arp, 1072
- show ip traffic, 1078, 1082
- show ip tunnel traffic, 1061
- show ip vrrp, 1685
- show ip vrrp brief, 1688
- show ip vrrp stat, 1693
- show ip vrrp vrid, 1691
- show ip vrrp-extended, 1685, 1688
- show ipsec policy, 1335
- show ipv6 cache, 397
- show ipv6 interface, 398
- show ipv6 mld-snooping error, 1642
- show ipv6 mld-snooping group, 1643
- show ipv6 mld-snooping traffic, 1646
- show ipv6 neighbor, 400
- show ipv6 ospf area, 1315, 1337
- show ipv6 ospf database, 1317
- show ipv6 ospf database extensive, 1319
- show ipv6 ospf interface, 1322, 1323, 1339
- show ipv6 ospf memory, 1325
- show ipv6 ospf neighbor, 1325
- show ipv6 ospf neighbor router-id, 1326
- show ipv6 ospf redistribute route, 1328
- show ipv6 ospf routes, 1329
- show ipv6 ospf spf node area, 1331
- show ipv6 ospf virtual-link, 1333
- show ipv6 ospf virtual-neighbor, 1333
- show ipv6 rip, 1222
- show ipv6 rip route, 1223
- show ipv6 route command, 402
- show ipv6 router, 403
- show ipv6 tcp connections, 404

- show ipv6 tcp status, 406
- show ipv6 traffic, 407
- show license, 229
- show license unit, 229
- show link-aggregate, 732
- show link-error-disable, 63
- show link-keepalive, 688, 689
- show lldp neighbors, 491
- show lldp statistics, 489, 490
- show loop-detection resource, 69
- show mac-address, 920
- show metro, 628, 629
- show mstp, 1181
- show optic, 503
- show pod unit, 211
- show port security ethernet, 1844
- show port security mac, 1844
- show port security statistics, 1845
- show protected-link group, 694
- show qd-buffer-profile all, 598
- show qos-tos, 1993
- show rate-limit fixed, 1951
- show rate-limit fixed input, 1961
- show reserved-vlan-map, 768
- show sflow, 551
- show snmp associations, 25
- show snmp associations details, 26
- show snmp server-mode, 29
- show snmp status, 27
- show span, 1095
- show span detail, 1100
- show span pvst-mode, 1160
- show spanning-tree 802.1W, 1144
- show stack command, 295
- show stack connection, 310
- show stack flash, 297
- show stack neighbors, 304
- show stack stack-ports, 305
- show statistics stack-ports, 309
- show table-mac-vlan, 915, 919
- show table-mac-vlan denied-mac, 917
- show topology-group, 611
- show traffic-policy, 1786
- show transmit-counter values, 528
- show trunk, 716
- show vsrp, 652
- show vsrp aware, 654
- show webauth, 1916
- commands
 - configuring mac-movement notification interval
 - history, 577
 - line editing, 5
 - mac-movement notification threshold-rate, 575
 - searching and filtering output, 7

configuration

- basic port parameter, 40
- basis system parameters, 18
- Brocade FCX IronStack, 258
- buffer profiles, 587
- command authorization, 153
- DNS resolver, 970
- dynamic loading, 84
- entering system information, 19
- flow control, 47
- hitless OS upgrade, 107
- Interpacket Gap (IPG), 53
- IP addresses, 964
- IP load sharing, 1001
- IP parameters on Layer 2 switches, 1037
- IPv4 and IPv6 protocol stack, 365
- IPv6 connectivity on a Layer 3 switch, 362
- IPv6 management ACLs, 366
- IPv6 management port, 362
- IPv6 neighbor discovery, 384
- IPv6 static neighbor entries, 392
- IPv6 Syslog server, 372
- loading and saving files, 82
- MAC address filter-based mirroring, 936
- manual IPv6 tunnel, 377
- MDI, 45
- mirroring on an IronStack, 930
- packet parameters, 973
- passwords, 129
- PHY FIFO Rx and Tx depth, 53
- port flap dampening, 61
- port mirroring and monitoring, 927
- RADIUS, 160
- RADIUS authorization, 169
- route learning and advertising, 1218
- secure shell (SSH2), 181
- SNMP for an IronStack, 315
- SNMP parameters, 19
- SNMP V3 and SNMP over IPv6, 367
- static IPv6 route, 373
- static routes, 991
- STP per VLAN group, 1154
- TACACS and TACACS+, 145
- TFTP server, 1024
- topology group, 609
- trunk group, 707
- username, 129
- viewing information, 522
- VLAN group, 791
- Voice over IP (VoIP), 60

configuration file

- copying to or from TFTP server, 83
- running, 82
- startup, 82

configuration files

- loading and saving with IPv6, 87
- Configuring an ICX 6450 and ICX 6430 Ironstack, 266
 - Configuring ICX 6430 or ICX 6450 trunked stacking ports, 267
- Connecting ICX 6430 and ICX 6450 devices in a stack, 244
- Connecting ICX 6450 and ICX 6430 devices in a stack
 - Connecting ICX 6430 devices in a stack, 245
 - Connecting ICX 6450 devices in a stack, 245
 - Trunking configuration considerations for ICX 6450 and ICX 6430 devices, 245
- console idle time, defining, 115
- CPU processing
 - command syntax to disable, 1592
- CPU rate-limiting
 - and traffic policies, 1787
- CPU utilization
 - displaying OSPF statistics, 1273
 - displaying statistics, 901, 1097, 1212
 - displaying statistics for VRRP and VRRP-E, 1694
- CPU utilization statistics
 - displaying, 1067

D

- defining Telnet idle time, 118
- denial of service (DoS)
 - avoiding being a victim in a Smurf attack, 1922
 - avoiding being an intermediary in a Smurf attack, 1922
 - displaying information, 1926
 - enabling for multi-device port authentication, 1864
 - Smurf attacks, 1921
 - TCP security enhancement, 1925
 - TCP SYN attacks, 1923
- DHCP
 - changing the IP address used for requests, 1012
 - client-based auto-configuration, 1028, 1030
 - configuration, 1011
 - configuring the reply source address, 1012
 - displaying configuration information, 1034
 - log messages, 1036
 - relay parameters, 1011
- DHCP assist
 - configuring, 1040
 - how it works, 1041

DHCP server

- disabling or re-enabling auto-configuration, 1034
- disabling or re-enabling auto-update, 1034
- supported options, 1033

diagnostic error codes, 93

digital optical monitoring, 500

disabling Syslog messages and traps, 24

displaying on Layer 2 switches, 1080

Distance Vector Multicast Routing Protocol (DVMRP)

- configuration on Layer 3 switch, 1582
- displaying information, 1587
- globally enabling and disabling, 1583
- initiating multicasts on a network, 1580
- modifying interface parameters, 1586
- modifying parameters, 1583
- overview, 1580
- pruning a multicast tree, 1581

DNS

- using to initiate a trace route, 972

DNS resolver

- configuring, 970

DNS server address

- defining, 971

domain list

- defining, 972

domain name

- defining, 971

Domain Name Server (DNS)

- configuring, 1038
- defining an entry, 1038
- using to initiate a trace route, 1038

Dot1x

- auth-fail-action restricted-vlan, 1815
- auth-fail-action restrict-vlan, 1816
- auth-fail-max-attempts, 1816
- auth-fail-vlanid, 1815
- auth-max, 1813
- dot1x disable-filter-strict-security, 1808
- dot1x initialize ethernet, 1814
- enable all, 1810
- enable ethernet, 1810
- global-filter-strict-security, 1808
- mac-session-aging no-aging denied-mac-only, 1816
- mac-session-aging no-aging permitted-mac-only, 1816
- max-req, 1814
- re-authentication, 1811
- save-dynamicvlan-to-config, 1805
- servertimeout, 1814
- supptimeout, 1814
- timeout quiet-period, 1812
- timeout re-authperiod, 1812
- timeout restrict-fwd-period, 1818
- timeout tx-period, 1813

DSA authentication

- configuring challenge-response authentication, 183
- deleting key pairs, 183
- enabling challenge-response, 185
- exporting client public keys, 196
- generating a client key pair, 196
- importing public keys into Brocade device, 184
- providing the public key to clients, 183

DSCP

- using ACL to map, 1741

Dynamic ARP

- about inspection, 1928
- configuration notes and feature limitations, 1929
- poisoning, 1927

Dynamic ARP inspection

- displaying status and ports, 1931
- enabling on a VLAN, 1930
- enabling trust on a port, 1931
- using with IP source guard, 1942

dynamic buffer allocation, 589

dynamic buffer allocation for QoS priorities, 585

dynamic configuration loading, 84

Dynamic Host Configuration Protocol (DHCP)

- binding database, 1932
 - changing the forwarding policy, 1940
 - clearing the binding database, 1935
 - CLI commands, 1018
 - configuration example, 1936
 - configuration flow chart, 1016
 - configuration notes, 1014
 - configuration notes and feature limitations, 1933
 - configuring on a device, 1017
 - configuring snooping, 1934
 - default server settings, 1018
 - defining static IP source bindings, 1944
 - description of, 1013
 - disabling on the management port, 1020
 - disabling the learning of clients on a port, 1934
 - displaying learned IP addresses, 1945
 - enabling and disabling subscriber ID processing, 1940
 - enabling IP source guard on a port, 1944
 - enabling IP source guard on a virtual interface, 1945
 - enabling IP source guard per-port-per-VLAN, 1945
 - option 82, 1937
 - option 82 support, 1014
 - options, 1015
 - overview, 1927
 - relay agent information, 1936
 - removing leases, 1020
 - snooping, 1931
- dynamic link aggregation
- assigning a unique key, 724
 - enabling, 724
 - overview, 718
- dynamic MAC-based VLAN
- CLI commands, 907
 - configuration example, 908
 - configuration notes and feature limitations, 907
 - disabling aging, 912
 - overview, 907
- dynamic port assignment, 766
- dynamic ports
- aging, 783
 - configuring an IP subnet VLAN, 785
 - configuring an IP, IPX, or AppleTalk protocol VLAN, 784
 - configuring an IPX network VLAN, 785
 - disabling membership aging, 783

E

- egress queue counters
- viewing on FCX devices, 530

- egress queue counters, clearing, 531
- EMCP load sharing for IPv6, 381
- encapsulation, changing the type, 973
- error codes used for diagnostics, 93

F

- FastIron IPv6 models, 1190
- FastIron stackable devices
- slot and port designations, 262
- FastIron X series
- configuring total transmit queue depth limit, 585
 - default queue depth limits, 585, 589
 - dynamic buffer allocation for QoS priorities, 585
 - port regions, 556
- FCX devices
- configuring TCAM space, 394
 - viewing egress queue counters, 530
- FDP
- changing the hold time, 439
 - changing the update timer, 438
 - clearing information, 442
 - clearing statistics, 442
 - configuration, 438
 - displaying entries, 441
 - displaying information, 439
 - displaying packet statistics, 442
 - enabling at the interface level, 438
 - enabling globally, 438
 - specifying the IP management address, 438
- feature, 1158
- feature support
- 802.1x port security, 1789
 - base Layer 3 and routing protocols, 1185
 - basic layer 2 features, 555
 - basic software, 17
 - Border Gateway Protocol (BGP4), 1341
 - Brocade stackable devices, 235
 - Denial of Service (DoS), 1921
 - Dynamic Host Configuration Protocol (DHCP), 1927
 - Foundry Discovery Protocol (FDP) and Cisco Discovery Protocol (CDP) packets, 437
 - FWS series switch IPv6 management, 411
 - GARP VLAN Registration Protocol (GVRP), 883
 - hardware component monitoring, 497
 - IP features, 945
 - IP multicast protocols, 1515
 - IP multicast reduction, 1461
 - IP multicast traffic reduction for FastIron WS and

- Brocade FCX and ICX switches, 1491
- IPv6 ACLs, 1755
- IPv6 configuration on FastIron X series, FCX series, and ICX series switches, 353
- Link Layer Discovery Protocol (LLDP), 447
- MAC port security, 1837
- MAC-based VLANs, 905
- management applications, 1
- Metro features, 607
- MLD snooping, 1631
- multi-device port authentication, 1849
- network monitoring, 521
- Open Shortest Path First (OSPF), 1225
- Open Shortest Path First (OSPF), IPv6, 1287
- Operations, Administration, and Maintenance (OAM), 71
- port mirroring and monitoring, 925
- Power over Ethernet (PoE), 659
- Quality of Service, 1963
- rate limiting and rate shaping on FastIron X series and FCX and ICX series switches, 1947
- rate limiting on FastIron WS series switches, 1955
- RIP (IPv4), 1199
- RIP (IPv6), 1215
- rule-based IP ACLs, 1703
- security access, 109
- SNMP access, 421
- software-based licensing, 199
- Spanning Tree Protocol (STP), 1087
- SSH2 and SCP, 179
- Syslog, 507
- traffic policies, 1773
- trunk groups and dynamic link aggregation, 697
- Uni-Directional Link Detection (UDLD), 685
- vlan, 737
- VRRP and VRRP-E, 1649
- web authentication, 1887
- firmware
 - installing PoE, 666
- flash image
 - CLI commands, 75
 - determining version running on device, 72
 - file types, 76
 - update, 1028
 - verification, 74, 75
- flash memory
 - copying a file to, 88

- flow control
 - configuration, 47
 - configuration notes, 47, 50
 - disabling or re-enabling, 47
 - displaying status, 48
 - enabling and disabling, 51
 - negotiation and advertisement, 47
 - symmetric and asymmetric, 49
- flow-based MAC address learning, 564
- Foundry Discovery Protocol (FDP) overview, 437

G

- GARP VLAN Registration Protocol (GVRP)
 - application examples, 884
 - configuration, 888
 - configuration notes, 886
 - displaying information, 893
 - displaying statistics, 898
 - dynamic core and dynamic edge, 886
 - dynamic core and fixed edge, 885
 - enabling, 889
 - fixed core and dynamic edge, 886
 - fixed core and fixed edge, 886
 - overview, 883
 - VLAN names created by GVRP, 886
- Generating, 196
- global VLAN
 - displaying information, 823
- GRE statistics, clearing, 1064
- GRE tunnels
 - changing the maximum number supported, 1055
 - configuration considerations, 1048
 - configuration tasks, 1050
 - configuring GRE link keepalive, 1056
 - displaying information, 1059, 1062
 - enabling IPv4 multicast routing, 1057
 - example point-to-point configuration, 1058
 - loopback ports, 1049
- GRE, support with other features, 1047
- GVRP
 - changing the GVRP base VLAN ID, 888
 - changing timers, 890
 - command examples, 902
 - converting a VLAN created by GVRP into a statically-configured VLAN, 892
 - default-timers, 892
 - displaying information, 894
 - dynamic core and dynamic edge configuration

- example, 903
- dynamic core and fixed edge configuration example, 902
- fixed core and dynamic edge configuration example, 904
- fixed core and fixed edge configuration examples, 904
- gvrp block applicant all, 889
- gvrp block-learning all | ethernet, 890
- join-timer, 892
- leave-timer, 892

H

- hitless failover
 - description, 97
- hitless failover, enabling, 100
- hitless management, 97
 - benefits of, 98
 - configuration notes and feature limitations, 100
 - supported protocols, 98
- hitless OS upgrade, 97, 104
- hitless OS upgrade configuration, 107
- hitless reload or switchover, 100
- hitless stacking, 329
 - configuration notes and feature limitations, 333
 - default behavior, 340
 - displaying information, 350
 - displaying status, 341
 - failover, 342
 - standby controller role, 335
 - support during stack formation, merge, and split, 337
 - switchover, 343
- hitless switchover
 - description, 97
 - executing, 104

I

- ICMP
 - configuring rate limiting, 383
 - disabling redirect messages, 990
 - enabling redirect messages, 384
- ICMP feature configuration for IPv6, 383
- ICMP Router Discovery Protocol (IRDP)
 - configuration, 1004
 - enabling globally, 1005
 - enabling on an individual port, 1006
 - parameters, 1005

IGMP

- changing V1 and V2 parameters, 1519
- configuring for individual ports in a VLAN, 1470
- configuring the mode, 1469
- disabling transmission and receipt of packets on a port, 1471
- membership tracking and fast leave for the VLAN, 1474
- modifying membership time, 1520
- modifying query interval period, 1519
- modifying the age interval for group membership entries, 1471
- IGMP on FastIron WS and Brocade FCX and ICX switches
 - clear commands, 1514
 - clearing IGMP counters on VLANs, 1514
 - clearing IGMP mcache, 1514
 - clearing mcache on a specific VLAN, 1514
 - clearing traffic on a specific VLAN, 1514
 - configuring fast leave, 1504
 - displaying errors, 1505
 - displaying group information, 1505
 - displaying mcache information, 1507
 - displaying querier information, 1510
 - enabling fast convergence, 1504
 - enabling membership tracking and fast leave for the VLAN, 1503
 - turning off static group proxy, 1503
- IGMP snooping
 - clear commands, 1489
 - clearing IGMP counters on VLANs, 1489
 - clearing the IGMP mcache, 1489
 - clearing the mcache on a specific VLAN, 1489
 - clearing traffic on a specific VLAN, 1489
 - configuration, 1467
 - configuring report control, 1472
 - disabling on a VLAN, 1471
 - displaying mcache information, 1480
 - displaying querier information, 1482
 - displaying software resource usage for VLANs, 1480
 - displaying the status, 1481
 - MAC-based implementation on FastIron X series, 1462
 - modifying the maximum response time, 1472
 - modifying the query interval, 1472
 - overview, 1461
 - queriers and non-queriers, 1463
 - together with Layer 3 multicast routing on same device, 1464
 - tracking and fast leave, 1463
 - VLAN-specific configuration, 1463
- IGMP snooping on FastIron WS and Brocade FCX and ICX

- switches
 - configuration notes, 1493
 - disabling for the VLAN, 1501
 - displaying information, 1505
 - displaying information by VLAN, 1510
 - IGMPv2 with IGMPv3, 1495
 - overview, 1491
 - querier and non-querier configuration, 1494
 - VLAN-specific configuration, 1495
- IGMP snooping traffic on FastIron WS and Brocade FCX and ICX switches
 - displaying status, 1509
- Integrated Switch Routing (ISR), 752
- Interface
 - 100-fx, 59
 - 100-tx, 57
 - acl-mirror-port ethernet, 932
 - advertise-local on | off, 1586
 - age, 1840
 - arp inspection trust, 1931
 - broadcast limit, 34, 36
 - dhcp snooping relay information, 1939
 - dhcp snooping relay information option subscriber-id, 1940
 - dhcp-gateway-list, 1043
 - dot1x auth-filter, 573
 - dot1x auth-timeout-action failure, 1802
 - dot1x auth-timeout-action success, 1801
 - dot1x port-control auto, 1811
 - dot1x re-auth-timeout-success, 1802
 - dual-mode, 819, 1159
 - enable, 46, 1839
 - flow-control, 48
 - gig-default, 59
 - idhcp snooping trust, 1934
 - inline power, 669
 - inline power power-by-class, 673
 - inline power power-limit, 671
 - inline power priority, 674
 - interface loopback, 1354
 - interface ve, 967
 - ip access-group frag deny, 1732
 - ip access-group in, 1778
 - ip address, 365, 965
 - ip arp-age, 983
 - ip bootp-gateway, 1013
 - ip dhcp-client enable, 1034
 - ip dvmrp metric, 1586
 - ip encapsulation snap, 973
 - ip follow ve, 968
 - ip helper-address, 1011
 - ip icmp burst-normal burst-max lockup, 1923
 - ip igmp port-version, 1598
 - ip igmp proxy, 1605
 - ip igmp tracking, 1599
 - ip igmp version, 1598
 - ip irdp, 1006
 - ip local-proxy-arp, 985
 - ip metric, 1203
 - ip mtu, 975, 1795
 - ip ospf auth-change-wait-time, 1245
 - ip ospf database-filter all out, 1246
 - ip ospf network non-broadcast, 1246
 - ip ospf network point-to-point, 1269
 - ip pim border, 1535
 - ip pim ttl, 1530
 - ip pim version, 1526
 - ip pim-sparse, 1535
 - ip policy route-map, 1751
 - ip proxy-arp enable | disable, 984
 - ip rip filter-group in | out, 1210
 - ip rip learn-default, 1208
 - ip rip poison-reverse, 1196, 1209
 - ip rip v1-only | v1-compatible-v2 | v2-only, 1193,

1202

ip tcp burst-normal burst-max lockup, 1924
ip use-acl-on-arp, 1737
ip vrrp-extended auth-type no-auth | simple-text-auth, 1672
ip vrrp auth-type no-auth | simple-text-auth, 645
ipg, 56
ipg-gmii, 54
ipg-mii, 54
ipg-xgmii, 54
ip-multicast-disable, 1471, 1498
ipv6 address, 363, 364, 365, 366
ipv6 address eui, 364
ipv6 address link-local, 365
ipv6 enable, 364
ipv6 mtu, 392
ipv6 nd dad attempt, 387
ipv6 nd managed-config-flag, 390
ipv6 nd ns-interval, 387
ipv6 nd other-config-flag, 390
ipv6 nd prefix-advertisement, 389
ipv6 nd ra-interval, 388
ipv6 nd ra-lifetime, 388
ipv6 nd reachable-time, 391
ipv6 nd suppress-ra, 390
ipv6 ospf area, 1291
ipv6 redirects, 384
ipv6 rip default-information only | originate, 1218
ipv6 rip enable, 1217
ipv6 rip metric-offset, 1219
ipv6 rip metric-offset out, 1219
ipv6 rip summary-address, 1219
ipv6-multicast-disable, 1617, 1636
link-aggregate, 725
link-aggregate configure, 725, 730
link-aggregate configure singleton, 735
link-aggregate configure timeout, 730
link-error-disable, 62
link-fault-signal, 605
mac filter log-enable, 572
mac filter-group log-enable, 570, 572
mac-authentication apply-mac-auth-filter, 1857
mac-authentication auth-fail-action block-traffic, 1857
mac-authentication auth-fail-action restrict-vlan, 1869
mac-authentication auth-fail-vlan-id, 1856
mac-authentication auth-timeout-action failure, 1869
mac-authentication auth-timeout-action success,

1869

mac-authentication clear-mac-session, 1866
mac-authentication disable-aging, 913, 1867
mac-authentication disable-ingress-filtering, 1860
mac-authentication dos-protecti, 1864
mac-authentication enable, 1856
mac-authentication enable-dynamic-vlan, 1858
mac-authentication max-accepted-session, 1870
mac-authentication move-back-to-old-vlan, 1860
mac-authentication no-override-restrict-vlan, 1858
mac-authentication source-guard-protection enable, 1865
mac-learn disable, 560
maximum, 1840
mdi-mdix, 46
monitor ethernet, 929
multicast limit, 34, 36
no ip address, 968
optical-monitor, 500
per-vlan, 1735
phy-fifo-depth, 53
port security, 1839
port-name, 41
priority, 1978
pvst-mode, 1159
rate-limit input fixed, 1949, 1958
rate-limit output fixed, 1959
rate-limit output shaping, 1953
rate-limit output shaping ethernet, 1954
restrict-vlan, 1856
secure-mac-address, 1841
set interface null0, 1750
source-guard enable, 1944, 1945
spanning-tree root-protect, 1166
speed-duplex, 41, 45
stp-bpdu-guard, 1163
stp-protect, 1093
tagged ethernet, 892
tag-profile, 758
tag-profile enable, 808
track-port ethernet, 1700
ttl-threshold, 1586
unknown-unicast limit, 34, 36
use-radius-server, 165
violation restrict, 1842
violation shutdown, 1843
voice-vlan, 61
vrrp restart-port, 655
interface module
 setting the power budget for PoE, 673
Internet Control Message Protocol (ICMP)
 disabling messages, 989

- Internet Group Management Protocol (IGMP)
 - and IP multicast protocols on the same device, 1606
 - clearing statistics, 1604
 - default version, 1597
 - displaying information on Layer 3 switches, 1600
 - displaying proxy traffic information, 1606
 - displaying the status of an interface, 1602
 - displaying the traffic status, 1604
 - enabling membership tracking and fast leave, 1598
 - enabling on a physical port within a virtual routing interface, 1598
 - globally enabling, 1597
 - overview, 1596
 - proxy configuration, 1605
 - setting the group membership time, 1599
 - setting the maximum response time, 1600
 - source-specific multicast protocols, 1600
- Interpacket Gap (IPG) configuration, 53
- IP
 - basic configuration, 946
 - basic parameters and defaults (Layer 2), 962
 - basic parameters and defaults (Layer 3), 955
 - configuration overview, 947
 - configuring load sharing, 1001
 - displaying global configuration information, 1066
 - displaying Layer 3 switch information, 1065
 - displaying network mask information, 1065
 - edge Layer 3 support, 947
 - full Layer 3 support, 947
 - global parameters (Layer 3 switches), 956
 - global parameters for Layer 2 switch, 962
 - interface parameters (Layer 2), 964
 - interface parameters (Layer 3), 959
 - Layer 4 session table, 953
- ip, 1396
- IP access policies, 955
- IP address
 - assigning to a Layer 2 switch, 1037
 - assigning to a loopback interface, 966
 - assigning to a virtual interface, 966
 - assigning to an Ethernet port, 965
 - deleting, 968
 - specifying for all Telnet packets, 978
- IP address pool
 - configuring the lease duration, 1022
- IP addresses
 - configuring, 964
- IP checksum check
 - disabling, 1084
- IP configuration
 - displaying information and statistics, 1064
- IP Follow
 - configuring on a virtual routing interface, 967
- IP forwarding cache, 953
- IP forwarding cache, displaying, 1073
- IP helper
 - configuring an address, 1011, 1012
- IP information, 1080
 - displaying global configuration, 1080
- IP interface information, displaying, 1069
- IP interface redundancy protocols, 954
- IP interfaces
 - Layer 2 switches, 949
 - Layer 3 switches, 948
- IP load sharing
 - changing the number of ECMP paths, 1004
 - how it works, 1003
 - path cost parameters, 1002
 - path costs in IP route table, 1001
 - static route, OSPF, and BGP4, 1003
- IP multicast
 - changing dynamic memory allocation for groups, 1518
 - changing IGMP parameters, 1519
 - defining the maximum number of PIM cache entries, 1519
 - defining the number of DVMRP cache entries, 1518
 - displaying all cache entries in a pruned state, 1531
 - failover time in a multi-path topology, 1529
 - global parameters, 1517
 - grafts to a multicast tree, 1524
 - initiating PIM multicasts on a network, 1522
 - IPv4 group addresses, 1516
 - mapping of IPv4 group addresses to Ethernet MAC addresses, 1516
 - modifying IGMP maximum response time, 1520
 - modifying IGMP query interval period, 1519
 - modifying the time to live (TTL), 1529
 - PIM dense, 1522
 - PIM DM configuration, 1525
 - PIM DM versions, 1524
 - pruning a multicast tree, 1522
 - supported Layer 3 routing protocols, 1517
 - suppression of unregistered packets, 1517
 - terms, 1517
- IP multicast protocols, 954
- IP packet
 - flow through a Layer 3 switch, 949
- IP packets
 - disabling forwarding, 988
- IP parameters
 - configuring on Layer 2 switches, 1037
- IP route exchange protocols, 954

- IP route table, 951
- IP route table, displaying, 1074
- IP routes
 - clearing, 1077
- IP source guard
 - configuration notes and feature limitations, 1943
- IP static routes
 - enabling redistribution into RIP, 1193
- IP subnet address
 - configuring on multiple port-based VLAN, 788
- IP subnet broadcasts, enabling support, 988
- IP traffic
 - displaying statistics, 1082
- IP tunnels
 - configuration, 1587
- IPv4
 - configuring Layer 3 system parameters, 1189
 - disabling CPU processing for multicast groups, 1591
 - enabling multicast routing on GRE tunnels, 1057
 - GRE packet, 1044
 - multicast routing over GRE tunnels, 1046
 - point-to-point GRE tunnels, 1043
- IPv4 and IPv6
 - protocol stack configuration, 365
- IPv4 route, tracing, 96
- IPv6
 - ACL configuration notes, 1757
 - ACL traffic filtering criteria, 1756
 - address types, 356
 - addressing, 412
 - addressing overview, 355
 - advertising address summaries, 1218
 - changing the MTU, 392
 - clearing global information, 395
 - clearing neighbor information, 395
 - clearing RIPng routes, 1221
 - clearing routes from IPv6 route table, 396
 - clearing the cache, 395
 - clearing traffic statistics, 396
 - clearing tunnel statistics, 378
 - configuring a global or site-local address, 361
 - configuring a link-local address as a system-wide address for a switch, 361
 - configuring a manual tunnel, 377
 - configuring a Syslog server, 372
 - configuring address resolution using DNS resolver, 370
 - configuring an ACL, 1758
 - configuring anycast address, 365
 - configuring basic connectivity on a Layer 3 switch, 362
 - configuring ICMP rate limiting, 383
 - configuring IPv6 management ACLs, 366
 - configuring on each router interface, 362
 - configuring reachable time for remote nodes, 391
 - configuring SNMP V3, 367
 - configuring SNTP, 367
 - configuring the management port, 362
 - creating an ACL, 1761
 - default and implicit ACL action, 1759
 - defining a DNS entry, 370, 415
 - disabling on a Layer 2 switch, 373
 - disabling router advertisement and solicitation messages, 373
 - disabling router advertisement and solicitation messages, 373
 - displaying cache information, 397
 - displaying ECMP load-sharing information, 382
 - displaying global information, 396
 - displaying interface information, 398
 - displaying interface-level settings, 380
 - displaying local routers, 402
 - displaying neighbor information, 400
 - displaying route table, 401
 - displaying TCP information, 403
 - displaying traffic statistics, 407
 - displaying tunnel information, 378
 - ECMP load sharing, 381
 - enabling and disabling, 413
 - enabling and disabling router advertisements, 390
 - enabling ICMP redirect messages, 384
 - enabling routing, 362
 - establishing a telnet session, 418
 - full layer supported features, 355
 - host address on a Layer 2 switch, 360
 - ICMP feature configuration, 383
 - limiting the number of hops a packet can traverse, 393
 - logging, 414
 - management on FastIron X series devices, 366
 - maximum transmission unit (MTU), 391
 - name-to-IPv6 address resolution using DNS server, 415
 - neighbor discovery configuration, 384
 - neighbor redirect messages, 386
 - neighbor solicitation and advertisement messages, 370

- 385
- pinging, 415
- pinging and address, 370
- prefixes advertised in router messages, 389
- protocol names and numbers, 1756
- protocol VLAN configuration, 776
- restricting web access, 414
- router advertisement and solicitation messages, 386
- secure shell (SSH) and SCP, 417
- secure shell and SCP, 367
- setting flags in router advertisement messages, 390
- setting neighbor solicitation parameters, 387
- setting router advertisement parameters, 387
- SNMP3, 417
- SNTP, 417
- source routing security enhancements, 393
- specifying a Syslog server, 414
- specifying an SNMP trap receiver, 367, 417
- stateless auto-configuration, 358
- static neighbor entries configuration, 392
- static route configuration, 373
- static route parameters, 375
- supported CLI commands, 358
- Telnet, 417
- telnet, 367
- traceroute, 368, 419
- viewing SNMP server address, 372
- web management using HTTP and HTTPS, 413
- web management using HTTP and HTTPs, 369
- IPv6 ACL
 - adding a comment to an entry, 1769
 - applying to a trunk group, 1769
 - command syntax descriptions, 1763
 - configuring for ICMP, 1761
 - configuring for TCP, 1762
 - configuring for UDP, 1762
 - deleting a comment from an entry, 1770
 - displaying, 1770
 - enabling on an interface, 1768
 - permit | deny, 1761
 - router remark, 1769
 - support for logging, 1770
- IPv6 host support, 366
- IPv6 management
 - ACLs, 413
 - CLI commands, 419
 - features, 411
 - overview, 412
- IPv6 over IPv4 tunnels, 376
- ipv6 rip summary-address, 1219
- IPv6 TFTP server file upload, 90

J

- jumbo frame support for Layer 2, 606

L

LACP

- additional trunking options, 725
- configuration notes and limitations for configuring trunk group, 721
- configuration notes for single link, 735
- displaying status information, 734
- single link configuration, 734
- trunk group examples, 719

Layer 2

- adding information to trunk hash output, 706
- changing the MAC age time, 560
- disabling MAC address learning, 560
- enabling and disabling STP, 558
- enabling or disabling, 1197
- jumbo frame support, 606
- MAC learning rate control, 559
- management MAC address, 559
- MCT protocols, 850
- multicast failover with MCT, 852
- port loop detection with MCT, 849
- port regions, 556
- unicast with MCT, 855

Layer 2 switch

- basic IP parameters and defaults, 962
- configuring IP parameters, 1037
- displaying IP information, 1080
- interface IP parameters, 964
- IP global parameters, 962
- IPv6 host address, 360

Layer 3

- base with FWS, 1188
- behavior with MCT, 854
- configuring system parameters on FastIron X series IPv4 models, 1189
- modifying and displaying parameter limits, 1188

Layer 3 switch

- basic IP parameters and defaults, 955
- configuring a default network route, 1000
- configuring to drop IP packets, 995
- IP global parameters, 956
- IP interface parameters, 959

- license file
 - copying using TFTP, 221
 - deleting, 222
 - verifying installation, 222
- licensed features and part numbers, 202
- Licensing for Ports on Demand, 208
 - Configuration considerations for stacking or trunking PoD ports, 213
 - Configuration considerations when configuring PoD on an interface, 213
 - Configuring PoD on an interface, 208
 - Configuring the upper PoD ports in a stack for ICX 6610 devices, 209
 - Upgrading or downgrading configuration considerations for PoD, 208, 212
- link aggregation
 - clearing the negotiated links table, 734
 - configuring parameters, 729
 - displaying port status information for FastIron stackable devices, 734
 - key, 726
 - parameters, 725
 - port priority, 726
 - system priority, 726
 - timeout, 726
 - viewing keys for tagged ports, 728
- link aggregation group (LAG)
 - with MCT, 846
- link fault signaling (LFS)
 - enabling, 605
 - for Ethernet devices, 604
 - viewing the status, 605
- Link layer discovery protocol (LLDP), description of, 447
- LLDP
 - 802.1 capabilities when enabled, 470
 - 802.3 capabilities when enabled on a global basis, 471
 - basic management TLV, 454
 - benefits, 450
 - changing a port operating mode, 459
 - changing the holdtime multiplier, 464
 - changing the interval between regular transmissions, 464
 - changing the minimum time between port reinitializations, 465
 - changing the minimum time between transmissions, 463
 - clearing cached LLDP neighbor information, 495
 - configuration summary, 488
 - displaying neighbors detail, 492
 - enabling and disabling, 459
 - enabling SNMP notifications and Syslog messages, 462
 - enabling support for tagged packets, 459
 - general operating principles, 452
 - general system information, 466
 - global configuration tasks, 458
 - MIB support, 457
 - organizationally-specific TLVs, 454
 - overview, 449
 - packets, 453
 - receive mode, 453
 - resetting statistics, 495
 - specifying the maximum number of LLDP neighbors per port, 462
 - specifying the maximum number of neighbors per device, 462
 - specifying the minimum time between SNMP traps and Syslog messages, 463
 - Syslog messages, 457
 - terms used in chapter, 448
 - TLVs advertised by Brocade device, 465
 - transmit mode, 452
- LLDP media endpoint devices (LLDP-MED), description of, 447
- LLDP_MED
 - benefits, 451
 - general operating principles, 452
 - overview, 450
- LLDP-MED
 - 802.3af power classes, 487
 - attributes advertised by the Brocade device, 485
 - capabilities, 485
 - class, 452
 - configuration tasks, 474
 - configuring civic address location, 479
 - configuring emergency call service, 482
 - coordinate-based location, 477
 - defining a location ID, 476
 - defining a network policy, 483
 - displaying statistics and configuration settings, 488
 - elements used with civic address, 480
 - enabling, 475
 - enabling SNMP notifications and Syslog messages, 475
 - example civic address location advertisement, 482
 - extended power-via-MDI information, 486
 - fast start repeat count, 476
 - TLVs, 455
- load balancing, configuring using multiple static routes, 996

- Local User Database
 - delete-all, 1893
 - import-users tftp filename, 1894
 - no username, 1893
 - username password, 1893
- log messages for DHCP, 1036
- logging, 514
- logging changes to, 83
- login attempts, specifying maximum number for Telnet
 - access, 118
- loop detection
 - clearing, 68
 - configuring a global interval, 66
 - displaying resource information, 69
 - enabling, 66
 - specifying the recovery time interval, 67

M

- MAC address
 - clearing entries, 563
 - clearing flow-based entries, 567
 - cluster types with MCT, 846
 - configuration, 561
 - configuring flow-based learning, 566
 - configuring the maximum per port, 913
 - defining filters, 569
 - displaying flow-based learning configuration, 567
 - filter-based mirroring, 936
 - filters command syntax, 570
 - filters for EAP frames, 1817
 - flow-based learning, 564
 - MDUP with MCT, 846
 - movement from port-to-port, 574
 - multi-port static, 561
 - using for Layer 2 traffic, 40
 - VLAN-based static configuration, 563
- MAC address filter
 - command syntax, 572
 - creating a filter with a mirroring clause, 936
 - enabling logging of management traffic, 571
 - override for 802.1x-enabled ports, 572
- MAC address for stackable devices, 559
- MAC address learning, disabling, 560
- MAC address movement
 - notification collection interval, configuring, 577
 - statistics, viewing, 577
 - threshold rate, configuring, 575
 - viewing threshold rate configuration, 576
- MAC address table, displaying, 561

- MAC addresses
 - displaying, 917
 - displaying in a MAC-based VLAN, 920
- MAC age time, changing, 560
- MAC Database Update (MDUP)
 - displaying packet statistics, 848
 - with MCT, 846
- MAC learning rate control, 559
- MAC port security
 - autosaving to the startup configuration, 1841
 - clearing restricted MAC addresses, 1843
 - clearing statistics, 1843
 - clearing violation statistics, 1843
 - configuration, 1839
 - configuration notes and feature limitations, 1838
 - disabling the port, 1842
 - displaying information, 1844
 - displaying restricted MAC addresses on a port, 1847
 - displaying secure MAC addresses, 1844
 - displaying statistics, 1845
 - enabling, 1839
 - local and global resources, 1838
 - overview, 1838
 - setting the age timer, 1840
 - setting the maximum number of addresses, 1840
 - specifying secure MAC addresses, 1841
- MAC-based VLAN
 - aging, 911
 - and port up or down events, 906
 - clearing information, 921
 - configuration, 909
 - configuring for a dynamic host, 914
 - configuring for a static host, 913
 - configuring using SNMP, 915
 - displaying information, 915
 - displaying logging, 921
 - dynamic configuration, 914
 - feature structure, 906
 - overview, 905
 - policy-based classification, 906
 - sample application, 921
 - source MAC address authentication, 906
 - static and dynamic hosts, 905
 - using with 802.1x security on the same port, 910
- mac-movement notification interval-history command, 577
- mac-movement notification threshold-rate command, 575
- MAC-VLAN
 - displaying for a specified interface, 919
- management application feature support, 1
- management function restrictions, 112

- management interface
 - logging on, 12
 - navigating, 13
 - using, 14
- management IP address
 - configuring and specifying the default gateway, 1037
- management port
 - commands, 2
 - overview, 1
 - rules, 2
- management privilege levels, 126
- management privileges, 129
- Maximum Transmission Unit (MTU)
 - changing, 974
 - changing on an individual port, 975
 - path discovery (RFC 1191) support, 976
- Maximum Transmission Unit (MTU)
 - globally changing, 975
- MDI configuration, 45
- Media Dependent Interface (MDI) configuration, 45
- media, displaying information, 501
- metro ring protocol (MRP)
 - adding a ring to a VLAN, 626
 - CLI example, 631
 - configuration, 625
 - configuration notes, 613
 - diagnostics, 628
 - displaying MRP diagnostics, 628
 - displaying topology group information, 629
 - enabling MRP diagnostics, 628
 - how ring breaks are detected and healed, 621
 - master VLANs and customer VLANs, 623
 - overview, 611
 - phase 1–MRP rings without shared interfaces, 613
 - phase 2–MRP rings with shared interfaces, 614
 - ring initialization, 616
- mirror port
 - defining, 936
 - specifying the destination for physical ports, 932
 - specifying the destination for trunk ports, 933
- mirroring
 - ACL-based, 931
 - configuration on an IronStack, 930
 - configuring ACLs bound to virtual interfaces, 935
 - configuring for ports on different members in an IronStack, 931
 - configuring for ports on the same stack member in an IronStack, 931
 - configuring MAC address (filter-based), 936
 - VLAN-based, 937
- MRP
 - diagnostics, 628
 - enable, 627
 - hello-time, 627
 - master, 626
 - metro-ring, 626
 - name, 626
 - preforwarding-time, 627
 - ring-interface ethernet, 627
- MSTP
 - forcing ports to transmit a BPDU, 1178
 - setting the name, 1175
 - setting the revision number, 1175
- MTU
 - changing the value for a tunnel interface, 1055
 - configuring path discovery, 1056
 - path discovery, 1045
- MTU for IPv6, 391
- multicast
 - enabling or disabling error and warning messages, 1473
 - Layer 2 failover with MCT, 852
 - modifying the cache age time, 1473
 - snooping configuration for MCT, 879
 - snooping with MCT, 851
 - turning off static group proxy, 1474
- multicast groups
 - disabling CPU processing, 1591
- Multicast Learning Discovery (MLD) snooping
 - configuration notes, 1633
 - configuration on VLANs, 1634
 - configuring queriers, 1634
 - configuring report control, 1637
 - configuring static groups to VLAN or individual ports, 1640
 - configuring static router ports, 1640
 - configuring the global mode, 1636
 - configuring the global version, 1637
 - configuring the version for individual ports, 1639
 - configuring the VLAN mode, 1639
 - configuring the VLAN version, 1639
 - disabling error and warning messages, 1638
 - disabling packets on a port, 1636
 - disabling snooping for the VLAN, 1639
 - displaying information, 1642
 - enabling fast convergence, 1642
 - enabling MLDv2 tracking and fast leave for VLAN,

- 1641
- global task configuration, 1635
- modifying the age interval, 1637
- modifying the mcache aging time, 1638
- modifying the query interval, 1637
- modifying the wait time, 1638
- overview, 1631
- turning off static group proxy, 1640
- VLAN-specific task configuration, 1635
- multicast neighbor
 - displaying information, 1553
- multicast protocols
 - displaying information, 1062
- multicast route
 - configuring static, 1593
 - displaying the configuration, 1595
- Multicast Source Discovery Protocol (MSDP)
 - clearing information, 1578
 - configuration, 1562
 - configuring a mesh group, 1567
 - configuring peers, 1562
 - displaying information, 1573
 - displaying peer information, 1574
 - displaying source active cache information, 1575, 1578
 - enabling, 1562
 - filtering source-group pairs, 1563
 - mesh groups, 1567
 - overview, 1558
 - passive route insertion, 1579
 - peer reverse path forwarding flooding, 1561
 - source active caching, 1561

Multi-Chassis Trunking (MCT)

- benefits, 827
- cluster client automatic configuration, 835
- cluster MAC types, 846
- configuration examples, 867
- creating a basic configuration, 837
- displaying information, 860
- dynamic LAGs, 846
- failover scenarios, 843
- feature interaction, 835
- for VRRP/VRRP-E, 857
- Layer 2 multicast failover, 852
- Layer 2 protocols, 850
- Layer 3 behavior, 854
- Layer 3 unicast, 855
- MDUP, 846
- MRP, 850
- multicast snooping, 851
- multicast snooping configuration example, 879
- overview, 827
- port loop detection, 849
- protocol-based VLANs, 850
- setting up cluster client automatic configuration, 841
- single level configuration example, 867
- single level extension configuration example, 867
- supported features, 827
- synching router MACs, 849
- terminology, 829
- two level configuration example, 871
- uplink switch, 851
- VRRP/VRRP-E configuration, 858
- multi-device port
 - RADIUS authentication, 1850
 - supported RADIUS attributes, 1851
- multi-device port authentication
 - 802.1x security on the same port, 1853
 - clearing hardware aging period for blocked MAC addresses, 1867
 - clearing MAC addresses, 1866
 - configuration considerations, 1862
 - configuring, 1855
 - configuring Brocade-specific attributes on RADIUS server, 1854
 - configuring dynamic VLAN assignment, 1858
 - defining MAC address filters, 1857
 - disabling aging for authenticated MAC addresses, 1866
 - displaying information, 1870
 - displaying MAC authentication table for FCX devices, 1878
 - dynamically applying IP ACLs, 1861
 - enabling denial of service (DoS) attack protection,

1864

- enabling source guard protection, 1864
- example configurations, 1879
- generating SNMP traps, 1857
- how it works, 1850
- limiting the number of MAC addresses, 1870
- password override, 1870
- specifying the aging time for blocked MAC addresses, 1868
- specifying the authentication-failure action, 1856
- specifying the MAC addresses, 1856
- specifying the RADIUS timeout action, 1868
- support for DHCP snooping, 1852
- support for dynamic ACLs, 1852
- support for dynamic ARP inspection, 1852
- support for dynamic VLAN assignment, 1852
- support for source guard protection, 1853
- viewing the ACL, 1865

Multiple Spanning Tree Protocol (MSTP)

- configuring additional parameters, 1174
- configuring an instance, 1175
- configuring bridge priority for an instance, 1176
- configuring mode and scope, 1171
- deleting a VLAN to MSTI mapping, 1173
- disabling on a port, 1178
- displaying information for a specific instance, 1183
- displaying statistics, 1181
- example configuration, 1179
- overview, 1169
- reconvergence, 1172
- regions, 1170
- setting point-to-point link, 1177
- setting ports to be operational edge ports, 1177
- setting the global parameters, 1176
- viewing the configuration digest, 1174

- multi-port static MAC address, configuring, 562

N

- negotiation mode, changing the speed, 59
- network connectivity testing, 95
- network routes
 - configuring default, 1000
- NIAP-CCEVS certification, 2021

O

Open Shortest Path First (OSPF)

- differences between version 2 and version 3, 1288
- overview, 1226
- point-to-point links, 1228

Operations, Administration, and Maintenance (OAM)

- overview, 71

- optical monitoring, viewing, 502

- optical transceiver thresholds, viewing, 504

- optical transceivers

- Syslog messages, 505

OSPF

- assigning a totally stubby area, 1238
- assigning an area range, 1241
- assigning areas, 1237
- assigning interfaces to an area, 1242
- assigning virtual links, 1247
- auth-change-wait-time, 1243
- authentication-key, 1243
- block flooding of outbound LSAs, 1245
- changing administrative distance, 1265
- changing the reference bandwidth, 1252
- changing the reference bandwidth for the cost, 1251
- changing the timer for authentication changes, 1245
- clearing information, 1270
- clearing information for areas, 1271
- clearing redistributed routes from the routing table, 1271
- clearing topology information, 1271
- CLI commands, 1272
- configuring group Link State Advertisement (LSA) pacing, 1266
- configuration, 1235
- configuring a non-broadcast interface, 1246
- configuring a point-to-point link, 1269
- configuring default route origination, 1263
- configuring external route summarization, 1262
- configuring graceful restart, 1269
- cost, 1243
- dead-interval, 1243
- defining redistribution filters, 1252
- designated router election in multi-access networks,

1228

- designated routers in multi-access networks, 1228
- disabling or re-enabling load sharing, 1260
- displaying ABR information, 1284
- displaying area information, 1275
- displaying data in an LSA, 1283
- displaying graceful restart information, 1285
- displaying information, 1272
- displaying interface information, 1276, 1278
- displaying link state information, 1282
- displaying route information, 1279
- displaying trap status, 1284
- displaying virtual link information, 1284
- displaying virtual neighbor information, 1283
- dynamic activation and configuration, 1233
- dynamic memory, 1234
- enabling on the router, 1236
- enabling route redistribution, 1258
- encrypted display of the authentication string or MD5 authentication key, 1244
- external LSA reduction, 1230
- global parameters, 1235
- graceful restart, 1234
- hello-interval, 1243
- interface parameters, 1236, 1242
- MD5-authentication activation wait time, 1243
- MD5-authentication key ID and key, 1243
- modifying interface defaults, 1242
- modifying SPF timers, 1264
- modifying the default metric for redistribution, 1258
- modifying the exit overflow interval, 1268
- modifying the redistribution metric type, 1265
- modifying the standard compliance setting, 1268
- modifying traps generated, 1266
- modifying virtual link parameters, 1249
- Not-So-Stubby-Area (NSSA), 1239
- preventing specific routes from being installed in the IP route table, 1255
- priority, 1243
- resetting, 1237
- RFC 1583 and 2178 compliance, 1230
- specifying Syslog messages to log, 1267
- transit-delay, 1244
- using a standard ACL as input to the distribution list, 1255

OSPF (IPv6)

- administrative distance, 1304
- assigning a totally stubby area, 1290
- assigning a virtual link source address, 1292
- assigning areas, 1290
- assigning interfaces to an area, 1291
- changing the key rollover timer, 1313
- changing the reference bandwidth for the cost on interfaces, 1294
- clearing IPsec statistics, 1314
- configuration, 1289
- configuring a default route origination, 1302
- configuring a distribution list, 1299
- configuring administrative distance based on route type, 1304
- configuring distribution list using a route map, 1301
- configuring external route summarization, 1298
- configuring IPsec, 1308
- configuring IPsec for a virtual link, 1312
- configuring IPsec for an area, 1311
- configuring route redistribution, 1295
- configuring the LSA pacing interval, 1305
- configuring virtual links, 1291
- default route origination, 1302
- disabling IPsec on an interface, 1313
- disabling or re-enabling event logging, 1307
- displaying database information, 1316
- displaying information, 1314
- displaying interface information, 1321
- displaying IPsec configuration for an area, 1337
- displaying IPsec for a virtual link, 1339
- displaying memory usage, 1324
- displaying neighbor information, 1325
- displaying redistributed routes, 1327
- displaying route information, 1328
- displaying SPF information, 1330
- displaying virtual link information, 1333
- displaying virtual neighbor information, 1333
- enabling, 1289
- external route summarization, 1298
- filtering routes, 1299
- implementing Internet Protocol Security (IPsec), 1307
- interface and area IPsec considerations, 1309
- IPsec examples, 1334
- link state advertisement types, 1288
- modifying external link state database limit, 1305
- modifying interface defaults, 1306
- modifying shortest path first timers, 1303
- modifying the default metric for redistributed routes, 1297
- modifying the exit overflow interval, 1305
- modifying the metric type for redistributed routes,

1297

- modifying virtual link parameters, 1293
- overview, 1287
- shortest path first timers, 1303
- showing IPsec policy, 1335
- showing IPsec statistics, 1336
- specifying the key rollover time, 1310

P

packet parameters, configuring, 973

packet types

- specifying a single source interface, 977

password

- enable read-only-password, 126

password logins, enabling, 186

passwords

- changing a local user password, 136
- configuring, 129
- configuring password history, 132
- creating a password option, 135
- enabling user password aging, 131
- enabling user password masking, 131
- enhanced login lockout, 132
- recovering from a lost password, 127
- setting a Telnet password, 124
- setting for management privilege levels, 125
- setting to expire, 133
- specifying a minimum password length, 128

passwords, used to secure access, 124

path MTU discovery, 1045

Per VLAN Spanning Tree (PVST)

- configuration examples, 1160
- configuring support, 1159
- displaying support information, 1160

Per VLAN Spanning Tree plus (PVST+)

- compatibility, 1157
- overview, 1157

PHY FIFO configuration, 53

PIM

- candidate rendezvous point (RP), 1535
- configuring rendezvous point (RP), 1536
- displaying flow cache information, 1554
- displaying resources, 1549
- enabling on the router and an interface, 1525
- initiating multicasts on a network, 1522
- modifying global parameters, 1526
- overview, 1558

PIM SM snooping

- configuration, 1476
- disabling for the VLAN, 1501
- disabling on a VLAN, 1477
- displaying errors, 1478
- displaying information, 1486
- displaying information for a specific group or source group pair, 1488
- displaying information on a Layer 2 switch, 1486
- enabling for the VLAN, 1501
- enabling on a VLAN, 1476
- enabling or disabling, 1476
- show commands, 1483, 1486
- show command
- show ip multicast, 1477

PIM SM snooping on FastIron WS and Brocade FCX and ICX switches

- application example, 1495
- configuration, 1497
- configuring for individual ports, 1502
- configuring for the VLAN, 1502
- configuring mode for a VLAN, 1501
- configuring report control, 1499
- configuring static router ports, 1503
- configuring the global mode, 1498
- configuring the global version, 1499
- configuring the hardware and software resource limits, 1498
- enabling or disabling error and warning messages, 1500
- enabling or disabling PIM sparse snooping, 1500
- enabling or disabling transmission and receipt of packets on a port, 1498
- global configuration tasks, 1497
- modifying the age interval, 1499
- modifying the multicast cache age time, 1500
- modifying the query interval, 1499
- modifying the wait time, 1500
- overview, 1495
- VLAN-specific configuration tasks, 1497

PIM SM traffic snooping

- application examples, 1465
- configuration notes and limitations, 1466
- configuring the software resource limits, 1468
- enabling globally on the device, 1469
- global tasks, 1467
- overview, 1465
- port-specific tasks, 1468
- VLAN-specific tasks, 1468

PIM snooping on FastIron WS and Brocade FCX and ICX switches

- displaying information, 1507

PIM Sparse

- boot strap router (BSR), 1535
- changing the join and prune message interval, 1538
- changing the shortest path tree (SPT) threshold, 1537
- configuration, 1533
- configuring interface parameters, 1535
- displaying information, 1545
- dropping traffic, 1539
- globally enabling and disabling, 1534
- limitations, 1534
- RP paths and SPT paths, 1533
- switch types, 1532

ping

- IPv6 address, 370

policy-based routing (PBR), 1747

- basic example, 1751
- enabling, 1751
- setting the next hop, 1752
- setting the output interface, 1753
- trunk formation, 1753

port

- assigning 802.1Q tagging, 568
- buffers and descriptors values, 596
- dynamic port assignment, 766
- eligibility for link aggregation, 723
- locking to restrict addresses, 574
- removing from a VLAN, 744
- support for 802.1Q in tagging, 757

port configuration

- assigning a port name, 40
- configuring maximum port speed advertisement, 44
- disabling or re-enabling a port, 46
- enabling port speed, 42
- enabling port speed down-shift, 43
- modifying port duplex mode, 45
- modifying port speed and duplex mode, 41

port default VLAN ID (PVID)

- displaying, 825
- displaying information, 825

port dual-mode VLAN

- displaying membership, 824

port flap dampening configuration, 61

port loop detection, 65

port mirroring and monitoring

- command syntax, 929
- configuration, 927
- number of ports supported, 927
- overview, 925

port monitoring

- monitoring an individual trunk port, 930

port regions on FastIron X series, 556

Port Security

- autosave, 1841

port statistics

- parameters, 524
- viewing, 523

port VLAN

- displaying membership, 824

port-based VLAN

- configuring uplink ports, 787
- modifying, 744
- removing, 744

port-based VLANs

- configuring, 740

port-based VLANs, enabling, 567

ports on demand (PoD)

- configuring on an interface, 208
- configuring the upper ports in a stack, 209
- displaying license configuration, 210

power class setting for PoE devices, 672

power level configuration for PoE, 671

power level for PoE devices, 670

Power over Ethernet (POE)

- and CPU utilization, 668
 - autodiscovery, 662
 - cabling requirements, 665
 - configuring power levels, 671
 - disabling support for power-consuming devices, 669
 - displaying information, 675
 - dynamic upgrade of power supplies, 663
 - enabling and disabling, 669
 - enabling the detection of power requirements, 670
 - endspan method, 660
 - installing firmware, 666
 - installing firmware on FCX platform, 667
 - installing firmware on FSC platform, 666
 - IP surveillance cameras, 666
 - methods for delivery, 660
 - midspan method, 661
 - overview, 659
 - power class, 662
 - power specifications, 663
 - resetting parameters, 675
 - setting the inline power priority for a port, 673
 - setting the maximum power level, 670
 - setting the power budget for interface module, 673
 - setting the power class, 672
 - supported firmware file types, 667
 - supported powered devices, 665
 - terminology, 660
 - Voice over IP, 666
 - voltage selection during bootup, 664
 - voltage selection when a PoE power supply is installed, 664
 - voltage selection when a PoE power supply is removed, 665
- private VLAN (PVLAN)
- configuration, 809
 - configuration notes, 812, 813
 - configuring an isolated or community, 815
 - difference between standard VLAN, 812
 - enabling broadcast or unknown unicast traffic, 816
- privilege levels, 126
- Protocol Independent Multicast (PIM) sparse, 1531
- protocols, 1516

Q

QoS

- 802.1p priority override, 1979
 - assigning static MAC entries to priority queues, 1978
 - behavior for 802.1p marking in an IronStack, 1971
 - behavior for Layer 2 (802.1p) in an IronStack, 1971
 - behavior for Layer 3 (DSCP) in an IronStack, 1971
 - behavior on port priority and VLAN priority in an IronStack, 1971
 - buffer allocation and thresholds, 1979
 - changing a port priority, 1978
 - changing the DSCP to internal forwarding priority mappings, 1984
 - changing the minimum bandwidth percentages, 1989
 - configuring the QoS queues, 1989
 - default mappings, 1969
 - default scheduling configuration for the ICX 6430, 1986
 - default scheduling configuration for the SX-F148GPP module, 1985
 - default values for FCX and ICX 6450 platforms, 1976
 - default values for ICX 6430 platforms, 1977
 - determining the trust level of a packet, 1964
 - displaying DSCP-based settings, 1992
 - displaying the user-configurable scheduler profile configuration, 1976
 - DSCP-based, 1981
 - enabling 802.1p priority override, 1980
 - mapping a VLAN priority to hardware forwarding queue, 1985
 - mapping configuration, 1982
 - marking, 1980
 - options for IP ACLs, 1740
 - priorities-to-traffic assignment, 1978
 - processing of classified traffic, 1964
 - profile restrictions, 1970
 - queues, 1971
 - queues for the ICX 6430 switch, 1973
 - queues for the SX-F148GPP interface modules, 1972
 - queuing methods, 1987
 - scheduling information, 1987
 - support on Brocade FastIron units, 1970
 - user-configurable scheduler profile, 1975
 - using ACLs to honor DSCP-based QoS, 1981
- QoS priorities
- dynamic buffer allocation, 585
- Quality of Service (QoS)
- overview, 1963
- queue depth limits, default, 585, 589

R

RADIUS

- AAA operations, 159
 - accounting configuration, 158
 - authentication configuration, 157
 - authentication method values, 168
 - authentication, authorization, and accounting (AAA), 157
 - authentication-method list examples, 175
 - authentication-method lists, 174
 - authorization configuration, 158
 - Brocade-specific attributes on the server, 161
 - command authorization and accounting for console commands, 170
 - configuration, 161
 - configuration considerations, 160
 - configuring accounting for CLI commands, 171
 - configuring accounting for system events, 172
 - configuring accounting for Telnet/SSH (Shell) access, 171
 - configuring an interface as the source for all packets, 172
 - configuring command authorization, 170
 - configuring enable authentication, 169
 - displaying configuration information, 172
 - entering privileged EXEC mode, 169
 - identifying the server to the Brocade device, 163
 - servers per port, 164
 - setting authentication-method lists, 167
 - setting over IPv6, 167
 - setting the key, 166
 - setting the retransmission limit, 166
 - setting the timeout parameter, 167
 - specifying different servers for individual AAA functions, 163
- RADIUS authorization, 169
- RADIUS parameters, 166
- RADIUS security, 157
- RADIUS server
- generic attributes, 910

Rapid Spanning Tree Protocol (RSTP)

- bridge port states, 1113
 - changing bridge parameters, 1139
 - changing port parameters, 1140
 - configuring parameters on a Brocade device, 1137
 - convergence after a link failure, 1129
 - convergence at link restoration, 1130
 - convergence in a complex topology, 1131
 - convergence in a simple topology, 1125
 - displaying information, 1141
 - edge ports and edge port roles, 1112
 - overview, 1109
 - point-to-point ports, 1113
 - propagation of topology change, 1133
 - recommended path cost values, 1140
- rate limiting
- configuring ACL-based for FastIron X series, FCX, and ICX series switches, 1950
 - configuring for FastIron X series, FCX, and ICX series switches, 1949
 - displaying information for FastIron X series, FCX, and ICX series switches, 1951
 - fixed on inbound port configuration, 1957
 - fixed rate limiting, 1948
 - hardware for FastIron X series, FCX, and ICX series switches, 1948
 - hardware on FastIron WS series switches, 1956
 - overview for FastIron X series, FCX, and ICX series switches, 1947
- rate limiting on FastIron WS series switches
- ACL-based policy configuration, 1960
 - configuration notes, 1957
 - configuration syntax, 1957
 - configuring port-based, 1959
 - displaying configuration, 1960
 - how it works, 1956
 - minimum and maximum outbound rate limits, 1958
 - outbound port configuration, 1958
 - overview, 1955
- rate shaping
- configuring outbound for a port, 1953
 - configuring outbound for a specific priority, 1953
 - displaying configurations, 1954
 - for FastIron X series, FCX, and ICX series switches, 1952
- rconsole, 274
- remote access restrictions, 115
- remote fault notification (RFN)
- enabling and disabling, 604
 - on fiber connections, 603

- rendezvous point (RP)
 - ACL-based assignment, 1539
 - anycast method, 1540
 - configuring an ACL-based assignment, 1539
 - designating as an interface IP address, 1563
 - displaying information, 1550
- restrict mode access
 - using ACL, 112
- restricting
 - HTTP and HTTPS connection, 117
 - SNMP access to a specific VLAN, 119
 - snmp-server enable vlan, 119
 - SSH connection, 117
 - Telnet access to a specific VLAN, 119
 - Telnet connection, 117
 - TFTP access to a specific vlan, 120
 - Web management access to a specific VLAN, 119
- restricting access to device based on IP or MAC address, 117
- Reverse Address Resolution Protocol (RARP)
 - changing the maximum number of supported entries, 1008
 - configuration, 1007
 - creating static entries, 1008
 - disabling, 1008
 - how it differs from BootP and DHCP, 1007
- RIP
 - suppression of advertisements, 1674
- RMON
 - alarm (group 3), 534
 - event (group 9), 535
 - export configuration and statistics, 533
 - history (group 2), 534
 - maximum number of entries allowed in control table, 531
 - statistics, 532
- RMON support, 531
- route, 1404
- route learning
 - configuring, 1218
 - with Routing Information Protocol (RIP), 1207
- route map
 - configuring, 1750

- Routemap
 - match, 1405
 - match as-path, 1406
 - match community, 1406
 - match community exact-match, 1408
 - match ip address, 1406
 - match ip address prefix-list, 1406
 - match ip next-hop, 1407
 - match ip next-hop prefix-list, 1407
 - match ip route-source, 1407
 - set, 1408
 - set comm-list delete, 1411
 - set ip next-hop peer-address, 1410
 - set metric-type internal, 1410

Router

- activate, 1700
- address-filter, 1394
- aggregate-address, 1384
- always-compare-med, 1377
- area, 1238, 1239, 1248, 1249, 1290, 1291, 1293
- area | nssa | default-information-originate, 1240
- area | range, 1241
- area virtual-link, 1292
- as-path-filter, 1396
- as-path-ignore, 1376
- auto-cost reference-bandwidth, 1252, 1295, 1305
- bgp-redistribute-internal, 1393
- bsr-candidate ethernet, 1535
- client-to-client-reflection, 1381
- cluster-id, 1380
- community-filter, 1399
- compare-routerid, 1377
- confederation identifier, 1383
- confederation peers, 1383
- dampening, 1415
- database-overflow-interval, 1268
- default-gateway, 1585
- default-information-originate, 1264, 1302, 1371
- default-local-preference, 1370
- default-metric, 1206, 1258, 1371
- deny | permit redistribute, 1254
- disable-dvmrp, 1583, 1587
- disable-pim, 1526, 1534
- distance, 1204, 1265, 1304, 1375
- distribute-list, 1256
- distribute-list prefix-list, 1299
- distribute-list prefix-list in | out, 1220
- distribute-list route-map, 1301
- enforce-first-as, 1376
- fast-external-fallover, 1366
- filter permit | deny, 1210
- graft-retransmit-time, 1585
- graft-retransmit-timer, 1528
- hardware-drop, 1579
- hello-timer, 1527
- inactivity-timer, 1528
- interface ethernet, 1193
- ip-address, 1700
- ipv6 router ospf, 1289
- ipv6 router rip, 1216
- learn-default, 1208
- local-as, 1353, 1383
- log, 1268
- log-status-change, 1307
- maximum-paths, 1368
- med-missing-as-worst, 1378

- message-interval, 1538
- metric-type, 1265, 1297
- multipath, 1368
- nbr-timeout, 1527, 1584
- neighbor, 1354, 1363
- neighbor distribute-list, 1402
- neighbor password, 1360
- neighbor peer-group, 1363, 1364
- neighbor permit | deny, 1208
- neighbor route-reflector-client, 1380
- neighbor shutdown, 1365
- neighbor soft-reconfiguration inbound, 1452
- neighbor unsuppress-map, 1419
- network, 1369
- next-hop-enable-default, 1370
- next-hop-recursion, 1374
- offset-list in | out offset, 1204
- permit | deny redistribute, 1195, 1205
- poison-local-routes, 1221
- poison-reverse, 1221
- probe-interval, 1585
- prune-age, 1584
- prune-timer, 1527
- prune-wait, 1528
- readvertise, 1393
- redistribute connected, 1391
- redistribute bgp, 1295, 1296
- redistribute bgp | connected | isis | ospf | static, 1220
- redistribute connected, 1391
- redistribute ospf, 1392
- redistribute rip, 1392
- redistribute static, 1393
- redistribution, 1195, 1207, 1254, 1259
- report-interval, 1585
- rfc1583-compatibility, 1268
- route-discard-timeout, 1584
- route-expire-timeout, 1584
- router vrrp, 1700
- rp-address, 1537, 1589
- rp-candidate add, 1536
- rp-candidate delete, 1536
- rp-candidate ethernet, 1536, 1591
- slow-start, 1679
- snmp-server trap ospf, 1267
- spt-threshold infinity, 1538
- ssm-enable, 1600
- summary address, 1241
- summary-address, 1262, 1298
- timers, 1217
- timers keep-alive hold-time, 1365
- timers lsa-group-pacing, 1266, 1305
- timers spf, 1264, 1303

- trigger-interval, 1585
- update-time, 1207, 1366
- use-vrrp-path, 651, 1209, 1675
- virtual-link-if-address interface ethernet, 1292
- router ID, changing, 976
- routing between VLANs (Layer 3 only), 765
- Routing Information Protocol (RIP)
 - applying route filter to an interface, 1210
 - changing the administrative distance, 1204
 - changing the cost of routes learned on a port, 1203
 - changing the redistribution metric, 1206
 - changing the route loop prevention method, 1196, 1209
 - configuration, 1193
 - configuring a neighbor filter, 1208
 - configuring a redistribution filter, 1195
 - configuring an offset list, 1203
 - configuring metric parameters, 1203
 - configuring redistribution, 1205
 - configuring redistribution filters, 1205
 - configuring route filters, 1210
 - denying route advertisements, 1208
 - disabling poison-reverse, 1209
 - displaying filters, 1211
 - enabling, 1193, 1202
 - enabling learning of default routes, 1196
 - enabling redistribution, 1195, 1206
 - enabling redistribution of IP static routes, 1193
 - global parameters, 1200
 - interface parameters, 1201
 - overview, 1199
 - parameter configuration, 1202
 - parameters and defaults, 1200
 - removing a redistribution deny filter, 1207
 - route learning and advertising parameters, 1207
 - suppressing route advertisement on a VRRP or VRRP-E backup interface, 1209
- Routing Information Protocol Next Generation (RIPng)
 - clearing routes from the IPv6 route table, 1221
 - configuration, 1216
 - configuring poison reverse parameters, 1221
 - controlling distribution of routes, 1220
 - displaying configuration, 1221
 - displaying routing table, 1222, 1223
 - enabling, 1216
 - overview, 1215
 - redistributing routes, 1219
 - route learning and advertising parameters, 1218
 - timers, 1217
- routing protocols
 - enabling or disabling, 1196

RSA authentication

- configuring challenge-response authentication, 183
- enabling challenge-response, 185
- exporting client public keys, 196
- generating a client key pair, 196
- generating and deleting a key pair, 182
- importing public keys into Brocade device, 184
- providing the public key to clients, 183

RSTP

- assignment of port roles, 1110
- bridges and bridge port roles, 1109
- changes to port roles and states, 1114
- edge port and non-edge port states, 1114

S

SCP

- and IPv6, 417

search string, using special characters, 9

secure access

- passwords, 124

secure copy (SCP)

- configuration notes, 191
- copying a software image file from flash memory, 193
- enabling and disabling, 191
- example file transfers, 191
- importing a digital certificate, 194
- importing a DSA or RSA public key, 194
- importing an RSA private key, 194
- using to install a software license, 221
- with SSH2, 191

secure management access to Brocade devices, 110

secure shell (SSH)

- and IPv6, 367, 417

- overview, 179

secure shell (SSH2)

- authentication types, 181
- clients, 180
- configuration, 181
- enabling and disabling with host keys, 181
- optional parameters, 185
- supported features, 180
- unsupported features, 180

secure sockets layer (SSL), 136

secure-setup

- troubleshooting in stack topologies, 324

security

- AAA for RADIUS commands, 160
- AAA operations for RADIUS, 159
- allowing SNMP access to Brocade device, 121
- allowing SSHv2 access to Brocade device, 121
- allowing Web management through HTTP for Brocade device, 121
- allowing Web management through HTTPS, 122
- authentication method values, 150
- changing the SSL server certificate key size, 137
- deleting the SSL certificate, 139
- device management, 121
- edge port, 177
- edge ports, 176
- enabling SSL server on Brocade device, 137
- generating an SSL certificate, 138
- RADIUS, 157
- specifying a port for SSL communication, 137
- SSL for the Web management interface, 136
- support for SSL digital certificates, 138
- TACACS and TACACS+, 139
- TACACS authentication, 142
- TACACS+ accounting, 143
- TACACS+ authorization, 142
- TCP flags, 176, 177

sFlow

- and 802.1x port security, 1798
- and CPU utilization, 537
- and hardware support, 537
- and port monitoring, 538
- and sampling rate, 538
- and source address, 537
- and source port, 538
- changing the polling interval, 541
- changing the sampling rate, 541
- changing the source port, 545
- clearing statistics, 552
- command syntax for sFlow forwarding, 545
- configuration considerations, 537
- configuring and enabling, 539
- configuring version 5 features, 546
- displaying information, 549
- enabling forwarding, 545
- exporting CPU and memory usage information, 548
- extended gateway information, 537
- extended router information, 536
- IPv6 packet sampling, 537
- overview, 535
- specifying the collector, 539
- specifying the maximum flow sample size, 548
- specifying the polling interval, 548
- specifying the version used for exporting sFlow data,

- 547
- support for IPv6 packets, 536
- uplink utilization list configuration, 552
- utilization list for an uplink port command syntax, 553
- version 5, 536
- show, 1420, 1642, 1687

show command

- ip ospf database external-link-state advertise, 1283
- ipv6 inter tunnel, 380
- show 802-1w, 1141
- show aaa, 156, 172
- show access-list, 1727, 1745
- show access-list accounting traffic-policy, 1785
- show access-list all, 1746
- show arp, 1070, 1081, 1931
- show authenticated-mac-address, 1866
- show auth-mac-address, 1871
- show boot-preference, 81
- show clock, 32
- show cluster, 860
- show cluster ccp peer, 862
- show cluster client, 861
- show configuration, 589
- show default value, 1191
- show default values, 578
- show dir, 77
- show disabled-multicast-to-cpu, 1592
- show dot1x, 1821
- show dot1x mac-address-filter, 1826
- show dot1x mac-session, 1831
- show dot1x statistics, 1824
- show errdisable recovery, 1168
- show fdp entry, 441, 445
- show fdp neighbor, 439
- show fdp neighbors, 444
- show fdp traffic, 445
- show flash, 74
- show gvrp, 894
- show gvrp statistics, 898
- show inline power, 675
- show inline power detail, 678
- show interface, 48, 530, 694, 1825
- show interface ethernet, 862, 1102
- show interfaces stack-ports, 308
- show interfaces tunnel, 379
- show ip, 1034, 1080
- show ip access-list, 1727
- show ip address, 1035
- show ip arp inspection, 1931
- show ip bgp, 1444
- show ip bgp attribute-entries, 1447
- show ip bgp config, 1425
- show ip bgp filtered-routes, 1453
- show ip bgp flap-statistics, 1420, 1449
- show ip bgp neighbors, 1413, 1427, 1436, 1451,

1454, 1456

- show ip bgp peer-group, 1439
- show ip bgp route, 1372, 1374
- show ip bgp route o, 1389
- show ip bgp routes, 1441
- show ip bgp routes best, 1442
- show ip bgp routes not-installed-best, 1443
- show ip bgp routes summary, 1440
- show ip bgp routes unreachable, 1443
- show ip bgp summary, 1422
- show ip cache, 1073
- show ip client-pub-key, 185
- show ip dhcp relay information, 1941
- show ip dhcp snooping, 1935
- show ip dhcp-server address-pool, 1025
- show ip dhcp-server binding, 1024
- show ip dhcp-server flash, 1026
- show ip dhcp-server summary, 1027
- show ip dvmrp rpf, 1587
- show ip igmp group, 1601
- show ip igmp group tracking, 1599
- show ip igmp interface, 1603
- show ip igmp traffic, 1604, 1606
- show ip interface, 1060, 1069
- show ip interface tunnel, 1060
- show ip msdp peer, 1574
- show ip msdp sa-cache, 1575, 1578
- show ip msdp summary, 1573
- show ip multicast error, 1478, 1505
- show ip multicast group, 1478, 1479, 1505
- show ip multicast mcache, 1480, 1507
- show ip multicast pimsm-snooping, 864, 1486, 1488,

1507

show ip multicast resource, 1480, 1508
show ip multicast traffic, 1481, 1509
show ip multicast vlan, 1477, 1482, 1483, 1510
show ip ospf area, 1275
show ip ospf border-routers, 1284
show ip ospf config, 1272
show ip ospf database external-link-state, 1281
show ip ospf database link-state, 1282
show ip ospf interface, 1247, 1276, 1278
show ip ospf neighbor, 1276
show ip ospf neighbors, 1285
show ip ospf redistribute route, 1280
show ip ospf routes, 1279
show ip ospf trap, 1284
show ip ospf virtual-link, 1284
show ip ospf virtual-neighbor, 1283
show ip pim bsr, 1547
show ip pim error, 1557
show ip pim flow, 1063, 1554
show ip pim group, 1547
show ip pim interface, 1062
show ip pim mcache, 1063, 1555, 1580
show ip pim mcachePIM
 displaying multicast cache information, 1555
show ip pim nbr, 1063
show ip pim neighbor, 1553
show ip pim prune, 1531
show ip pim resource, 1549
show ip pim rp-candidate, 1550
show ip pim rp-hash, 1551
show ip pim rp-map, 1540, 1551, 1590
show ip pim rp-set, 1540, 1552, 1589
show ip pim sparse, 1545
show ip pim traffic, 1556
show ip pimsm-snooping, 1487
show ip pimsm-snooping vlan, 1487
show ip rip, 1211
show ip route, 1060, 1074, 1188, 1448
show ip route static, 1389
show ip route summary, 1076
show ip ssh, 188
show ip ssh config, 189
show ip static-arp, 1072
show ip traffic, 1077, 1082
show ip tunnel traffic, 1061
show ip vrrp brief, 1684
show ip vrrp stat, 1692
show ip vrrp vrid, 1690
show ipc, 108
show ipc_stat, 108
show ipsec policy, 1335
show ipsec sa, 1334
show ipsec statistics, 1314, 1336
show ipv6, 382
show ipv6 access-list, 1759, 1770
show ipv6 cache, 397
show ipv6 interface, 398
show ipv6 mld-snooping, 1645
show ipv6 mld-snooping error, 1642
show ipv6 mld-snooping group, 1643
show ipv6 mld-snooping mcache, 1644
show ipv6 mld-snooping traffic, 1646
show ipv6 mld-snooping vlan, 1647
show ipv6 neighbor, 400
show ipv6 ospf area, 1315, 1337
show ipv6 ospf database, 1316
show ipv6 ospf database extensive, 1318
show ipv6 ospf interface, 1321, 1338
show ipv6 ospf memory, 1324
show ipv6 ospf neighbor, 1325
show ipv6 ospf neighbor router-id, 1326
show ipv6 ospf redistribute route, 1328
show ipv6 ospf route, 1299, 1300, 1301
show ipv6 ospf routes, 1328
show ipv6 ospf spf node area, 1330
show ipv6 ospf virtual-link, 1333, 1339
show ipv6 ospf virtual-neighbor, 1333
show ipv6 rip, 1221
show ipv6 rip route, 1222, 1223
show ipv6 route, 401
show ipv6 router, 402
show ipv6 tcp connections, 403
show ipv6 tcp status, 405
show ipv6 traffic, 407
show ipv6 tunnel, 378
show ipv6 vrrp, 1696
show ipv6 vrrp brief, 1684
show ipv6 vrrp-extended, 1697
show ipv6 vrrp-extended brief, 1685
show license, 210
show link-agg, 735
show link-aggregate, 731
show link-aggregate ethernet, 728
show link-error-disable, 63
show link-keepalive, 688
show lldp, 488
show lldp local-info, 472, 474, 483, 486, 488, 493
show lldp neighbors, 491
show lldp neighbors detail, 492
show lldp statistics, 463, 490
show lldp statisticsLLDP
 displaying statistics, 489
show local-userdb, 1919

- show local-userdb test, 1920
- show log, 1731
- show logging, 508, 510, 511, 921
- show loop-detection resource, 69
- show loop-detection status, 68
- show mac, 567, 717
- show mac cluster, 847
- show mac-address, 561, 567, 920
- show media, 57
- show media slot, 501
- show metro, 628, 629
- show mstp, 1181
- show mstp config, 1174
- show notification mac-movement threshold-rate, 576
- show optic, 502
- show optic slot, 503
- show optic threshold, 504
- show pod, 210
- show port security ethernet, 1844, 1847
- show port security mac, 1844
- show port security statistics, 1845
- show process cpu, 901, 1067, 1069, 1097, 1098,
1212, 1213, 1274, 1426, 1695
- show pvlan, 825
- show qd-buffer-profile, 598
- show qd-share-level, 602
- show qos-profiles
 - QoS

- displaying settings, 1992
- show qos-tos, 1984, 1992
- show rate-limit broadcast, 36
- show rate-limit fixed, 1951, 1961
- show rate-limit fixed input, 1960
- show rate-limit output-shaping, 1954
- show rate-limit unknown-unicast, 36
- show relative-utilization, 553, 554
- show reserved-vlan-map, 767
- show rmon statistics, 532
- show route-map, 1450
- show run, 131, 1035
- show run interface, 36
- show run interface ethernet, 1980
- show sflow, 542, 550
- show snmp engineid, 426, 434
- show snmp group, 435
- show snmp server, 372, 425, 433
- show snmp user, 435
- show snmp associations, 25
- show snmp status, 27
- show span, 525, 864, 1095, 1152
- show span detail, 1099
- show span detail vlan, 1102
- show span fast-uplink-span, 1108
- show span pvst-mode, 1160
- show span vlan, 610, 1109
- show spanning-tree root-protect, 1166
- show stack, 210, 271, 277, 336, 341, 350
- show stack flash, 296
- show stack neighbors, 303
- show stack rel-ipc stats unit, 301
- show stack rel-ipc-stats, 297
- show stack stack-ports, 304
- show statistics, 1063
- show statistics dos-attack, 1926
- show statistics ethernet, 523
- show statistics stack-ports, 309
- show statistics tunnel, 1061
- show stp-protect, 1093
- show symmetric-flow-control, 52
- show table, 1878
- show table-mac-vlan, 915, 919, 923
- show table-mac-vlan denied-mac, 917
- show topology-group, 610, 629
- show transmit-counter profiles, 528
- show transmit-counter values, 528
- show trunk, 716, 734
- show version, 72, 307, 522
- show vlan, 821, 1099
- show vlan brief, 823
- show vlan-group, 793

- show vlans, 822
- show voice-vlan, 61
- show vsrp aware, 654
- show vsrp vrid, 652, 655
- show web connection, 157
- show webauth, 1915
- show webauth blocked-list, 1919
- show webauth vlan, 1920
- show who, 190
- show-traffic policy, 1786
- snmp server-mode, 29
- span vlan, 525
- show commands
 - show notification mac-movement interval-history, 577
- Simple Network Time Protocol (SNTP), 24
- Single Spanning Tree Protocol (SSTP)
 - defaults, 1151
 - displaying information, 1152
 - enabling, 1151
 - overview, 1151
- Smurf attack protection, 1921
- SNMP
 - community strings, 422
 - configuration for an IronStack, 315
 - configuring V3 over IPv6, 367
 - configuring version 3 on Brocade devices, 426
 - defining a group, 427
 - defining a user account, 428
 - defining the engine ID, 426
 - defining the UDP port for traps, 431
 - defining views, 429
 - disabling traps, 21, 23, 24
 - displaying community string, 128
 - displaying groups, 435
 - displaying the community strings, 424
 - displaying the engine ID, 434
 - displaying user information, 435
 - enabling to configure RADIUS, 163
 - encryption of community strings, 422
 - engine IDs for stackable devices, 315
 - generating traps for multi-device port authentication,

- 1857
- interpreting varbinds in report packets, 435
- IPv6 support, 433
- Layer 2 generated traps, 21
- Layer 3 generated traps, 22
- over IPv6, 417
- overview, 421
- restricting access to an IPv6 node, 366
- setting the trap holdown time, 21
- specifying a single trap source, 20
- specifying a trap receiver, 19
- specifying an IPv6 host as trap receiver, 433
- specifying an IPv6 trap receiver, 367
- trap MIB changes, 432
- user-based security model, 425
- using to configure MAC-based VLANs, 915
- using to save and load configuration information, 91
- using to upgrade software, 80
- v3 configuration examples, 436
- version 3 traps, 430
- viewing IPv6 server addresses, 372, 433
- SNMP parameter configuration, 19
- SNTP
 - configuring Brocade device to function as server, 28
 - configuring over IPv6, 367
 - displaying server information, 29
 - enabling broadcast mode for client, 30
 - over IPv6, 417
 - specifying a server, 24
- software
 - viewing packages installed in the device, 232
- software image files, 76
- software license
 - non-licensed features, 202
 - overview, 200
 - supported software packages, 206
 - terminology, 199
 - transferring, 226
 - types, 201
 - viewing from the Brocade software portal, 224
 - viewing information about, 227
 - viewing the license database, 229
- software licensing
 - configuration tasks, 216
 - how it works, 200
 - installing a license file, 221
 - obtaining a license, 216
 - replacing for legacy devices, 226
 - rules and caveats, 206
 - seamless transition for legacy devices, 200, 201
 - trial license expiration, 224
 - using a trial license, 223
- software reboot, 81
- software upgrade, 80
- spanning tree
 - enabling on a VLAN, 768
- Spanning Tree Protocol (STP)
 - changing bridge and port parameters, 1091
 - changing port parameters, 1092
 - clearing BPDU drop counters, 1093
 - compatibility of 802.1W with 802.1D, 1133, 1136
 - configuration example for load sharing, 1155
 - configuring a fast uplink port group, 1107
 - configuring fast uplink span within a VLAN, 1108
 - configuring per VLAN group, 1154
 - default bridge parameters, 1088
 - default port parameters, 1089
 - disabling and re-enabling fast port span, 1104
 - displaying detailed information for each interface, 1099
 - displaying information, 1094
 - displaying information for a single port in a specific VLAN, 1102
 - displaying information for an entire device, 1095
 - displaying state information for an individual interface, 1102
 - displaying the root guard by VLAN, 1167
 - displaying the root guard state, 1166
 - displaying the state of a port-based VLAN, 1098
 - enabling or disabling, 1089
 - enabling or disabling globally, 1090
 - enabling or disabling in a port-based VLAN, 1090
 - enabling or disabling on an individual port, 1090
 - enabling protection, 1093
 - enabling root guard, 1166
 - excluding specific ports from fast port span, 1105
 - feature configuration, 1103
 - load balancing, 1154
 - overview, 1087
 - parameters and defaults, 1088
 - per VLAN group, 1153
 - protection enhancement, 1092
 - reconvergence time, 1149
 - root guard
 - overview, 1166
 - standard parameter configuration, 1088
 - viewing protection configuration, 1093
- spanning tree protocol (STP)
 - overview, 758
- special characters used in search string, 9
- SSH
 - configuring maximum idle time, 187
 - designating an interface as the source for all packets,

- 187
- displaying information, 188
- filtering access using ACLs, 188
- setting login timeout value, 187
- setting port number, 187
- terminating an active connection, 188
- using to install a software license, 221
- SSH authentication
 - setting the number of retries, 186
- SSH keys
 - CPU priority for generation, 182
- SSH2
 - configuration, 181
 - DSA challenge-response authentication, 181
 - password authentication, 181
 - RSA challenge-response authentication, 181
 - use with secure copy, 191
- SSH2 client
 - configuring public key authentication, 195
 - displaying information, 197
 - enabling, 195
 - overview, 195
 - using, 196
- SSL
 - changing the server certificate key size, 137
 - deleting the certificate, 139
 - enabling on Brocade device, 137
 - generating a certificate, 138
 - importing digital certificates and RSA private key files, 138
- stack unit
 - priority, 255, 278
- startup configuration, 83
- static ARP
 - adding an entry, 1187
- static IP route
 - adding, 1186
- static IP routes
 - configuring to the same destination, 997
- static IPv6 route configuration, 373
- static IPv6 route parameters, 375
- static MAC address
 - configuration, 561
 - multi-port, 561
 - VLAN-based configuration, 562
- static route
 - adding to the destination network, 992
 - and port states, 993
 - configuration, 991
 - configuring, 993
 - configuring to a tunnel destination, 1054
 - IP parameters, 992
 - types, 991
- static router ports, configuring, 1473
- statistics
 - clearing, 526
 - displaying virtual routing interface, 22
 - enabling SNMP VE, 22
- STP
 - bridge parameters, 769
 - modifying bridge and port parameters, 558
 - port parameters, 769
- STP feature
 - active uplink port failure, 1106
 - fast port span, 1103
 - fast uplink span, 1106
 - fast uplink span rules for trunk groups, 1107
 - switchover to the active uplink port, 1106
- STP Group
 - master-vlan, 1155
 - member-group, 1155
 - member-vlan, 1155
- STP statistics, viewing, 525
- Subnet
 - ip-subnet, 771
 - ipx-network, 771
- super aggregated VLAN
 - configuration, 796
- super aggregated VLANs, 763
- Syslog
 - changing the log facility, 516
 - changing the number of entries the local buffer can hold, 515
 - clearing log entries, 512
 - clearing messages from the local buffer, 518
 - CLI display of buffer configuration, 511
 - disabling, 24
 - disabling logging of a message level, 515
 - disabling or re-enabling, 514
 - displaying interface names in messages, 517
 - displaying messages, 508
 - displaying real-time messages, 510
 - displaying TCP or UDP port numbers in messages, 517
 - displaying the configuration, 510
 - enabling real-time display for a Telnet or SSH session,

- 509
- enabling real-time display of messages, 509
- message due to disabled port in loop detection, 70
- message for hitless management events, 107
- message types, 1995
- messages for CLI access, 23
- messages for hardware errors, 518
- messages for hitless stacking failover and switchover, 351
- messages for port flap dampening, 65
- messages for VRRP-E authentication, 1672
- messages on a device with the onboard clock set, 513
- messages related to GRE IP tunnels, 1048
- messages supported for software-based licensing, 227
- overview, 508
- retaining messages after a soft reboot, 518
- service configuration, 510
- specifying a server, 514
- specifying an additional server, 514
- static and dynamic buffers, 511
- STP messages, 1169
- time stamps, 512
- Syslog messages
 - disabling, 23
- system clock
 - setting, 30
- system management
 - basic, 521
 - viewing information, 521
- system parameter settings
 - default settings, 578
 - displaying, 578
 - displaying and modifying, 578
- system parameters
 - modifying default values, 584
- system reload scheduling, 92

T

TACACS

- authentication, 142
- enabling, 146

TACACS and TACACS+

- authentication, authorization, and accounting, 139
- configuration, 145
- configuration considerations, 145
- configuring an interface for all packets, 155
- configuring authentication-method lists, 149
- configuring for devices in a Brocade IronStack, 140
- how they differ, 139
- identifying servers, 146
- security, 139
- setting optional parameters, 147
- setting the retransmission limit, 148

TACACS+

- accounting, 143
- accounting configuration, 154
- authorization, 142
- configuring authorization, 151
- prompts when server is unavailable, 151
- setting the key, 148
- specifying servers for individual AAA functions, 147

TCAM

- entries in FWS devices, 1186

TCAM space on FCX devices, 394

TCP flags, 176, 177

TCP flags and edge port security, 1739

Telnet

- cancelling an outbound session, 24
- designated VLANs to Layer 2 switch, 120

testing network connectivity, 95

Test-Route

- set ip next hop, 1750

TFTP

- server file upload, 90

- transfer remedies, 93

TFTP server, configuring, 1024

thresholds

- changing XON and XOFF, 51

- XON and XOFF, 50

Time to Live (TTL)

- changing the threshold, 987, 1039

Topology Group

- master-vlan, 609

- member-group, 609

- member-vlan, 609

topology group

- configuration, 609

- configuration considerations, 608

- control ports and free ports, 608

- displaying information, 610

- master VLAN and member VLANs, 608

- overview, 607

- total transmit queue depth limit, configuring, 585
- trace route
 - using DNS to initiate, 972
- tracing an IPv4 route, 96
- traffic counters
 - displaying enhanced statistics, 528
 - for outbound traffic, 526
- traffic policies
 - configuration notes and feature limitations, 1774
 - CoS parameters for packets, 1775
 - CPU rate-limiting, 1787
 - enabling ACL statistics, 1783
 - maximum number supported on a device, 1775
 - overview, 1773
 - setting the maximum number on a Layer 3 device, 1776
 - using ACL-based rate limiting, 1776
 - viewing, 1786
- Trunk
 - config-trunk-ind, 712, 714
 - disable, 713
 - disable ethernet, 712, 714
 - enable ethernet, 713
 - port-name, 712
 - sflow-subsampling ethernet, 544
 - threshold, 715
- trunk group
 - additional trunking options, 711
 - application diagram, 698
 - assigning ports, 768
 - configuration, 707
 - configuration examples, 702
 - configuration notes for FastIron devices in an IronStack, 702
 - configuration rules, 699
 - configuring a multi-slot trunk group with one port per module, 710
 - configuring a static trunk group for devices in an IronStack, 711
 - configuring a trunk group of 10 Gbps Ethernet ports, 710
 - configuring a trunk group that spans two Ethernet modules, 709
 - configuring FastIron X series devices, 722
 - configuring stackable devices in an IronStack, 721
 - configuring trunk groups, 709
 - connectivity to a server, 698
 - deleting static, 714
 - displaying configuration information, 716
 - effects of changing VLAN membership of a port, 725
 - LACP configuration example, 719
 - LACP examples, 719
 - load sharing, 705
 - load sharing on FastIron stackable devices, 706
 - load sharing on FastIron X series devices, 706
 - overview, 697
 - ports and VLAN membership, 763
 - specifying the minimum number of ports, 714
 - support for flexible membership, 704
 - tearing down an aggregate link, 722
 - trunk port eligibility in an aggregate link, 723
 - viewing the first and last ports, 717
- trunk hash, adding Layer 2 information, 706
- trunk port
 - changing the sampling rate, 544
 - configuring outbound rate shaping, 715
 - disabling or re-enabling, 712
 - enabling sFlow forwarding, 715
 - monitoring, 715
 - naming, 712
 - setting the sFlow sampling rate, 716
- Tunnel
 - ipv6 address, 378
 - ipv6 enable, 377
 - tunnel destination, 377
 - tunnel mode ipv6ip, 377
 - tunnel source, 377
- tunnel interface
 - applying an ACL or PBR on a FastIron X series module, 1053
 - applying an ACL or PBR on the SX-F148GPP interface module, 1054
 - changing the MTU value, 1055
 - configuring a tunnel loopback port, 1053
 - configuring an IP address, 1054
 - configuring the destination address, 1052
 - configuring the source address or source interface, 1051
 - creating, 1051
 - enabling GRE encapsulation, 1052

U

UDLD

- changing the keepalive interval, 688
- changing the keepalive retries, 688
- configuration notes and feature limitations, 686
- creating a protected link group and assigning an active port, 692

UDP

- configuring broadcast parameters, 1009
- enabling forwarding for an application, 1010

unicast limiting command syntax, 33

Uni-Directional Link Detection (UDLD)

- clearing statistics, 690
- displaying information, 688
- displaying information for a single port, 689
- enabling, 686
- enabling for tagged ports, 687
- for tagged ports, 686
- overview, 685
- protected link groups, 691

user accounts

- defining local, 129
- local configuration, 133
- local with encrypted passwords, 135
- local with no passwords, 134
- local with unencrypted passwords, 134

user authentication, deactivating, 186

username

- configuration, 129

V

viewing system information, 521

virtual cable testing, 497

Virtual Router Redundancy Protocol (VRRP)

- additional configuration, 1670
- architectural differences between VRRP-E, 1659
- authentication, 1654
- authentication types, 1671
- backup preempt configuration, 1678
- basic configuration, 1664
- changing the timer scale, 1678
- clearing statistics, 1694
- comparison to VRRP-E, 1658
- configuration considerations for IPv6 version 3, 1667
- configuration examples, 1698
- configuring a backup for IPv4, 1666
- configuring a backup for IPv6, 1666
- configuring the Hello interval, 1675
- configuring the owner for IPv6, 1665
- dead interval configuration, 1676
- disabling, 1663
- displaying information, 1684
- displaying information for IPv6, 1697
- displaying statistics, 1692
- forcing a master router to abdicate to a standby router, 1683
- hello messages, 1653
- master negotiation, 1653
- MCT configuration, 858
- overview, 1650
- parameters, 1660
- router type, 1673
- suppression of RIP advertisements, 1654
- track port configuration, 1677
- track ports and track priority, 1654
- track priority configuration, 1677
- Virtual Router ID (VRID), 1652
- virtual router IP address, 1652
- virtual router MAC address, 1652
- with MCT, 857

Virtual Router Redundancy Protocol Extended (VRRP-E)

- MCT configuration, 858
- overview, 1655
- with MCT, 857

virtual routing interface, 759

- configuring, 791

virtual routing interface group

- configuring, 793
- displaying, 794

virtual routing interface statistics, 22

virtual routing interfaces

- allocating memory, 795

virtual switch redundancy protocol (VSRP)

- changing the backup priority, 646
- changing the timer scale, 644
- configuration notes and feature limitations, 634
- configuring authentication, 644
- configuring basic parameters, 642
- configuring fast start, 655
- configuring security features on a VSRP-aware device, 645
- disabling or re-enabling, 643
- displaying information, 651
- displaying the active interfaces for a VRID, 654
- fast start, 654
- interval timers, 639
- MAC address failover, 639
- master election and failover, 635
- overview, 633
- parameters, 640
- priority calculation, 636
- security features, 639
- suppressing RIP advertisement from backups, 650
- track ports, 637
- used to provide redundancy, 635
- VSRP-aware interoperability, 651

VLAN

- 802.1Q tagging, 755
- allocating memory, 795
- and virtual routing interface groups, 760
- appletalk-cable-vlan, 785
- assigning different IDs to reserved VLANs, 767
- broadcast leaks, 763
- configurable parameters, 747
- configuration notes for aggregated VLANs, 799
- configuration rules, 764
- configuring a group, 791
- configuring a multi-range, 746
- configuring an IP, IPX, or AppleTalk protocol VLAN, 784
- configuring an IPX network with dynamic ports, 785
- configuring an IP subnet with dynamic ports, 785
- configuring IP subnet address on multiple port-based

VLAN, 788

- configuring port-based, 740
- configuring uplink ports within a port-based VLAN, 787
- configuring with dynamic ports, 782
- default, 754
- deleting a multi-range, 746
- disable multicast-to-cpu, 1592
- disabling advertising, 889
- disabling learning, 890
- disabling membership aging of dynamic ports, 783
- displaying dual-mode ports, 821
- displaying information, 821
- displaying information for specific ports, 823
- dynamic, 785
- dynamic port membership, 761
- excluded ports, 763
- increasing the number you can configure, 795
- ip access-group, 1735
- IP subnet, IPX network, and AppleTalk cable, 753
- IP subnet, IPX network, and protocol-based
 - configuration example, 770
- IP subnet, IPX network, and protocol-based VLANs

- within port-based VLANs, 772
- ip-proto, 785
- ip-subnet, 785
- IPv6 protocol configuration, 776
- ipv6-proto, 776
- ipx-network ethernet_snap, 787
- ipx-proto, 785
- Layer 2 port-based, 739
- Layer 3 protocol-based, 751
- loop-detection, 66
- mac-vlan-permit, 914
- mld-snooping active | passive, 1619, 1639
- mld-snooping disable-mld-snoop, 1620, 1639
- mld-snooping fast-convergence, 1623, 1642
- mld-snooping fast-leave-v1, 1622, 1641
- mld-snooping port-version 1 | 2, 1640
- mld-snooping port-version 1 | 2 ethernet, 1620
- mld-snooping proxy-off, 1621, 1640
- mld-snooping router-port, 1640
- mld-snooping router-port ethernet, 1621
- mld-snooping static-group, 1621, 1640
- mld-snooping tracking, 1622, 1641
- mld-snooping version 1 | 2, 1620, 1639
- monitor, 937
- multicast active, 1470
- multicast active | passive, 1501
- multicast disable-multicast-snoop, 1471, 1501
- multicast disable-pimsm-snoop, 1477, 1501
- multicast fast-convergence, 1475
- multicast fast-leave-v2, 1475
- multicast passive, 1470
- multicast pimsm-snooping, 1476, 1501
- multicast port-version, 1470
- multicast proxy-off, 1474
- multicast router-port, 1473
- multicast tracking, 1475
- multicast version 2, 1470
- multicast version 3, 1470
- multiple VLAN membership rules, 764
- multi-range, 745
- multi-range show commands, 749
- no webauth, 1907
- no-dynamic-aging, 784
- overview, 738
- protocol-based configuration, 770
- protocol-based for MCT, 850
- pvlan mapping, 814
- pvlan type community, 814, 815
- removing a port, 744
- router-interface ve, 790, 967
- routing using virtual routing interfaces (Layer 3 only),

- 765, 776
- show parameters, 751
- source-guard enable, 1945
- spanning-tree, 744
- spanning-tree 802-1w ethernet, 1140
- spanning-tree ethernet, 1092
- spanning-tree rstp, 1150
- static port membership, 763
- static-mac-address, 1979
- static-mac-address drop, 563
- super aggregated, 763
- super aggregated configuration, 796
- tagged ethernet, 568, 744
- types, 738
- untagged ethernet, 741, 744, 784, 785, 815
- uplink-switch ethernet, 787
- viewing reassigned VLAN IDs, 767
- virtual routing interfaces (Layer 2 only), 765
- vlan by port, 744, 769
- vsrp-aware vrid, 645
- vsrp-aware vrid no-auth, 645
- vsrp-aware vrid tc-vlan-flush, 651
- webauth, 1891
- VLAN group
 - add-vlan, 792
 - configuring, 791
 - displaying, 793, 794
 - group-router-interface, 794
 - remove-vlan, 792
- VLAN port, dual-mode, 818
- VLAN-based mirroring, 937, 939
- VLAN-based static MAC address configuration, 562, 563
- Voice over IP (VoIP) configuration, 60
- voltage selection during bootup (PoE), 664
- voltage selection when a PoE power supply is intalled, 664
- voltage selection when a PoE power supply is removed, 665
- VRID
 - activate, 643
 - advertise backup, 648, 1676
 - backup, 643, 1674, 1677, 1700
 - backup priority, 647
 - backup track-priority, 650
 - backup-hello-interval, 649, 1676
 - changing the backup hello state and interval setting,

648

- changing the dead interval setting, 648
- changing the default track priority setting, 649
- changing the hello interval setting, 647
- changing the hold-down interval setting, 649
- configuring a VRID IP address, 646
- dead-interval, 648
- disabling or re-enabling backup pre-emption setting, 650
- enable | disable, 643
- hello-interval, 648, 1675
- hold-down-interval, 649
- include-port-ethernet, 646
- initial-ttl, 648
- ip vrrp vrid, 1700
- ip-address, 646
- non-preempt-mode, 650, 1678
- owner, 1674, 1677, 1700
- owner priority | track-priority, 1683
- removing a port from the VRID VLAN, 646
- restart-ports, 655
- save-current-values, 647
- specifying a track port setting, 650
- track-port ethernet, 650, 1677
- vsrp vrid, 643

VRRP-E

- authentication types, 1671
- configuration examples, 1700
- configuring IPv4, 1668
- configuring IPv6, 1669
- parameter configuration, 1668
- parameters, 1660
- server virtualization, 1680
- short-path forwarding, 1680
- slow start timer configuration, 1679
- Syslog messages for authentication, 1672

VSRP and MRP signaling, 656

VSRP-aware interoperability, 651

W

web authentication

- automatic authentication, 1900
- changing the login mode (HTTPS or HTTP), 1901
- clearing hosts from the table, 1903
- configuration considerations, 1888
- configuration tasks, 1890
- configuring the re-authentication period, 1903
- customizing pages, 1912
- defining the redirect address, 1906
- defining the web authentication cycle, 1903
- deleting a VLAN, 1906
- displaying information, 1915
- displaying passcodes, 1920
- enabling and disabling, 1891
- enabling RADIUS accounting, 1901
- filtering DNS queries, 1905
- forcing re-authentication when ports are down, 1905
- limiting the number of web authentication attempts, 1903
- mode configuration, 1892
- mode configuration using a RADIUS server, 1894
- mode configuration using local user databases, 1892
- overview, 1887
- passcodes for user authentication, 1896
- specifying hosts that are permanently authenticated, 1902
- specifying trusted ports, 1901

web management

- restricting access, 369

Webauth

- accounting, 1901
- add mac, 1902
- add mac duration | ethernet duration, 1902
- attempt-max-num, 1903
- authenticated-mac-age-time, 1906
- auth-mode none, 1900
- auth-mode passcode, 1897
- auth-mode passcode flush-expired, 1899
- auth-mode passcode generate, 1900
- auth-mode passcode grace-period, 1899
- auth-mode passcode length, 1897
- auth-mode passcode log syslog | snmp-trap, 1899
- auth-mode passcode refresh-type duration, 1898
- auth-mode passcode refresh-type time, 1898
- auth-mode passcode refresh-type time delete-all, 1898
- auth-mode passcode resend-log, 1900
- auth-mode passcode static, 1896
- auth-mode username-password auth-methods, 1895
- auth-mode username-password auth-methods local,

1895
auth-mode username-password auth-methods radius,
1895
auth-mode username-password local-user-database,
1896
block duration, 1904
block mac, 1904
block mac duration, 1904
cycle time, 1903
dns-filter, 1905
enable, 1891
host-max-num, 1905
port-down-auth-mac-cleanup, 1905
reauth-time, 1903
secure-login, 1901
trust-port ethernet, 1902
webauth-redirect-address, 1906
webpage custom-text bottom, 1915
webpage custom-text login-button, 1915
webpage custom-text title, 1913
webpage custom-text top, 1914
webpage logo align center | left | right, 1914
webpage logo copy tftp, 1913
webpage terms copy tftp, 1914

X

XON and XOFF
changing thresholds, 51
thresholds, 50

Z

zero-based IP subnet broadcasts, 988