



Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide—Release 12.1 E

Cisco IOS Release 12.1 E

Corporate Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7814099=
Text Part Number: 78-14099-04



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0807R)

Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide—Release 12.1 E
Copyright © 2003 Cisco Systems, Inc. All rights reserved.

Preface 21

- Audience 21
- Organization 21
- Related Documentation 23
- Conventions 24

CHAPTER 1**Product Overview 1**

- Configuring Embedded CiscoView Support 2
 - Understanding Embedded CiscoView 2
 - Installing and Configuring Embedded CiscoView 2
 - Displaying Embedded CiscoView Information 3

CHAPTER 2**Command-Line Interfaces 1**

- Accessing the CLI 1
 - Accessing the CLI through the EIA/TIA-232 Console Interface 1
 - Accessing the CLI through Telnet 2
- Performing Command Line Processing 3
- Performing History Substitution 3
- Cisco IOS Command Modes 4
- Displaying a List of Cisco IOS Commands and Syntax 5
- ROM-Monitor Command-Line Interface 6

CHAPTER 3**Configuring the Switch for the First Time 1**

- Default Configuration 1
- Configuring the Switch 2
 - Using the Setup Facility or the setup Command 2
 - Using Configuration Mode 10
 - Checking the Running Configuration Before Saving 10
 - Saving the Running Configuration Settings 11
 - Reviewing the Configuration 11
 - Configuring a Default Gateway 12
 - Configuring a Static Route 12
 - Configuring a BOOTP Server 14

- Protecting Access to Privileged EXEC Commands 15
 - Setting or Changing a Static Enable Password 15
 - Using the enable password and enable secret Commands 15
 - Setting or Changing a Line Password 16
 - Setting TACACS+ Password Protection for Privileged EXEC Mode 16
 - Encrypting Passwords 17
 - Configuring Multiple Privilege Levels 17
- Recovering a Lost Enable Password 19
- Modifying the Supervisor Engine Startup Configuration 20
 - Understanding the Supervisor Engine Boot Configuration 20
 - Configuring the Software Configuration Register 21
 - Specifying the Startup System Image 24
 - Understanding Flash Memory 24
 - BOOTLDR Environment Variable 25
 - CONFIG_FILE Environment Variable 26
 - Controlling Environment Variables 26

CHAPTER 4

Configuring EHSA Supervisor Engine Redundancy 1

- Supervisor Engine Redundant Operation 1
 - Supervisor Engine Redundancy Requirements 2
 - Synchronizing the Supervisor Engine Configurations 3
 - Displaying the Supervisor Engine Redundancy 4
 - Copying Files to the Redundant Supervisor Engine 4

CHAPTER 5

Configuring RPR and RPR+ Supervisor Engine Redundancy 1

- Understanding Supervisor Engine Redundancy 1
 - Supervisor Engine Redundancy Overview 1
 - RPR Operation 2
 - RPR+ Operation 2
 - Supervisor Engine Synchronization 3
- Supervisor Engine Redundancy Guidelines and Restrictions 4
 - RPR+ Guidelines and Restrictions 4
 - Hardware Configuration Guidelines and Restrictions 5
 - Restrictions 5
 - Configuration Mode Restrictions 6
- Configuring Supervisor Engine Redundancy 6
 - Configuring RPR and RPR+ 6
 - Synchronizing the Supervisor Engine Configurations 7
 - Displaying the Redundancy States 8

Performing a Fast Software Upgrade	9
Copying Files to an MSFC	10

CHAPTER 6

Configuring Interfaces	1
Understanding Interface Configuration	1
Using the Interface Command	2
Configuring a Range of Interfaces	4
Defining and Using Interface-Range Macros	6
Configuring Optional Interface Features	7
Configuring Ethernet Interface Speed and Duplex Mode	7
Configuring Jumbo Frame Support	10
Configuring IEEE 802.3Z Flow Control	14
Configuring the Port Debounce Timer	15
Adding a Description for an Interface	16
Understanding Online Insertion and Removal	17
Monitoring and Maintaining Interfaces	17
Monitoring Interface Status	17
Clearing Counters on an Interface	18
Resetting an Interface	19
Shutting Down and Restarting an Interface	19

CHAPTER 7

Configuring LAN Ports for Layer 2 Switching	1
Understanding How Layer 2 Switching Works	1
Understanding Layer 2 Ethernet Switching	1
Understanding VLAN Trunks	2
Layer 2 LAN Port Modes	4
Default Layer 2 LAN Interface Configuration	5
Layer 2 LAN Interface Configuration Guidelines and Restrictions	6
Restrictions	6
Guidelines	6
Configuring LAN Interfaces for Layer 2 Switching	7
Configuring a LAN Port for Layer 2 Switching	7
Configuring a Layer 2 Switching Port as a Trunk	8
Configuring a LAN Interface as a Layer 2 Access Port	14
Configuring a Custom IEEE 802.1Q EtherType Field Value	16

CHAPTER 8

Configuring VTP	1
Understanding How VTP Works	1

- Understanding the VTP Domain 2
- Understanding VTP Modes 2
- Understanding VTP Advertisements 2
- Understanding VTP Version 2 3
- Understanding VTP Pruning 3
- VTP Default Configuration 5
- VTP Configuration Guidelines and Restrictions 5
- Configuring VTP 6
 - Configuring VTP Global Parameters 6
 - Configuring the VTP Mode 8
 - Displaying VTP Statistics 10

CHAPTER 9

Configuring VLANs 1

- Understanding How VLANs Work 1
 - VLAN Overview 1
 - VLAN Ranges 2
 - Configurable VLAN Parameters 3
 - Understanding Token Ring VLANs 3
- VLAN Default Configuration 6
- VLAN Configuration Guidelines and Restrictions 8
 - Restrictions 8
 - Guidelines 8
- Configuring VLANs 9
 - VLAN Configuration Options 9
 - Creating or Modifying an Ethernet VLAN 10
 - Assigning a Layer 2 LAN Interface to a VLAN 12
 - Configuring the Internal VLAN Allocation Policy 12
 - Mapping 802.1Q VLANs to ISL VLANs 12

CHAPTER 10

Configuring Private VLANs 1

- Understanding How Private VLANs Work 1
- Private VLAN Configuration Restrictions and Guidelines 2
- Configuring Private VLANs 5
 - Configuring a VLAN as a Private VLAN 5
 - Associating Secondary VLANs with a Primary VLAN 6
 - Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN 7
 - Configuring a Layer 2 Interface as a Private VLAN Host Port 8
 - Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port 9

CHAPTER 11**Configuring Cisco IP Phone Support 1**

- Understanding Cisco IP Phone Support 1
 - Cisco IP Phone Connections 1
 - Cisco IP Phone Voice Traffic 2
 - Cisco IP Phone Data Traffic 3
 - Cisco IP Phone Power Configurations 3
- Default Cisco IP Phone Support Configuration 4
- Cisco IP Phone Support Configuration Guidelines and Restrictions 4
- Configuring Cisco IP Phone Support 5
 - Configuring Voice Traffic Support 5
 - Configuring Data Traffic Support 7
 - Configuring Inline Power Support 8

CHAPTER 12**Configuring Layer 3 Interfaces 1**

- Configuring IP Routing and Addresses 2
- Configuring IPX Routing and Network Numbers 6
- Configuring AppleTalk Routing, Cable Ranges, and Zones 7
- Configuring Other Protocols on Layer 3 Interfaces 8

CHAPTER 13**Configuring EtherChannels 1**

- Understanding How EtherChannels Work 1
 - EtherChannel Feature Overview 2
 - Understanding How EtherChannels Are Configured 2
 - Understanding Port Channel Interfaces 5
 - Understanding Load Balancing 5
- EtherChannel Feature Configuration Guidelines and Restrictions 5
- Configuring EtherChannels 6
 - Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels 7
 - Configuring Channel Groups 8
 - Configuring the LACP System Priority and System ID 10
 - Configuring EtherChannel Load Balancing 11

CHAPTER 14**Configuring IEEE 802.1Q Tunneling and Layer 2 Protocol Tunneling 1**

- Understanding How 802.1Q Tunneling Works 1
- 802.1Q Tunneling Configuration Guidelines and Restrictions 4
 - Restrictions 4
 - Guidelines 4
- Configuring 802.1Q Tunneling 5

Preconfiguration Tasks 6
 Configuring 802.1Q Tunnel Ports 6
 Configuring the Switch to Tag Native VLAN Traffic 7
 Understanding How Layer 2 Protocol Tunneling Works 7
 Configuring Support for Layer 2 Protocol Tunneling 8

CHAPTER 15

Configuring STP and IEEE 802.1s MST 1
 Understanding How STP Works 2
 STP Overview 2
 Understanding the Bridge ID 3
 Understanding Bridge Protocol Data Units 4
 Election of the Root Bridge 4
 STP Protocol Timers 5
 Creating the Spanning Tree Topology 5
 STP Port States 6
 STP and IEEE 802.1Q Trunks 12
 Understanding How IEEE 802.1w RSTP Works 13
 IEEE 802.1w RSTP Overview 13
 RSTP Port Roles 13
 RSTP Port States 14
 Rapid-PVST 14
 Understanding How IEEE 802.1s MST Works 14
 IEEE 802.1s MST Overview 15
 MST-to-PVST Interoperability 16
 Common Spanning Tree 18
 MST Instances 18
 MST Configuration Parameters 18
 MST Regions 19
 Message Age and Hop Count 20
 Default STP Configuration 21
 STP and MST Configuration Guidelines 21
 Configuring STP 22
 Enabling STP 23
 Enabling the Extended System ID 24
 Configuring the Root Bridge 25
 Configuring a Secondary Root Bridge 26
 Configuring STP Port Priority 27
 Configuring STP Port Cost 29
 Configuring the Bridge Priority of a VLAN 30

Configuring the Hello Time	32
Configuring the Forward-Delay Time for a VLAN	32
Configuring the Maximum Aging Time for a VLAN	33
Enabling Rapid-PVST	33
Configuring IEEE 802.1s MST	34
Enabling MST	34
Displaying MST Configurations	36
Configuring MST Instance Parameters	39
Configuring MST Instance Port Parameters	40
Restarting Protocol Migration	40

CHAPTER 16**Configuring Optional STP Features 1**

Understanding How PortFast Works	2
Understanding How BPDU Guard Works	2
Understanding How PortFast BPDU Filtering Works	2
Understanding How UplinkFast Works	3
Understanding How BackboneFast Works	4
Understanding How EtherChannel Guard Works	6
Understanding How Root Guard Works	6
Understanding How Loop Guard Works	6
Enabling PortFast	8
Enabling PortFast BPDU Filtering	10
Enabling BPDU Guard	11
Enabling UplinkFast	12
Enabling BackboneFast	13
Enabling EtherChannel Guard	14
Enabling Root Guard	14
Enabling Loop Guard	15

CHAPTER 17**Configuring IP Unicast Layer 3 Switching on Supervisor Engine 2 1**

Understanding How Layer 3 Switching Works	1
Understanding Hardware Layer 3 Switching on PFC2 and DFCs	2
Understanding Layer 3-Switched Packet Rewrite	2
Default Hardware Layer 3 Switching Configuration	4
Layer 3 Switching Configuration Guidelines and Restrictions	4
Configuring Hardware Layer 3 Switching	5
Displaying Hardware Layer 3 Switching Statistics	6

CHAPTER 18

Configuring IP Multicast Layer 3 Switching 1

- Understanding How IP Multicast Layer 3 Switching Works 1
 - IP Multicast Layer 3 Switching Overview 2
 - Multicast Layer 3 Switching Cache 2
 - IP Multicast Layer 3 Switching Flow Mask 3
 - Layer 3-Switched Multicast Packet Rewrite 3
 - Partially and Completely Switched Flows 4
 - Non-RPF Traffic Processing 5
- Default IP Multicast Layer 3 Switching Configuration 7
- IP Multicast Layer 3 Switching Configuration Guidelines and Restrictions 8
 - PFC2 with MSCF2 8
 - PFC1 with MSFC or MSCF2 8
 - PFC1 and PFC2 General Restrictions 9
 - Unsupported Features 9
- Configuring IP Multicast Layer 3 Switching 9
 - Source Specific Multicast with IGMPv3, IGMP v3lite, and URD 10
 - Enabling IP Multicast Routing Globally 10
 - Enabling IP PIM on Layer 3 Interfaces 10
 - Enabling IP Multicast Layer 3 Switching on Layer 3 Interfaces 11
 - Configuring the Layer 3 Switching Global Threshold 11
 - Enabling Installation of Directly Connected Subnets 12
 - Enabling NetFlow-Based Rate Limiting of RPF Failures 12
 - Enabling CEF-Based Rate Limiting of RPF Failures 13
 - Enabling Shortcut-Consistency Checking 13
 - Configuring ACL-Based Filtering of RPF Failures 14
 - Displaying RPF Failure Rate-Limiting Information 14
 - Displaying IP Multicast Layer 3 Hardware Switching Summary 14
 - Displaying the IP Multicast Routing Table 16
 - Displaying IP Multicast Layer 3 Switching Statistics 17
 - Using Debug Commands 18
 - Clearing IP Multicast Layer 3 Switching Statistics 19

CHAPTER 19

Configuring IP Unicast Layer 3 Switching on Supervisor Engine 1 1

- Understanding How IP MLS Works 2
 - IP MLS Overview 2
 - IP MLS Flows 2
 - Layer 3 MLS Cache 3
 - Flow Masks 3
 - Layer 3-Switched Packet Rewrite 4

IP MLS Operation	5
Default IP MLS Configuration	6
IP MLS Configuration Guidelines and Restrictions	6
Configuring IP MLS	6
Enabling IP MLS Globally	6
Disabling and Enabling IP MLS on a Layer 3 Interface	7
Displaying the Interface IP MLS Configuration	7
Configuring the MLS Aging-Time	8
Setting the Minimum IP MLS Flow Mask	8
Displaying IP MLS Cache Entries	9
Displaying IP MLS Information	9
Displaying IP MLS Cache Entries for a Specific Destination Address	10
Displaying Cache Entries for a Specific Source IP Address	10
Displaying Entries for a Specific IP Flow	11
Clearing IP MLS Cache Entries	11
Displaying IP MLS Contention Table and Statistics	12
Troubleshooting IP MLS	14

CHAPTER 20**Configuring IPX Unicast Layer 3 Switching on Supervisor Engine 1**

Understanding How IPX MLS Works	2
IPX MLS Overview	2
IPX MLS Flows	2
Layer 3 MLS Cache	2
Flow Masks	3
Layer 3-Switched Packet Rewrite	3
IPX MLS Operation	4
Default IPX MLS Configuration	5
Configuration Guidelines and Restrictions	5
Configuring IPX MLS	6
Enabling IPX MLS Globally	6
Enabling IPX MLS on a Layer 3 Interface	6
Configuring the MLS Aging Time	7
Configuring the Minimum IPX MLS Flow Mask	8
Displaying IPX MLS Information	8
Displaying IPX MLS Cache Entries	9
Displaying the IPX MLS Contention Table	11
Displaying IPX MLS VLAN Statistics	12
Clearing IPX MLS Cache Entries	13

Troubleshooting IPX MLS 14

CHAPTER 21

Configuring IGMP Snooping 1

- Understanding How IGMP Snooping Works 1
 - IGMP Snooping Overview 2
 - Joining a Multicast Group 2
 - Leaving a Multicast Group 4
 - Understanding IGMP Snooping Querier 5
 - Understanding IGMP Version 3 Support 6
- Default IGMP Snooping Configuration 6
- IGMP Snooping and IGMP Snooping Querier Configuration Guidelines and Restrictions 6
 - Guidelines 6
 - Restrictions 7
- Enabling the IGMP Snooping Querier 7
- Configuring IGMP Snooping 8
 - Enabling IGMP Snooping 9
 - Configuring IGMP Snooping Learning 10
 - Configuring a Multicast Router Port Statically 10
 - Configuring the IGMP Query Interval 11
 - Enabling IGMP Fast-Leave Processing 11
 - Configuring a Host Statically 12
 - Displaying IGMP Snooping Information 12

CHAPTER 22

Configuring RGMP 1

- Understanding How RGMP Works 1
- Default RGMP Configuration 2
- RGMP Configuration Guidelines and Restrictions 2
- Enabling RGMP on Layer 3 Interfaces 3

CHAPTER 23

Configuring Network Security 1

- ACL Configuration Guidelines 1
- Hardware and Software ACL Support 2
- Guidelines and Restrictions for Using Layer 4 Operators in ACLs 3
 - Determining Layer 4 Operation Usage 3
 - Determining Logical Operation Unit Usage 4
- Configuring the Cisco IOS Firewall Feature Set 5
 - Cisco IOS Firewall Feature Set Support Overview 5
 - Firewall Configuration Guidelines and Restrictions 6

Configuring CBAC on Catalyst 6500 Series Switches	7
Configuring MAC Address-Based Traffic Blocking	8
Configuring VLAN ACLs	8
Understanding VACLs	8
Configuring VACLs	11
Configuring VACL Logging	17
Configuring TCP Intercept	18
Configuring Unicast Reverse Path Forwarding	19
Understanding Unicast RPF Support	19
Configuring Unicast RPF	19
Enabling Self-Pinging	19
Configuring the Unicast RPF Checking Mode	20
Configuring Unicast Flood Protection	21
Configuring MAC Move Notification	22

CHAPTER 24**Configuring Denial of Service Protection 1**

DoS Protection Overview	1
Configuring DoS Protection	2
Supervisor Engine DoS Protection	2
Security ACLs	2
QoS ACLs	4
Forwarding Information Base Rate-Limiting	5
ARP Throttling	5
Monitoring Packet Drop Statistics	6

CHAPTER 25**Configuring IEEE 802.1X Port-Based Authentication 1**

Understanding 802.1X Port-Based Authentication	1
Device Roles	2
Authentication Initiation and Message Exchange	3
Ports in Authorized and Unauthorized States	4
Supported Topologies	4
Default 802.1X Port-Based Authentication Configuration	5
802.1X Port-Based Authentication Guidelines and Restrictions	6
Configuring 802.1X Port-Based Authentication	7
Enabling 802.1X Port-Based Authentication	7
Configuring Switch-to-RADIUS-Server Communication	8
Enabling Periodic Reauthentication	10
Manually Reauthenticating the Client Connected to a Port	11

- Initializing Authentication for the Client Connected to a Port 11
- Changing the Quiet Period 11
- Changing the Switch-to-Client Retransmission Time 12
- Setting the Switch-to-Client Retransmission Time for EAP-Request Frames 13
- Setting the Switch-to-Authentication-Server Retransmission Time for Layer 4 Packets 13
- Setting the Switch-to-Client Frame Retransmission Number 14
- Enabling Multiple Hosts 14
- Resetting the 802.1X Configuration to the Default Values 15
- Displaying 802.1X Status 15

CHAPTER 26

Configuring Port Security 1

- Understanding Port Security 1
- Default Port Security Configuration 2
- Port Security Guidelines and Restrictions 2
- Configuring Port Security 2
 - Configuring Port Security on an Interface 3
 - Configuring Port Security Aging 4
- Displaying Port Security Settings 5

CHAPTER 27

Configuring Layer 3 Protocol Filtering on Supervisor Engine 1 1

- Understanding How Layer 3 Protocol Filtering Works 1
- Configuring Layer 3 Protocol Filtering 2
 - Enabling Layer 3 Protocol Filtering 2
 - Configuring Layer 3 Protocol Filtering on a Layer 2 LAN Interface 3
 - Verifying Layer 3 Protocol Filtering Configuration 3

CHAPTER 28

Configuring Traffic Storm Control 1

- Understanding Traffic Storm Control 1
- Default Traffic Storm Control Configuration 2
- Enabling Traffic Storm Control 2
- Displaying Traffic Storm Control Settings 4

CHAPTER 29

Configuring Broadcast Suppression 1

- Understanding How Broadcast Suppression Works 1
- Broadcast Suppression Configuration Guidelines and Restrictions 2
- Enabling Broadcast Suppression 3

CHAPTER 30**Configuring CDP 1**

- Understanding How CDP Works 1
- Configuring CDP 1
 - Enabling CDP Globally 2
 - Displaying the CDP Global Configuration 2
 - Enabling CDP on a Port 2
 - Displaying the CDP Interface Configuration 3
 - Monitoring and Maintaining CDP 3

CHAPTER 31**Configuring PFC QoS 1**

- Understanding How PFC QoS Works 1
 - Hardware Supported by PFC QoS 2
 - QoS Terminology 3
 - PFC QoS Feature Flowcharts 6
 - PFC QoS Feature Summary 11
 - Ingress LAN Port Features 12
 - PFC Marking and Policing 16
 - LAN Egress Port Features 21
 - PFC QoS Statistics Data Export 24
- PFC QoS Default Configuration 25
- PFC QoS Configuration Guidelines and Restrictions 31
 - Guidelines: 31
 - Restrictions 32
- Configuring PFC QoS 33
 - Enabling PFC QoS Globally 33
 - Enabling Queueing-Only Mode 34
 - Creating Named Aggregate Policers 35
 - Configuring a PFC QoS Policy 37
 - Enabling or Disabling Microflow Policing 50
 - Enabling Microflow Policing of Bridged Traffic 50
 - Enabling or Disabling PFC Features on an Interface 51
 - Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports 52
 - Configuring the Trust State of Ethernet LAN and OSM Ingress Ports 53
 - Configuring the Ingress LAN Port CoS Value 54
 - Configuring Standard-Queue Drop Threshold Percentages 54
 - Mapping CoS Values to Drop Thresholds 59
 - Allocating Bandwidth Between LAN-Port Transmit Queues 64
 - Setting the Receive-Queue Size Ratio on a 1p1q0t or 1p1q8t Ingress LAN Ports 64
 - Setting the LAN-Port Transmit-Queue Size Ratio 65

Configuring DSCP Value Maps 66
 Configuring PFC QoS Statistics Data Export 70

CHAPTER 32

Configuring UDLD 1

Understanding How UDLD Works 1
 UDLD Overview 1
 UDLD Aggressive Mode 2
 Default UDLD Configuration 3
 Configuring UDLD 3
 Enabling UDLD Globally 3
 Enabling UDLD on Individual LAN Interfaces 4
 Disabling UDLD on Fiber-Optic LAN Interfaces 5
 Configuring the UDLD Probe Message Interval 5
 Resetting Disabled LAN Interfaces 6

CHAPTER 33

Configuring NDE 1

Understanding How NDE Works 1
 NDE Overview 2
 NDE from the MSFC 2
 NDE from the PFC 2
 Default NDE Configuration 7
 Configuring NDE 8
 Configuring NDE on the PFC 8
 Configuring NDE on the MSFC 13
 Displaying the NDE Address and Port Configuration 14
 Configuring NDE Flow Filters 15
 Displaying the NDE Configuration 17

CHAPTER 34

Configuring Local SPAN and RSPAN 1

Understanding How Local SPAN and RSPAN Work 1
 Local SPAN and RSPAN Overview 1
 Local SPAN and RSPAN Sessions 3
 Monitored Traffic 4
 SPAN Sources 4
 Destination Ports 5
 Local SPAN and RSPAN Configuration Guidelines and Restrictions 5
 Local SPAN and RSPAN Session Limits 5
 Local SPAN and RSPAN Source and Destination Limits 6

Local SPAN and RSPAN Guidelines and Restrictions	6
VSPAN Guidelines and Restrictions	7
RSPAN Guidelines and Restrictions	7
Configuring Local SPAN and RSPAN	8
Local SPAN and RSPAN Configuration Overview	8
Configuring RSPAN VLANs	9
Configuring Local or RSPAN Sources	9
Monitoring Specific Source VLANs on a Source Trunk Port	10
Configuring Local SPAN and RSPAN Destinations	10
Verifying the Configuration	12
Configuration Examples	13

CHAPTER 35

Configuring Web Cache Services Using WCCP	1
Understanding WCCP	2
WCCP Overview	2
Hardware Acceleration	2
Understanding WCCPv1 Configuration	3
Understanding WCCPv2 Configuration	4
WCCPv2 Features	5
Restrictions for WCCPv2	7
Configuring WCCP	7
Specifying a Version of WCCP	7
Configuring a Service Group Using WCCPv2	8
Excluding Traffic on a Specific Interface from Redirection	9
Registering a Router to a Multicast Address	10
Using Access Lists for a WCCP Service Group	10
Setting a Password for a Router and Cache Engines	11
Verifying and Monitoring WCCP Configuration Settings	12
WCCP Configuration Examples	12
Changing the Version of WCCP on a Router Example	13
Performing a General WCCPv2 Configuration Example	13
Running a Web Cache Service Example	13
Running a Reverse Proxy Service Example	14
Registering a Router to a Multicast Address Example	14
Using Access Lists Example	14
Setting a Password for a Router and Cache Engines Example	15
Verifying WCCP Settings Example	15

CHAPTER 36

Configuring SNMP IfIndex Persistence 1

- Understanding SNMP IfIndex Persistence 1
- Configuring SNMP IfIndex Persistence 1
 - Enabling and Disabling SNMP IfIndex Persistence Globally 2
 - Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces 2
- Configuration Examples 3

CHAPTER 37

Configuring the Switch Fabric Module 1

- Understanding How the Switch Fabric Module Works 1
 - Switch Fabric Module Overview 1
 - Switch Fabric Module Slots 2
 - Switch Fabric Redundancy 2
 - Forwarding Decisions for Layer 3-Switched Traffic 2
 - Switching Modes 2
- Configuring the Switch Fabric Module 3
 - Configuring the Switching Mode 3
 - Configuring Fabric-Required Mode 4
 - Configuring an LCD Message 5
- Monitoring the Switch Fabric Module 5
 - Displaying the Module Information 5
 - Displaying the Switch Fabric Module Redundancy Status 6
 - Displaying Fabric Channel Switching Modes 6
 - Displaying the Fabric Status 7
 - Displaying the Fabric Utilization 7
 - Displaying Fabric Errors 7

CHAPTER 38

Power Management and Environmental Monitoring 1

- Understanding How Power Management Works 1
 - Enabling or Disabling Power Redundancy 2
 - Using the CLI to Power Modules Off and On 3
 - Using the CLI to View System Power Status 3
 - Using the CLI to Power Cycle Modules 4
 - Determining System Power Requirements 4
- Understanding How Environmental Monitoring Works 4
 - Using CLI Commands to Monitor System Environmental Status 4
 - Understanding LED Environmental Indications 4

APPENDIX A **Acronyms** 1

INDEX



Preface

This preface describes who should read the *Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide*, how it is organized, and its document conventions.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining Catalyst 6500 series switches.

Organization

This guide is organized as follows:

Chapter	Title	Description
Chapter 1	Product Overview	Presents an overview of the Catalyst 6500 series switches.
Chapter 2	Command-Line Interfaces	Describes how to use the command-line interface (CLI).
Chapter 3	Configuring the Switch for the First Time	Describes how to perform a baseline configuration.
Chapter 4	Configuring EHSA Supervisor Engine Redundancy	Describes how to configure EHSA supervisor engine redundancy.
Chapter 5	Configuring RPR and RPR+ Supervisor Engine Redundancy	Describes how to configure RPR and RPR+ supervisor engine redundancy.
Chapter 6	Configuring Interfaces	Describes how to configure non-layer-specific features on LAN interfaces.
Chapter 7	Configuring LAN Ports for Layer 2 Switching	Describes how to configure LAN interfaces to support Layer 2 features, including VLAN trunks.
Chapter 8	Configuring VTP	Describes how to configure the VLAN Trunking Protocol (VTP).
Chapter 9	Configuring VLANs	Describes how to configure VLANs.
Chapter 10	Configuring Private VLANs	Describes how to configure private VLANs.

Chapter	Title	Description
Chapter 11	Configuring Cisco IP Phone Support	Describes how to configure Cisco IP Phone support.
Chapter 12	Configuring Layer 3 Interfaces	Describes how to configure LAN interfaces to support Layer 3 features.
Chapter 13	Configuring EtherChannels	Describes how to configure Layer 2 and Layer 3 EtherChannel port bundles.
Chapter 14	Configuring IEEE 802.1Q Tunneling and Layer 2 Protocol Tunneling	Describes how to configure IEEE 802.1Q tunneling and Layer 2 protocol tunneling.
Chapter 15	Configuring STP and IEEE 802.1s MST	Describes how to configure the Spanning Tree Protocol (STP) and explains how STP works.
Chapter 16	Configuring Optional STP Features	Describes how to configure the STP PortFast, UplinkFast, and BackboneFast features.
Chapter 17	Configuring IP Unicast Layer 3 Switching on Supervisor Engine 2	Describes how to configure IP unicast Layer 3 switching for Supervisor Engine 2.
Chapter 18	Configuring IP Multicast Layer 3 Switching	Describes how to configure IP Multicast Multilayer Switching (MMLS).
Chapter 19	Configuring IP Unicast Layer 3 Switching on Supervisor Engine 1	Describes how to configure IP unicast Layer 3 switching for Supervisor Engine 1.
Chapter 20	Configuring IPX Unicast Layer 3 Switching on Supervisor Engine 1	Describes how to configure IPX unicast Layer 3 switching for Supervisor Engine 1.
Chapter 21	Configuring IGMP Snooping	Describes how to configure Internet Group Management Protocol (IGMP) snooping.
Chapter 22	Configuring RGMP	Describes how to configure Router-Port Group Management Protocol (RGMP).
Chapter 23	Configuring Network Security	Describes how to configure network security features that are unique to the Catalyst 6500 series switches.
Chapter 24	Configuring Denial of Service Protection	Describes how to configure denial of service protection.
Chapter 25	Configuring IEEE 802.1X Port-Based Authentication	Describes how to configure IEEE 802.1X port-based authentication.
Chapter 26	Configuring Port Security	Describes how to configure port security.
Chapter 27	Configuring Layer 3 Protocol Filtering on Supervisor Engine 1	Describes how to configure Layer 3 protocol filtering on Supervisor Engine 1.
Chapter 28	Configuring Traffic Storm Control	Describes how to configure traffic storm control.
Chapter 29	Configuring Broadcast Suppression	Describes how to configure broadcast suppression.
Chapter 30	Configuring CDP	Describes how to configure Cisco Discovery Protocol (CDP).
Chapter 32	Configuring UDLD	Describes how to configure the UniDirectional Link Detection (UDLD) protocol.
Chapter 31	Configuring PFC QoS	Describes how to configure quality of service (QoS).

Chapter	Title	Description
Chapter 33	Configuring NDE	Describes how to configure Neflow Data Export (NDE).
Chapter 34	Configuring Local SPAN and RSPAN	Describes how to configure the Switch Port Analyzer (SPAN).
Chapter 35	Configuring Web Cache Services Using WCCP	Describes how to configure web cache services using WCCP.
Chapter 36	Configuring SNMP IfIndex Persistence	Describes how to configure SNMP ifIndex persistence.
Chapter 37	Configuring the Switch Fabric Module	Describes how to configure the Switch Fabric Module.
Chapter 38	Power Management and Environmental Monitoring	Describes how to configure power management and environmental monitoring features.

Related Documentation

The following publications are available for the Catalyst 6500 series switches:

- *Catalyst 6500 Series Switch Installation Guide*
- *Catalyst 6500 Series Switch Module Installation Guide*
- *Catalyst 6500 Series Switch Cisco IOS Command Reference*
- *Catalyst 6500 Series Switch Cisco IOS System Message Guide*
- *Release Notes for Cisco IOS Release 12.1 E on the Catalyst 6500 and Cisco 7600 Supervisor Engine and MSFC*
- *Cisco IOS Configuration Guides and Command References*—Use these publications to help you configure Cisco IOS software features not described in the Catalyst 6500 series switch publications:
 - *Configuration Fundamentals Configuration Guide*
 - *Configuration Fundamentals Command Reference*
 - *Bridging and IBM Networking Configuration Guide*
 - *Bridging and IBM Networking Command Reference*
 - *Interface Configuration Guide*
 - *Interface Command Reference*
 - *Network Protocols Configuration Guide, Part 1, 2, and 3*
 - *Network Protocols Command Reference, Part 1, 2, and 3*
 - *Security Configuration Guide*
 - *Security Command Reference*
 - *Switching Services Configuration Guide*
 - *Switching Services Command Reference*
 - *Voice, Video, and Home Applications Configuration Guide*
 - *Voice, Video, and Home Applications Command Reference*
 - *Software Command Summary*

- *Software System Error Messages*
- *Debug Command Reference*
- *Internetwork Design Guide*
- *Internetwork Troubleshooting Guide*
- *Configuration Builder Getting Started Guide*

The Cisco IOS Configuration Guides and Command References are located at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

- For information about MIBs, go to this URL:
<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

Conventions

This document uses the following conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in boldface .
<i>italic font</i>	Arguments for which you supply values are in <i>italics</i> .
[]	Elements in square brackets are optional.
{ x y z }	Alternative keywords are grouped in braces and separated by vertical bars.
[x y z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
<i>screen font</i>	Terminal sessions and information the system displays are in <i>screen font</i> .
boldface screen font	Information you must enter is in boldface screen font .
<i>italic screen font</i>	Arguments for which you supply values are in <i>italic screen font</i> .
→	This pointer highlights an important line of text in an example.
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.

Notes use the following conventions:



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Cautions use the following conventions:

**Caution**

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.



Product Overview

The Cisco IOS on the Catalyst 6500 Series Switches product supports the following hardware and software:

- Supervisor Engine 2, Policy Feature Card 2 (PFC2), and Multilayer Switch Feature Card 2 (MSFC2); and in Catalyst 6500 series switches:
 - Switch Fabric Module
 - Fabric-enabled switching modules
 - Fabric-enabled switching modules with a distributed forwarding card (DFC)
- Supervisor Engine 1, PFC, and MSFC or MSFC2
- All Layer 2 and Layer 3 configuration from the same user interface
- Except for VLANs, Layer 2 and Layer 3 configuration is stored in a standard IOS configuration file

Refer to the *Release Notes for Cisco IOS Release 12.1 E on the Catalyst 6500 and Cisco 7600 Supervisor Engine and MSFC* publication for complete information about the chassis, modules, and software features supported by the Catalyst 6500 series switches:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/12_1e/ol_2310.htm

The Cisco IOS on the Catalyst 6500 Series Switches product supports configuration using:

- CLI—See [Chapter 2, “Command-Line Interfaces”](#)
- SNMP—Refer to the *IOS Configuration Fundamentals Configuration Guide* and *Command Reference* at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>
- IOS web browser interface—Refer to “Using the Cisco Web Browser” in the *IOS Configuration Fundamentals Configuration Guide* and *Command Reference* at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>
- Embedded CiscoView—See the “[Configuring Embedded CiscoView Support](#)” section on page 1-2.

Configuring Embedded CiscoView Support

These sections describe the Embedded CiscoView support available with Release 12.1(20)E and later releases:

- [Understanding Embedded CiscoView, page 1-2](#)
- [Installing and Configuring Embedded CiscoView, page 1-2](#)
- [Displaying Embedded CiscoView Information, page 1-3](#)

Understanding Embedded CiscoView

The Embedded CiscoView network management system is a web-based interface that uses HTTP and SNMP to provide a graphical representation of the switch and to provide a GUI-based management and configuration interface. You can download the Java Archive (JAR) files for Embedded CiscoView at this URL:

<http://www.cisco.com/kobayashi/sw-center/netmgmt/ciscoview/embed-cview-planner.shtml>

Installing and Configuring Embedded CiscoView

To install and configure Embedded CiscoView, perform the following steps:

	Command	Purpose
Step 1	Router# <code>dir device_name</code>	Displays the contents of the device. If you are installing Embedded CiscoView for the first time, or if the CiscoView directory is empty, skip to Step 4 .
Step 2	Router# <code>delete device_name:cv/*</code>	Removes existing files from the CiscoView directory.
Step 3	Router# <code>squeeze device_name:</code>	Recovers the space in the file system.
Step 4	Router# <code>archive tar /xtract tftp:// ip address of tftp server/ciscoview.tar device_name:cv</code>	Extracts the CiscoView files from the tar file on the TFTP server to the CiscoView directory.
Step 5	Router# <code>dir device_name:</code>	Displays the contents of the device. In a redundant configuration, repeat Step 1 through Step 5 for the file system on the redundant supervisor engine.
Step 6	Router# <code>configure terminal</code>	Enters global configuration mode.
Step 7	Router(config)# <code>ip http server</code>	Enables the HTTP web server.
Step 8	Router(config)# <code>snmp-server community string ro</code>	Configures the SNMP password for read-only operation.
Step 9	Router(config)# <code>snmp-server community string rw</code>	Configures the SNMP password for read/write operation.



Note

The default password for accessing the switch web page is the enable-level password of the switch.

For more information about web access to the switch, refer to “Using the Cisco Web Browser” in the *IOS Configuration Fundamentals Configuration Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/fcprt1/fcd105.htm

Displaying Embedded CiscoView Information

To display the Embedded CiscoView information, enter the following EXEC commands:

Command	Purpose
Router# <code>show ciscoview package</code>	Displays information about the Embedded CiscoView files.
Router# <code>show ciscoview version</code>	Displays the Embedded CiscoView version.



Command-Line Interfaces

This chapter describes the command-line interfaces (CLIs) you use to configure the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication and the Release 12.1 publications at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

This chapter consists of these sections:

- [Accessing the CLI, page 2-1](#)
- [Performing Command Line Processing, page 2-3](#)
- [Performing History Substitution, page 2-3](#)
- [Cisco IOS Command Modes, page 2-4](#)
- [Displaying a List of Cisco IOS Commands and Syntax, page 2-5](#)
- [ROM-Monitor Command-Line Interface, page 2-6](#)

Accessing the CLI

These sections describe accessing the CLI:

- [Accessing the CLI through the EIA/TIA-232 Console Interface, page 2-1](#)
- [Accessing the CLI through Telnet, page 2-2](#)

Accessing the CLI through the EIA/TIA-232 Console Interface



Note

EIA/TIA-232 was known as recommended standard 232 (RS-232) before its acceptance as a standard by the Electronic Industries Alliance (EIA) and Telecommunications Industry Association (TIA).

Perform initial configuration over a connection to the EIA/TIA-232 console interface. Refer to the *Catalyst 6500 Series Switch Module Installation Guide* for console interface cable connection procedures.

To make a console connection, perform this task:

	Command	Purpose
Step 1	Press Return .	Brings up the prompt.
Step 2	Router> enable	Initiates enable mode enable.
Step 3	Password: <i>password</i> Router#	Completes enable mode enable.
Step 4	Router# quit	Exits the session when finished.

After making a console connection, you see this display:

Press Return for Console prompt

```
Router> enable
Password:
Router#
```

Accessing the CLI through Telnet



Note

Before you can make a Telnet connection to the switch, you must configure an IP address (see the [“Configuring IP Routing and Addresses”](#) section on page 12-2).

The switch supports up to eight simultaneous Telnet sessions. Telnet sessions disconnect automatically after remaining idle for the period specified with the **exec-timeout** command.

To make a Telnet connection to the switch, perform this task:

	Command	Purpose
Step 1	telnet { <i>hostname</i> <i>ip_addr</i> }	Makes a Telnet connection from the remote host, to the switch you want to access.
Step 2	Password: <i>password</i> Router#	Initiates authentication. Note If no password has been configured, press Return .
Step 3	Router> enable	Initiates enable mode enable.
Step 4	Password: <i>password</i> Router#	Completes enable mode enable.
Step 5	Router# quit	Exits the session when finished.

This example shows how to open a Telnet session to the switch:

```
unix_host% telnet Router_1
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.

User Access Verification

Password:
Router_1> enable
Password:
Router_1#
```

Performing Command Line Processing

Commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters. You can scroll through the last 20 commands stored in the history buffer, and enter or edit the command at the prompt. [Table 2-1](#) lists the keyboard shortcuts for entering and editing commands.

Table 2-1 Keyboard Shortcuts

Keystrokes	Purpose
Press Ctrl-B or press the left arrow key ¹	Moves the cursor back one character
Press Ctrl-F or press the right arrow key ¹	Moves the cursor forward one character
Press Ctrl-A	Moves the cursor to the beginning of the command line
Press Ctrl-E	Moves the cursor to the end of the command line
Press Esc B	Moves the cursor back one word
Press Esc F	Moves the cursor forward one word

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

Performing History Substitution

The history buffer stores the last 20 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands. [Table 2-2](#) lists the history substitution commands.

Table 2-2 History Substitution Commands

Command	Purpose
Ctrl-P or the up arrow key. ¹	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the down arrow key. ¹	Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.
Router# show history	While in EXEC mode, lists the last several commands you have just entered.

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

Cisco IOS Command Modes



Note

For complete information about Cisco IOS command modes, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and the *Cisco IOS Configuration Fundamentals Command Reference* publication at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

The Cisco IOS user interface is divided into many different modes. The commands available to you depend on which mode you are currently in. To get a list of the commands in a given mode, type a question mark (?) at the system prompt. See the “[Displaying a List of Cisco IOS Commands and Syntax](#)” section on page 2-5.

When you start a session on the switch, you begin in user mode, often called user EXEC mode. Only a limited subset of the commands are available in EXEC mode. To have access to all commands, you must enter privileged EXEC mode. Normally, you must type in a password to access privileged EXEC mode. From privileged EXEC mode, you can type in any EXEC command or access global configuration mode.

The configuration modes allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across reboots. You must start at global configuration mode. From global configuration mode, you can enter interface configuration mode, subinterface configuration mode, and a variety of protocol-specific modes.



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

ROM-monitor mode is a separate mode used when the switch cannot boot properly. For example, the switch might enter ROM-monitor mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted at startup. See the “[ROM-Monitor Command-Line Interface](#)” section on page 2-6.

Table 2-3 lists and describes frequently used Cisco IOS modes.

Table 2-3 Frequently Used Cisco IOS Command Modes

Mode	Description of Use	How to Access	Prompt
User EXEC	Connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and display system information.	Log in.	Router>
Privileged EXEC (enable)	Set operating parameters. The privileged command set includes the commands in user EXEC mode, as well as the configure command. Use this command to access the other command modes.	From the user EXEC mode, enter the enable command and the enable password.	Router#
Global configuration	Configure features that affect the system as a whole.	From the privileged EXEC mode, enter the configure terminal command.	Router(config)#
Interface configuration	Many features are enabled for a particular interface. Interface commands enable or modify the operation of an interface.	From global configuration mode, enter the interface <i>type slot/port</i> command.	Router(config-if)#
Console configuration	From the directly connected console or the virtual terminal used with Telnet, use this configuration mode to configure the console interface.	From global configuration mode, enter the line console 0 command.	Router(config-line)#

The Cisco IOS command interpreter, called the EXEC, interprets and executes the commands you enter. You can abbreviate commands and keywords by entering just enough characters to make the command unique from other commands. For example, you can abbreviate the **show** command to **sh** and the **configure terminal** command to **conf t**.

When you type **exit**, the switch backs out one level. To exit configuration mode completely and return to privileged EXEC mode, press **Ctrl-Z**.

Displaying a List of Cisco IOS Commands and Syntax

In any command mode, you can display a list of available commands by entering a question mark (?).

```
Router> ?
```

To display a list of commands that begin with a particular character sequence, type in those characters followed by the question mark (?). Do not include a space. This form of help is called word help because it completes a word for you.

```
Router# co?
configure
```

To display keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help because it reminds you which keywords or arguments are applicable based on the command, keywords, and arguments you have already entered.

For example:

```
Router# configure ?
memory          Configure from NV memory
network         Configure from a TFTP network host
overwrite-network Overwrite NV memory from TFTP network host
terminal        Configure from the terminal
<cr>
```

To redisplay a command you previously entered, press the up arrow key or **Ctrl-P**. You can continue to press the up arrow key to see the last 20 commands you entered.



Tip

If you are having trouble entering a command, check the system prompt, and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

Enter **exit** to return to the previous mode. Press **Ctrl-Z** or enter the **end** command in any mode to immediately return to privileged EXEC mode.

ROM-Monitor Command-Line Interface

The ROM-monitor is a ROM-based program that executes upon platform power-up, reset, or when a fatal exception occurs. The switch enters ROM-monitor mode if it does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROM-monitor mode. From the ROM-monitor mode, you can load a software image manually from Flash memory, from a network server file, or from bootflash.

You can also enter ROM-monitor mode by restarting and pressing the **Break** key during the first 60 seconds of startup.



Note

The **Break** key is always enabled for 60 seconds after rebooting, regardless of whether the **Break** key is configured to be off by configuration register settings.

To access the ROM-monitor mode through a terminal server, you can escape to the Telnet prompt and enter the **send break** command for your terminal emulation program to break into ROM-monitor mode.

Once you are in ROM-monitor mode, the prompt changes to rommon 1>. Enter a question mark (?) to see the available ROM-monitor commands.

For more information about the ROM-monitor commands, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.



Configuring the Switch for the First Time

This chapter contains information about how to initially configure the Catalyst 6500 series switch, which supplements the administration information and procedures in these publications:

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/index.htm
- *Cisco IOS Configuration Fundamentals Configuration Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/index.htm



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication and the Release 12.1 publications at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

This chapter consists of these sections:

- [Default Configuration](#), page 3-1
- [Configuring the Switch](#), page 3-2
- [Protecting Access to Privileged EXEC Commands](#), page 3-15
- [Recovering a Lost Enable Password](#), page 3-19
- [Modifying the Supervisor Engine Startup Configuration](#), page 3-20

Default Configuration

[Table 3-1](#) shows the default configuration.

Table 3-1 Default Configuration

Feature	Default Value
Administrative connection	Normal mode
Global information	No value for the following: <ul style="list-style-type: none"> • System name • System contact • Location
System clock	No value for system clock time
Passwords	No passwords configured for normal mode or enable mode (press the Return key)
Prompt	Router>

Configuring the Switch

These sections describe how to configure the switch:

- [Using the Setup Facility or the setup Command, page 3-2](#)
- [Using Configuration Mode, page 3-10](#)
- [Checking the Running Configuration Before Saving, page 3-10](#)
- [Saving the Running Configuration Settings, page 3-11](#)
- [Reviewing the Configuration, page 3-11](#)
- [Configuring a Default Gateway, page 3-12](#)
- [Configuring a Static Route, page 3-12](#)
- [Configuring a BOOTP Server, page 3-14](#)



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Using the Setup Facility or the setup Command

These sections describe the setup facility and the **setup** command:

- [Setup Overview, page 3-2](#)
- [Configuring the Global Parameters, page 3-3](#)
- [Configuring Interfaces, page 3-8](#)

Setup Overview

At initial startup, the switch automatically defaults to the setup facility. (The **setup** command facility functions exactly the same as a completely unconfigured system functions when you first boot it up.) You can run the setup facility by entering the **setup** command at the enable prompt (#).

When you enter the **setup** command, current system configuration defaults are displayed in square brackets [] as you move through the **setup** command process and are queried by the system to make changes.

For example, you will see this display when you use the setup facility:

```
Configuring interface FastEthernet3/1:
  Is this interface in use?: yes
  Configure IP on this interface?: yes
```

When you use the **setup** command, you see this display:

```
Configuring interface FastEthernet4/1:
  Is this interface in use?[yes]: yes
  Configure IP on this interface?[yes]: yes
```

Configuring the Global Parameters

When you first start the setup facility or enter the **setup** command, you are queried by the system to configure the global parameters, which are used for controlling system-wide settings.

To boot the switch and enter the global parameters, follow these steps:

- Step 1** Connect a console terminal to the console interface on the supervisor engine, and then boot the system to the user EXEC prompt (Router>).

The following display appears after you boot the Catalyst 6500 series switch (depending on your configuration, your display might not exactly match the example):

```
System Bootstrap, Version 6.1(2)
Copyright (c) 1994-2000 by cisco Systems, Inc.
c6k_sup2 processor with 131072 Kbytes of main memory

rommon 1 > boot slot0:c6sup22-jsv-mz.121-5c.EX.bin

Self decompressing the image : #####
#####
#####
#####
#####
[OK]

Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco Internetwork Operating System Software
IOS (tm) c6sup2_sp Software (c6sup2_sp-SPV-M), Version 12.1(5c)EX, EARLY DEPLOYM
ENT RELEASE SOFTWARE (fc1)
Synced to mainline version: 12.1(5c)
TAC:Home:Software:Ios General:CiscoIOSRoadmap:12.1
```

```

Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Wed 28-Mar-01 18:36 by hqluong
Image text-base: 0x30020980, data-base: 0x306B8000

Start as Primary processor

00:00:05: %SYS-3-LOGGER_FLUSHING: System pausing to ensure console debugging out
put.

00:00:03: Currently running ROMMON from S (Gold) region
00:00:05: %OIR-6-CONSOLE: Changing console ownership to route processor

System Bootstrap, Version 12.1(3r)E2, RELEASE SOFTWARE (fcl)
Copyright (c) 2000 by cisco Systems, Inc.
Cat6k-MSFC2 platform with 131072 Kbytes of main memory

rommon 1 > boot

Self decompressing the image : #####
#####
## [OK]

                Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

                cisco Systems, Inc.
                170 West Tasman Drive
                San Jose, California 95134-1706

Cisco Internetwork Operating System Software
IOS (tm) MSFC2 Software (C6MSFC2-BOOT-M), Version 12.1(3a)E4, EARLY DEPLOYMENT R
ELEASE SOFTWARE (fcl)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Sat 14-Oct-00 05:33 by eaarmas
Image text-base: 0x30008980, data-base: 0x303B6000

cisco Cat6k-MSFC2 (R7000) processor with 114688K/16384K bytes of memory.
Processor board ID SAD04430J9K
R7000 CPU at 300Mhz, Implementation 39, Rev 2.1, 256KB L2, 1024KB L3 Cache
Last reset from power-on
X.25 software, Version 3.0.0.
509K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).

Press RETURN to get started!

```



Note The first two sections of the configuration script (the banner and the installed hardware) appear only at initial system startup. On subsequent uses of the **setup** command facility, the setup script begins with the following System Configuration Dialog.


```
--- System Configuration Dialog ---
```

```
Continue with configuration dialog? [yes/no]: y
```

```
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '['].
```

```
Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system
```



Note The examples in this section are intended as examples only. Your configuration might look differently depending on your system configuration.

Step 2 Enter **yes** or press **Return** when asked if you want to enter the configuration dialog and if you want to see the current interface summary. Press **Return** to accept the default (yes):

```
Would you like to enter the initial configuration dialog? [yes]:
```

```
First, would you like to see the current interface summary? [yes]:
```

This example of a **yes** response (displayed during the setup facility) shows a switch at first-time startup; that is, nothing has been configured:

```
Current interface summary
```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	TFTP	administratively down	down
GigabitEthernet1/1	unassigned	YES	TFTP	administratively down	down
GigabitEthernet1/2	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/1	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/2	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/3	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/4	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/5	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/6	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/7	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/8	unassigned	YES	TFTP	administratively down	down

```
(Additional displayed text omitted from this example.)
```

This example of a **yes** response (displayed during the setup command facility) shows a switch with some interfaces already configured:

```
Current interface summary
```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	TFTP	administratively down	down

```

GigabitEthernet1/1      172.20.52.34    YES NVRAM  up          up
GigabitEthernet1/2      unassigned      YES TFTP   administratively down down
GigabitEthernet3/1      unassigned      YES TFTP   administratively down down
GigabitEthernet3/2      unassigned      YES TFTP   administratively down down
GigabitEthernet3/3      unassigned      YES TFTP   administratively down down
GigabitEthernet3/4      unassigned      YES TFTP   administratively down down
GigabitEthernet3/5      unassigned      YES TFTP   administratively down down
GigabitEthernet3/6      unassigned      YES TFTP   administratively down down
GigabitEthernet3/7      unassigned      YES TFTP   administratively down down
GigabitEthernet3/8      unassigned      YES TFTP   administratively down down
<...output truncated...>

```

Step 3 Choose which protocols to support on your interfaces. On IP installations only, you can accept the default values for most of the questions.

A typical minimal configuration using IP follows and continues through [Step 8](#):

Configuring global parameters:

```
Enter host name [Router]: Router
```

Step 4 Enter the enable secret password when the following is displayed (remember this password for future reference):

```

The enable secret is a password used to protect access to
privileged EXEC and configuration modes. This password, after
entered, becomes encrypted in the configuration.
Enter enable secret: barney

```

Step 5 Enter the enable password when the following is displayed (remember this password for future reference):

```

The enable password is used when you do not specify an
enable secret password, with some older software versions, and
some boot images.
Enter enable password: wilma

```

The commands available at the user EXEC level are a subset of those available at the privileged EXEC level. Because many privileged EXEC commands are used to set operating parameters, you should protect these commands with passwords to prevent unauthorized use.

You must enter the correct password to gain access to privileged EXEC commands. When you are running from the boot ROM monitor, the enable password might be the correct one to use, depending on your boot ROM level.

The enable and enable secret passwords need to be different for effective security. You can enter the same password for both enable and enable secret during the setup script, but you receive a warning message indicating that you should enter a different password.



Note An enable secret password can contain from 1 to 25 uppercase and lowercase alphanumeric characters; an enable password can contain any number of uppercase and lowercase alphanumeric characters. In both cases, a number cannot be the first character. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored; trailing spaces are recognized.

Step 6 Enter the virtual terminal password when the following is displayed (remember this password for future reference):

```
The virtual terminal password is used to protect
access to the router over a network interface.
Enter virtual terminal password: bambam
```

Step 7 In most cases you will use IP routing. If so, you must also select an interior routing protocol, for example, the Enhanced Interior Gateway Routing Protocol (EIGRP).

Enter **yes** (the default) or press **Return** to configure IP, and then select EIGRP:

```
Configure IP? [yes]:
Configure EIGRP routing? [yes]:
Your IGRP autonomous system number [1]: 301
```

Step 8 Enter **yes** or **no** to accept or refuse SNMP management:

```
Configure SNMP Network Management? [yes]:
Community string [public]:
```

For complete SNMP information and procedures, refer to these publications:

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.1, “Cisco IOS System Management,” “Configuring SNMP Support,” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/fcprt3/fcd301.htm
- *Cisco IOS Configuration Fundamentals Configuration Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/index.htm

To provide a review of what you have done, a display similar to the following appears and lists all of the configuration parameters you selected in Steps 3 through 8. These parameters and their defaults are shown in the order in which they appeared on your console terminal:

The following configuration command script was created:

```
hostname router
enable secret 5 $1$S3Lx$SuiTYg2UrFK1U0dgWdjvwxw.
enable password lab
line vty 0 4
password lab
no snmp-server
!
ip routing eigrp 301

!
interface Vlan1
shutdown
no ip address
!
interface GigabitEthernet1/1
```

```

shutdown
no ip address
!
interface GigabitEthernet1/2
shutdown
no ip address
!
.
<...output truncated...>
.!
end

```

```

[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

```

```

Enter your selection [2]: 2
% You can enter the setup, by typing setup at IOS command prompt
Router#

```

This completes the procedure on how to configure global parameters. The setup facility continues with the process to configure interfaces in the next section “[Configuring Interfaces](#).”

Configuring Interfaces

This section provides steps for configuring installed interfaces (using the setup facility or **setup** command facility) to allow communication over your external networks. To configure the interface parameters, you need your interface network addresses, subnet mask information, and which protocols you want to configure. (For additional interface configuration information on each of the modules available, refer to the individual configuration notes that shipped with your modules.)



Note

The examples in this section are intended as examples only. Your configuration might look differently depending on your system configuration.

To configure interfaces, follow these steps:

Step 1 At the prompt for the Gigabit Ethernet interface configuration, enter the appropriate responses for your requirements, using your own address and subnet mask:

```

Do you want to configure GigabitEthernet1/1 interface? [no]: yes
Configure IP on this interface? [no]: yes
IP address for this interface: 172.20.52.34
Subnet mask for this interface [255.255.0.0] : 255.255.255.224
Class B network is 172.20.0.0, 27 subnet bits; mask is /27

```

Step 2 At the prompt for all other interface types, enter the appropriate responses for your requirements:

```

Do you want to configure FastEthernet5/1 interface? [no]: y
Configure IP on this interface? [no]: y
IP address for this interface: 172.20.52.98
Subnet mask for this interface [255.255.0.0] : 255.255.255.248
Class B network is 172.20.0.0, 29 subnet bits; mask is /29

```

Repeat this step for each interface you need to configure. Proceed to Step 3 to check and verify your configuration parameters.

When you reach and respond to the configuration dialog for the last installed interface, your interface configuration is complete.

- Step 3** Check and verify the entire list of configuration parameters, which should display on your console terminal and end with the following query:

```
Use this configuration? [yes/no]:
```

A **no** response places you back at the enable prompt (#). You will need to reenter the **setup** command to reenter your configuration. A **yes** response saves the running configuration to NVRAM as follows:

```
Use this configuration? [yes/no]: yes
[OK]
Use the enabled mode 'configure' command to modify this configuration.
Press RETURN to get started!
```

After you press the **Return** key, this prompt appears:

```
Router>
```

This completes the procedures for configuring global parameters and interface parameters in your system. Your interfaces are now available for limited use.

If you want to modify the currently saved configuration parameters after the initial configuration, enter the **setup** command. To perform more complex configurations, enter configuration mode and use the **configure** command. Check the current state of the switch using the **show version** command, which displays the software version and the interfaces, as follows:

```
Router# show version
Cisco Internetwork Operating System Software
IOS (tm) c6sup2_rp Software (c6sup2_rp-JS-M), Version 12.1(13)E1, EARLY DEPLOYM
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Wed 06-Nov-02 13:57 by eaarmas
Image text-base: 0x40008C00, data-base: 0x41A72000

ROM: System Bootstrap, Version 12.1(11r)E1, RELEASE SOFTWARE (fc1)
BOOTLDR: c6sup2_rp Software (c6sup2_rp-JS-M), Version 12.1(13)E1, EARLY DEPLOYM

Router uptime is 4 hours, 22 minutes
Time since Router switched to active is 4 hours, 22 minutes
System returned to ROM by power-on (SP by power-on)
System image file is "sup-bootflash:c6sup22-js-mz.121-13.E1"

cisco Catalyst 6000 (R7000) processor with 112640K/18432K bytes of memory.
Processor board ID SAD06210067
R7000 CPU at 300Mhz, Implementation 39, Rev 3.3, 256KB L2, 1024KB L3 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
4 Virtual Ethernet/IEEE 802.3 interface(s)
48 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)
381K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).
Configuration register is 0x2102
Router#
```

For detailed interface configuration information, refer to the *Cisco IOS Interface Configuration Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_c/index.htm

Using Configuration Mode

If you prefer not to use the setup facility, you can configure the switch from configuration mode as follows:

-
- Step 1** Connect a console terminal to the console interface of your supervisor engine.
- Step 2** When you are asked if you want to enter the initial dialog, answer **no** to enter the normal operating mode as follows:
- ```
Would you like to enter the initial dialog? [yes]: no
```
- Step 3** After a few seconds you will see the user EXEC prompt (Router>). Type **enable** to enter enable mode:
- ```
Router> enable
```



Note Configuration changes can only be made in enable mode.

The prompt will change to the privileged EXEC prompt (#) as follows:

```
Router#
```

- Step 4** At the prompt (#), enter the **configure terminal** command to enter configuration mode as follows:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

At the prompt, enter the **interface** *type slot/interface* command to enter interface configuration mode as follows:

```
Router(config)# interface fastethernet 5/1
Router(config-if)#
```

In either of these configuration modes, you can enter any changes to the configuration. Enter the **end** command to exit configuration mode.

- Step 5** Save your settings. (See the “Saving the Running Configuration Settings” section on page 3-11.)
-

Your switch is now minimally configured and can boot with the configuration you entered. To see a list of the configuration commands, enter **?** at the prompt or press the **help** key in configuration mode.

Checking the Running Configuration Before Saving

You can check the configuration settings you entered or changes you made by entering the **show running-config** command at the privileged EXEC prompt (#) as follows:

```
Router# show running-config
Building configuration...
```

```
Current configuration:
Current configuration : 3441 bytes
!
version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
!
hostname Router
!
boot buffersize 522200
boot system flash slot0:c6sup22-jsv-mz.121-5c.EX.bin
boot bootldr bootflash:c6msfc2-boot-mz.121-3a.E4
enable password lab
!
redundancy
  main-cpu
  auto-sync standard
ip subnet-zero
no ip finger
!
cns event-service server
!
<...output truncated...>
!
interface FastEthernet3/3
  ip address 172.20.52.19 255.255.255.224
!
<...output truncated...>
!
line con 0
  exec-timeout 0 0
  transport input none
line vty 0 4
  exec-timeout 0 0
  password lab
  login
  transport input lat pad mop telnet rlogin udptn nasi
!
end
Router#
```

Saving the Running Configuration Settings

To store the configuration or changes to your startup configuration in NVRAM, enter the **copy running-config startup-config** command at the privileged EXEC prompt (#) as follows:

```
Router# copy running-config startup-config
```

This command saves the configuration settings that you created in configuration mode. If you fail to do this step, your configuration will be lost the next time you reload the system.

Reviewing the Configuration

To display information stored in NVRAM, enter the **show startup-config** EXEC command. The display should be similar to the display from the **show running-config** EXEC command.

Configuring a Default Gateway



Note

The switch uses the default gateway only when it is not configured with a routing protocol.

To send data to another subnet when the switch is not configured with a routing protocol, configure a default gateway. The default gateway must be the IP address of an interface on a router in the same subnet.

To configure a default gateway, perform this task:

	Command	Purpose
Step 1	Router(config)# ip default-gateway <i>A.B.C.D</i>	Configures a default gateway.
Step 2	Router# show ip route	Verifies that the default gateway appears correctly in the IP routing table.

This example shows how to configure a default gateway and how to verify the configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip default-gateway 172.20.52.35
Router(config)# end
3d17h: %SYS-5-CONFIG_I: Configured from console by console
Router# show ip route
Default gateway is 172.20.52.35

Host                Gateway                Last Use    Total Uses  Interface
ICMP redirect cache is empty
Router#
```

Configuring a Static Route

If your Telnet station or SNMP network management workstation is on a different network from your switch and a routing protocol has not been configured, you might need to add a static routing table entry for the network where your end station is located.

To configure a static route, perform this task:

	Command	Purpose
Step 1	Router(config)# ip route <i>dest_IP_address mask</i> { <i>forwarding_IP</i> vlan <i>vlan_ID</i> }	Configures a static route.
Step 2	Router# show running-config	Verifies the static route configuration.

This example shows how to use the **ip route** command to configure a static route to a workstation at IP address 171.10.5.10 on the switch with a subnet mask and IP address 172.20.3.35 of the forwarding router:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip route 171.10.5.10 255.255.255.255 172.20.3.35
Router(config)# end
Router#
```


This example shows how to use the **show running-config** command to confirm the configuration of the previously configured static route:

```
Router# show running-config
Building configuration...
.
<...output truncated...>
.
ip default-gateway 172.20.52.35
ip classless
ip route 171.10.5.10 255.255.255.255 172.20.3.35
no ip http server
!
line con 0
  transport input none
line vty 0 4
  exec-timeout 0 0
  password lab
  login
  transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end
```

Router#

This example shows how to use the **ip route** command to configure a static route to a workstation at IP address 171.20.5.3 on the switch with subnet mask and connected over VLAN 1:

```
Router# configure terminal
Router(config)# ip route 171.20.5.3 255.255.255.255 vlan 1
Router(config)# end
Router#
```

This example shows how to use the **show running-config** command to confirm the configuration of the previously configured static route:

```
Router# show running-config
Building configuration...
.
<...output truncated...>
.
ip default-gateway 172.20.52.35
ip classless
ip route 171.20.52.3 255.255.255.255 Vlan1
no ip http server
!
!
x25 host z
!
line con 0
  transport input none
line vty 0 4
  exec-timeout 0 0
  password lab
  login
  transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end
```

Router#

Configuring a BOOTP Server

The Bootstrap Protocol (BOOTP) automatically assigns an IP address by adding the MAC and IP addresses of the interface to the BOOTP server configuration file. When the switch boots, it automatically retrieves the IP address from the BOOTP server.

The switch performs a BOOTP request *only* if the current IP address is set to 0.0.0.0. (This address is the default address for a new switch or a switch that has had its startup-config file cleared using the **erase** command.)

To allow your switch to retrieve its IP address from a BOOTP server, you must first determine the MAC address of the switch and add that MAC address to the BOOTP configuration file on the BOOTP server. To create a BOOTP server configuration file, follow these steps:

-
- Step 1** Install the BOOTP server code on the workstation, if it is not already installed.
 - Step 2** Determine the MAC address from the label on the chassis.
 - Step 3** Add an entry in the BOOTP configuration file (usually `/usr/etc/bootptab`) for each switch. Press **Return** after each entry to create a blank line between each entry. See the example BOOTP configuration file that follows in Step 4.
 - Step 4** Enter the **reload** command to reboot and automatically request the IP address from the BOOTP server.

This example BOOTP configuration file shows the added entry:

```
# /etc/bootptab: database for bootp server (/etc/bootpd)
#
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
#     first field -- hostname
#                   (may be full domain name and probably should be)
#
#     hd -- home directory
#     bf -- bootfile
#     cs -- cookie servers
#     ds -- domain name servers
#     gw -- gateways
#     ha -- hardware address
#     ht -- hardware type
#     im -- impress servers
#     ip -- host IP address
#     lg -- log servers
#     lp -- LPR servers
#     ns -- IEN-116 name servers
#     rl -- resource location protocol servers
#     sm -- subnet mask
#     tc -- template host (points to similar host entry)
#     to -- time offset (seconds)
#     ts -- time servers
#
<information deleted>
#
#####
# Start of individual host entries
#####
Router:      tc=netcisco0:   ha=0000.0ca7.ce00:   ip=172.31.7.97:
dross:      tc=netcisco0:   ha=00000c000139:   ip=172.31.7.26:
<information deleted>
```

Protecting Access to Privileged EXEC Commands

The following tasks provide a way to control access to the system configuration file and privileged EXEC commands:

- [Setting or Changing a Static Enable Password, page 3-15](#)
- [Using the enable password and enable secret Commands, page 3-15](#)
- [Setting or Changing a Line Password, page 3-16](#)
- [Setting TACACS+ Password Protection for Privileged EXEC Mode, page 3-16](#)
- [Encrypting Passwords, page 3-17](#)
- [Configuring Multiple Privilege Levels, page 3-17](#)

Setting or Changing a Static Enable Password

To set or change a static password that controls access to the privileged EXEC mode, perform this task:

Command	Purpose
Router(config)# enable password <i>password</i>	Sets a new password or changes an existing password for the privileged EXEC mode.

This example shows how to configure an enable password as “lab” at the privileged EXEC mode:

```
Router# configure terminal
Router(config)# enable password lab
Router(config)#
```

To display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration” section on page 3-19](#).

Using the enable password and enable secret Commands

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a TFTP server, you can use either the **enable password** or **enable secret** commands. Both commands configure an encrypted password that you must enter to access enable mode (the default) or to access a specified privilege level. We recommend that you use the **enable secret** command.

If you configure the **enable secret** command, it takes precedence over the **enable password** command; the two commands cannot be in effect simultaneously.

To configure the switch to require an enable password, perform either of these tasks:

Command	Purpose
Router(config)# enable password [level <i>level</i>] { <i>password</i> <i>encryption-type encrypted-password</i> }	Establishes a password for the privileged EXEC mode.
Router(config)# enable secret [level <i>level</i>] { <i>password</i> <i>encryption-type encrypted-password</i> }	Specifies a secret password, saved using a nonreversible encryption method. (If enable password and enable secret commands are both set, users must enter the enable secret password.)

Use either of these commands with the **level** option to define a password for a specific privilege level. After you specify the level and set a password, give the password only to users who need to have access at this level. Use the **privilege level** configuration command to specify commands accessible at various levels.

If you enable the **service password-encryption** command, the password you enter is encrypted. When you display it with the **more system:running-config** command, it displays in encrypted form.

If you specify an encryption type, you must provide an encrypted password that you copy from another Catalyst 6500 series switch configuration.

**Note**

You cannot recover a lost encrypted password. You must clear NVRAM and set a new password. See the “[Recovering a Lost Enable Password](#)” section on page 3-19 if you lose or forget your password.

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

Setting or Changing a Line Password

To set or change a password on a line, perform this task:

Command	Purpose
Router(config-line)# password <i>password</i>	Sets a new password or change an existing password for the privileged level.

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

Setting TACACS+ Password Protection for Privileged EXEC Mode

For complete information about TACACS+, refer to these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.1, “Authentication, Authorization, and Accounting (AAA),” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt1/index.htm
- *Cisco IOS Security Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

To set the TACACS+ protocol to determine whether or not a user can access privileged EXEC mode, perform this task:

Command	Purpose
Router(config)# enable use-tacacs	Sets the TACACS-style user ID and password-checking mechanism for the privileged EXEC mode.

When you set TACACS password protection at the privileged EXEC mode, the **enable** EXEC command prompts for both a new username and a password. This information is then sent to the TACACS+ server for authentication. If you are using the extended TACACS+, it also sends any existing UNIX user identification code to the TACACS+ server.

**Caution**

If you enter the **enable use-tacacs** command, you must also enter **tacacs-server authenticate enable**, or you are locked out of the privileged EXEC mode.

**Note**

When used without extended TACACS, the **enable use-tacacs** command allows anyone with a valid username and password to access the privileged EXEC mode, creating a potential security problem. This problem occurs because the switch cannot tell the difference between a query resulting from entering the **enable** command and an attempt to log in without extended TACACS.

Encrypting Passwords

Because protocol analyzers can examine packets (and read passwords), you can increase access security by configuring the Cisco IOS software to encrypt passwords. Encryption prevents the password from being readable in the configuration file.

To configure the Cisco IOS software to encrypt passwords, perform this task:

Command	Purpose
Router(config)# service password-encryption	Encrypts a password.

Encryption occurs when the current configuration is written or when a password is configured. Password encryption is applied to all passwords, including authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and Border Gateway Protocol (BGP) neighbor passwords. The **service password-encryption** command keeps unauthorized individuals from viewing your password in your configuration file.

**Caution**

The **service password-encryption** command does not provide a high level of network security. If you use this command, you should also take additional network security measures.

Although you cannot recover a lost encrypted password (that is, you cannot get the original password back), you can regain control of the switch after you lose or forget the encrypted password. See the [“Recovering a Lost Enable Password”](#) section on page 3-19 if you lose or forget your password.

To display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration”](#) section on page 3-19.

Configuring Multiple Privilege Levels

By default, the Cisco IOS software has two modes of password security: user EXEC mode and privileged EXEC mode. You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

For example, if you want many users to have access to the **clear line** command, you can assign it level 2 security and distribute the level 2 password widely. If you want more restricted access to the **configure** command, you can assign it level 3 security and distribute that password to more restricted users.

These tasks describe how to configure additional levels of security:

- [Setting the Privilege Level for a Command, page 3-18](#)
- [Changing the Default Privilege Level for Lines, page 3-18](#)
- [Logging In to a Privilege Level, page 3-18](#)
- [Exiting a Privilege Level, page 3-19](#)
- [Displaying the Password, Access Level, and Privilege Level Configuration, page 3-19](#)

Setting the Privilege Level for a Command

To set the privilege level for a command, perform this task:

	Command	Purpose
Step 1	Router(config)# privilege <i>mode</i> level <i>level</i> <i>command</i>	Sets the privilege level for a command.
Step 2	Router(config)# enable password <i>level</i> <i>level</i> <i>[encryption-type]</i> <i>password</i>	Specifies the enable password for a privilege level.

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

Changing the Default Privilege Level for Lines

To change the default privilege level for a given line or a group of lines, perform this task:

Command	Purpose
Router(config-line)# privilege <i>level</i> <i>level</i>	Changes the default privilege level for the line.

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

Logging In to a Privilege Level

To log in at a specified privilege level, perform this task:

Command	Purpose
Router# enable <i>level</i>	Logs into a specified privilege level.

Exiting a Privilege Level

To exit to a specified privilege level, perform this task:

Command	Purpose
Router# disable <i>level</i>	Exits to a specified privilege level.

Displaying the Password, Access Level, and Privilege Level Configuration

To display the password, access level, and privilege level configuration, perform this task:

	Command	Purpose
Step 1	Router# show running-config	Displays the password and the access level configuration.
Step 2	Router# show privilege	Shows the privilege level configuration.

This example shows how to display the password and access level configuration:

```
Router# show running-config
<...output truncated...>
enable password lab
<...output truncated...>
```

This example shows how to display the privilege level configuration:

```
Router# show privilege
Current privilege level is 15
Router#
```

Recovering a Lost Enable Password

To recover a lost enable password, follow these steps:

-
- Step 1** Connect to the console interface.
 - Step 2** Configure the switch to boot up without reading the configuration memory (NVRAM).
 - Step 3** Reboot the system.
 - Step 4** Access enable mode (which can be done without a password when one is not configured).
 - Step 5** View or change the password, or erase the configuration.
 - Step 6** Reconfigure the switch to boot up and read the NVRAM as it normally does.
 - Step 7** Reboot the system.
-



Note

Password recovery requires the Break signal. You must be familiar with how your terminal or PC terminal emulator issues this signal. For example, in ProComm, the Alt-B keys generate the Break signal. In a Windows terminal session, you press the **Break** or **Ctrl** and **Break** keys simultaneously.

Modifying the Supervisor Engine Startup Configuration

These sections describe how the startup configuration on the supervisor engine works and how to modify the configuration register and BOOT variable:

- [Understanding the Supervisor Engine Boot Configuration, page 3-20](#)
- [Configuring the Software Configuration Register, page 3-21](#)
- [Specifying the Startup System Image, page 3-24](#)
- [Understanding Flash Memory, page 3-24](#)
- [BOOTLDR Environment Variable, page 3-25](#)
- [CONFIG_FILE Environment Variable, page 3-26](#)
- [Controlling Environment Variables, page 3-26](#)

Understanding the Supervisor Engine Boot Configuration

These next sections describe how the boot configuration works on the supervisor engine.

Understanding the Supervisor Engine Boot Process

The supervisor engine boot process involves two software images: ROM monitor and supervisor engine software. When the switch is powered up or reset, the ROM-monitor code is executed. Depending on the NVRAM configuration, the supervisor engine either stays in ROM-monitor mode or loads the supervisor engine software.

Two user-configurable parameters determine how the switch boots: the configuration register and the BOOT environment variable. The configuration register is described in the [“Modifying the Boot Field and Using the boot Command”](#) section on page 3-22. The BOOT environment variable is described in the [“Specifying the Startup System Image”](#) section on page 3-24.

Understanding the ROM Monitor

The ROM monitor executes upon power-up, reset, or when a fatal exception occurs. The switch enters ROM-monitor mode if the switch does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROM-monitor mode. From ROM-monitor mode, you can manually load a software image from bootflash or a Flash PC card.



Note

For complete syntax and usage information for the ROM monitor commands, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

You can also enter ROM-monitor mode by restarting and then pressing the **Break** key during the first 60 seconds of startup. If you are connected through a terminal server, you can escape to the Telnet prompt and enter the **send break** command to enter ROM-monitor mode.



Note

The **Break** key is always enabled for 60 seconds after rebooting, regardless of whether the configuration-register setting has the **Break** key disabled.

The ROM monitor has these features:

- Power-on confidence test
- Hardware initialization
- Boot capability (manual boot and autoboot)
- Debug utility and crash analysis
- Monitor call interface (EMT calls—the ROM monitor provides information and some functionality to the running software images through EMT calls)
- File system (the ROM monitor knows the simple file system and supports the newly developed file system through the dynamic linked file system library [MONLIB])
- Exception handling

Configuring the Software Configuration Register

The switch uses a 16-bit software configuration register, which allows you to set specific system parameters. Settings for the software configuration register are written into NVRAM.

Following are some reasons for changing the software configuration register settings:

- To select a boot source and default boot filename.
- To enable or disable the Break function.
- To control broadcast addresses.
- To set the console terminal baud rate.
- To load operating software from Flash memory.
- To recover a lost password.
- To allow you to manually boot the system using the **boot** command at the bootstrap program prompt.
- To force an automatic boot from the system bootstrap software (boot image) or from a default system image in onboard Flash memory, and read any **boot system** commands that are stored in the configuration file in NVRAM.

[Table 3-2](#) lists the meaning of each of the software configuration memory bits, and [Table 3-3](#) defines the boot field.



Caution

The recommended configuration register setting is 0x2102. If you configure a setting that leaves break enabled and you send a break sequence over a console connection, the switch drops into ROMMON.

Table 3-2 Software Configuration Register Bit Meaning

Bit Number ¹	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 3-3)
06	0x0040	Causes system software to ignore NVRAM contents
07	0x0080	OEM ² bit enabled
08	0x0100	Break disabled
09	0x0200	Use secondary bootstrap
10	0x0400	Internet Protocol (IP) broadcast with all zeros

Table 3-2 Software Configuration Register Bit Meaning (continued)

Bit Number ¹	Hexadecimal	Meaning
11 to 12	0x0800 to 0x1000	Console line speed (default is 9600 baud)
13	0x2000	Boot default Flash software if network boot fails
14	0x4000	IP broadcasts do not have network numbers
15	0x8000	Enable diagnostic messages and ignore NVRAM contents

1. The factory default value for the configuration register is 0x2102.
2. OEM = original equipment manufacturer.

Table 3-3 Explanation of Boot Field (Configuration Register Bits 00 to 03)

Boot Field	Meaning
00	Stays at the system bootstrap prompt
01	Boots the first system image in onboard Flash memory
02 to 0F	Specifies a default filename for booting over the network; enables boot system commands that override the default filename

Modifying the Boot Field and Using the boot Command

The configuration register boot field determines whether or not the switch loads an operating system image, and if so, where it obtains this system image. The following sections describe using and setting the configuration register boot field, and the tasks you must perform to modify the configuration register boot field.

Bits 0 through 3 of the software configuration register form the boot field.



Note

The factory default configuration register setting for systems and spares is 0x2102.

When the boot field is set to either 0 or 1 (0-0-0-0 or 0-0-0-1), the system ignores any boot instructions in the system configuration file and the following occurs:

- When the boot field is set to 0, you must boot the operating system manually by entering the **boot** command to the system bootstrap program or ROM monitor.
- When the boot field is set to 1, the system boots the first image in the onboard bootflash single in-line memory module (SIMM).
- When the entire boot field equals a value between 0-0-1-0 and 1-1-1-1, the switch loads the system image specified by **boot system** commands in the startup configuration file.

You can enter the **boot** command only, or enter the command and include additional boot instructions, such as the name of a file stored in Flash memory, or a file that you specify for booting from a network server. If you use the **boot** command without specifying a file or any other boot instructions, the system boots from the default Flash image (the first image in onboard Flash memory). Otherwise, you can instruct the system to boot from a specific Flash image (using the **boot system flash filename** command).

You can also use the **boot** command to boot images stored in the Flash PC cards located in Flash PC card slot 0 or slot 1 on the supervisor engine. If you set the boot field to any bit pattern other than 0 or 1, the system uses the resulting number to form a filename for booting over the network.

You must set the boot field for the boot functions you require.

Modifying the Boot Field

You modify the boot field from the software configuration register. To modify the software configuration register boot field, perform this task:

	Command	Purpose
Step 1	Router# show version	Determines the current configuration register setting.
Step 2	Router# configure terminal	Enters configuration mode, selecting the terminal option.
Step 3	Router(config)# config-register value	Modifies the existing configuration register setting to reflect the way in which you want the switch to load a system image.
Step 4	Router(config)# end	Exits configuration mode.
Step 5	Router# reload	Reboots to make your changes take effect.

To modify the configuration register while the switch is running Cisco IOS, follow these steps:

-
- Step 1** Enter the **enable** command and your password to enter privileged level as follows:
- ```
Router> enable
Password:
Router#
```
- Step 2** Enter the **configure terminal** command at the EXEC mode prompt (#) as follows:
- ```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```
- Step 3** Configure the configuration register to 0x2102 as follows:
- ```
Router(config)# config-register 0x2102
```
- Set the contents of the configuration register by entering the **config-register value** configuration command, where *value* is a hexadecimal number preceded by 0x (see [Table 3-2 on page 3-21](#)).
- Step 4** Enter the **end** command to exit configuration mode. The new value settings are saved to memory; however, the new settings do not take effect until the system software is reloaded by rebooting the system.
- Step 5** Enter the **show version EXEC** command to display the configuration register value currently in effect and that will be used at the next reload. The value is displayed on the last line of the screen display, as in this example:
- ```
Configuration register is 0x141 (will be 0x2102 at next reload)
```
- Step 6** Save your settings. (See the “[Saving the Running Configuration Settings](#)” section on page 3-11. However, note that configuration register changes take effect only after the system reloads, such as when you enter a **reload** command from the console.)
- Step 7** Reboot the system. The new configuration register value takes effect with the next system boot.
-

This completes the procedure for making configuration register changes.

Verifying the Configuration Register Setting

Enter the **show version EXEC** command to verify the current configuration register setting. In ROM-monitor mode, enter the **o** command to verify the value of the configuration register boot field.

To verify the configuration register setting, perform this task:

Command	Purpose
Router# show version include Configuration register	Displays the configuration register setting.

In this example, the **show version** command indicates that the current configuration register is set so that the switch does not automatically load an operating system image. Instead, it enters ROM-monitor mode and waits for user-entered ROM monitor commands. The new setting instructs the switch to load a system image from commands in the startup configuration file or from a default system image stored on a network server.

```
Router1# show version | include Configuration register
Configuration register is 0x2102
Router#
```

Specifying the Startup System Image

You can enter multiple boot commands in the startup configuration file or in the BOOT environment variable to provide backup methods for loading a system image.



Note

Store the system software image in the **sup-bootflash:** or **slot0:** device, not in the **bootflash:** device. Store the boot loader image, if any, in the MSFC **bootflash:** device.

The BOOT environment variable is also described in the “Specify the Startup System Image in the Configuration File” section in the “Loading and Maintaining System Images and Microcode” chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Understanding Flash Memory

The following sections describe Flash memory:

- [Flash Memory Features, page 3-25](#)
- [Security Features, page 3-25](#)
- [Flash Memory Configuration Process, page 3-25](#)



Note

The descriptions in the following sections applies to both the bootflash device and to removable Flash memory cards.

Flash Memory Features

The Flash memory components allow you to do the following:

- Copy the system image to Flash memory using TFTP.
- Copy the system image to Flash memory using rcp.
- Boot the system from Flash memory either automatically or manually.
- Copy the Flash memory image to a network server using TFTP or rcp.
- Boot manually or automatically from a system software image stored in Flash memory.

Security Features

The Flash memory components support the following security features:

- Flash memory cards contain a write-protect switch that you can use to protect data. You must set the switch to *unprotected* to write data to the Flash PC card.
- The system image stored in Flash memory can be changed only from privileged EXEC level on the console terminal.

Flash Memory Configuration Process

To configure your switch to boot from Flash memory, follow these steps:

-
- | | |
|---------------|---|
| Step 1 | Copy a system image to Flash memory using TFTP or rcp (refer to the <i>Cisco IOS Configuration Fundamentals Configuration Guide</i> , Release 12.1, “Cisco IOS File Management,” “Loading and Maintaining System Images,” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/fcprt2/fcd203.htm |
| Step 2 | Configure the system to boot automatically from the file in Flash memory. You might need to change the configuration register value. See the “ Modifying the Boot Field and Using the boot Command ” section on page 3-22, for more information on modifying the configuration register. |
| Step 3 | Save your configurations. |
| Step 4 | Power cycle and reboot your system to ensure that all is working as expected. |
-

BOOTLDR Environment Variable

The BOOTLDR environment specifies the Flash file system and filename containing the boot loader image.



Caution

With a Supervisor Engine 1 and MSFC1, do not erase the boot loader image from the MSFC1 **bootflash:** device. The boot loader must be present in the MSFC1 **bootflash:** device to boot a Supervisor Engine 1 and MSFC1 successfully.

CONFIG_FILE Environment Variable

For Class A Flash file systems, the CONFIG_FILE environment variable specifies the file system and filename of the configuration file to use for initialization (startup). Valid file systems can include **nvr**am:, **slot0**:, and **sup-bootflash**:

For detailed file management configuration information, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/fun_c/index.htm

After you save the CONFIG_FILE environment variable to your startup configuration, the switch checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

The switch uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the switch detects a problem with NVRAM or a checksum error, the switch enters **setup** mode. See the “Using the Setup Facility or the setup Command” section on page 3-2 for more information on the **setup** command facility.

Controlling Environment Variables

Although the ROM monitor controls environment variables, you can create, modify, or view them with certain commands. To create or modify the BOOT, BOOTLDR, and CONFIG_FILE environment variables, use the **boot system**, **boot bootldr**, and **boot config** global configuration commands.

Refer to the “Specify the Startup System Image in the Configuration File” section in the “Loading and Maintaining System Images and Microcode” chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the BOOT environment variable. Refer to the “Specify the Startup Configuration File” section in the “Modifying, Downloading, and Maintaining Configuration Files” chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the CONFIG_FILE variable.



Note

When you use the **boot system**, **boot bootldr**, and **boot config** global configuration commands, you affect only the running configuration. You must save the environment variable settings to your startup configuration to place the information under ROM monitor control and for the environment variables to function as expected. Enter the **copy system:running-config nvram:startup-config** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT, BOOTLDR, and the CONFIG_FILE environment variables using the **show bootvar** command. This command displays the settings for these variables as they exist in the startup configuration as well as in the running configuration if a running configuration setting differs from a startup configuration setting.

This example shows how to check the BOOT, BOOTLDR, and the CONFIG_FILE environment variables:

```
Router# show bootvar
BOOT variable = slot0:c6sup22-jsv-mz.121-5c.EX.bin,1;
CONFIG_FILE variable does not exist
BOOTLDR variable = bootflash:c6msfc2-boot-mz.121-3a.E4
Configuration register is 0x2
Router#
```

To display the contents of the configuration file pointed to by the CONFIG_FILE environment variable, enter the **more nvram:startup-config** command.

Setting the BOOTLDR Environment Variable

To set the BOOTLDR environment variable, perform this task:

	Command	Purpose
Step 1	Router# dir bootflash:	Verifies that bootflash contains the boot loader image.
Step 2	Router# configure terminal	Enters the configuration mode from the terminal.
Step 3	Router(config)# boot bootldr bootflash:boot_loader	Sets the BOOTLDR environment variable to specify the Flash device and filename of the boot loader image. This step modifies the runtime BOOTLDR environment variable.
Step 4	Router# end	Exits configuration mode.
Step 5	Router# copy system:running-config nvruntime:startup-config	Saves this runtime BOOTLDR environment variable to your startup configuration.
Step 6	Router# show bootvar	(Optional) Verifies the contents of the BOOTLDR environment variable.

This example shows how to set the BOOTLDR variable:

```
Router# dir bootflash:
Directory of bootflash:/

 1  -rw-      1599488   Nov 29 1999 11:12:29  c6msfc-boot-mz.120-7.XE.bin

15990784 bytes total (14391168 bytes free)
Router# configure terminal
Router (config)# boot bootldr bootflash:c6msfc-boot-mz.120-7.XE.bin
Router (config)# end
Router# copy system:running-config nvruntime:startup-config
[ok]
Router# show bootvar
BOOT variable = sup-bootflash:c6sup-js-mz.120-7.XE.bin,1;
CONFIG_FILE variable does not exist
BOOTLDR variable = bootflash:c6msfc-boot-mz.120-7.XE.bin
Configuration register is 0x0
```




Configuring EHSA Supervisor Engine Redundancy

With 12.1 E releases earlier than Release 12.1(13)E, the Catalyst 6500 series switch supports dual supervisor engines with EHSA.



Note

EHSA is not supported in Release 12.1(13)E and later releases (see [Chapter 5, “Configuring RPR and RPR+ Supervisor Engine Redundancy,”](#) for information about RPR or RPR+ redundancy in Release 12.1(13)E and later releases.

This chapter consists of these sections:

- [Supervisor Engine Redundant Operation, page 4-1](#)
- [Supervisor Engine Redundancy Requirements, page 4-2](#)
- [Synchronizing the Supervisor Engine Configurations, page 4-3](#)
- [Displaying the Supervisor Engine Redundancy, page 4-4](#)
- [Copying Files to the Redundant Supervisor Engine, page 4-4](#)



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication and the Release 12.1 publications at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

Supervisor Engine Redundant Operation

Catalyst 6500 series switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. The redundant supervisor engine runs in EHSA standby mode.



Note

The EHSA feature is not supervisor engine mirroring or load balancing. Network services are disrupted until the redundant supervisor engine takes over and the switch recovers.

EHSA standby mode provides the following features:

- Auto-startup and bootvar synchronization between active and redundant supervisor engines
- Hardware signals that detect and decide the active or redundant status of supervisor engines
- Clock synchronization every 60 seconds from the active to the redundant supervisor engine
- A redundant supervisor engine that is booted but not all subsystems are up: if the active supervisor engine fails, the redundant supervisor engine becomes fully operational
- An operational supervisor engine present in place of the failed unit becomes the redundant supervisor engine


Note

The two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.

When the switch is powered on, EHSA runs between the two supervisor engines. The supervisor engine that boots first, either in slot 1 or 2, becomes the EHSA active supervisor engine. The Multilayer Switch Feature Card (MSFC or MSFC2) and Policy Feature Card (PFC or PFC2) become fully operational. The MSFC and PFC on the redundant supervisor engine come out of reset but are not operational.

The following events cause an EHSA switchover:

- Clock synchronization failure between supervisor engines
- MSFC or PFC failure on the active supervisor engine

In a switchover, the redundant supervisor engine becomes fully operational and the following occurs:

- All switching modules power up again
- Remaining subsystems on the MSFC (including Layer 2 and Layer 3 protocols) are brought up
- Access control lists (ACLs) are reprogrammed into supervisor engine hardware


Note

In a switchover, there is a disruption of traffic because some address states are lost and then restored after they are dynamically redetermined.

Supervisor Engine Redundancy Requirements

For redundant operation, the following requirements must be met:

- The active and redundant supervisor engines must be in slots 1 and 2.
- Each supervisor engine must have the resources to run the switch on its own, which means all supervisor engine resources are duplicated. In other words, each supervisor engine has its own Flash device and console port connections.


Note

Make a separate console connection to each supervisor engine. Do not connect a “Y” cable to the console ports.

- Both supervisor engines must have the same system image (see the [“Copying Files to the Redundant Supervisor Engine”](#) section on page 4-4).

**Note**

If the redundant supervisor engine is running Catalyst operating system software, remove the active supervisor engine and boot the switch with only the redundant supervisor engine installed. Follow the procedures in the release notes to convert the redundant supervisor engine from Catalyst operating system software.

- The configuration register in the startup-config must be set to autoboot (see the “[Modifying the Boot Field](#)” section on page 3-23).

**Note**

EHSA does not support booting from the network.

If these requirements are met, the switch functions in EHSA mode by default.

Synchronizing the Supervisor Engine Configurations

During normal operation, the startup-config and config-registers configuration are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

**Note**

The boot variables are not synchronized by default.

To manually synchronize the configurations used by the two supervisor engines, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Router(config)# redundancy	Enters redundancy configuration mode.
Step 2	Router(config-r)# main-cpu	Enters main-cpu configuration submode.
Step 3	Router(config-r-mc)# auto-sync {startup-config config-register bootvar standard}	Synchronizes the configuration elements.
Step 4	Router(config-r-mc)# end	Returns to privileged EXEC mode.
Step 5	Router# copy running-config startup-config	Forces a manual synchronization of the configuration files in NVRAM. Note This step is not required to synchronize the running configuration file in DRAM.

**Note**

The **auto-sync standard** command does not synchronize the boot variables.

This example shows how to reenables the default automatic synchronization feature using the **auto-sync standard** command to synchronize the startup-config and config-register configuration of the active supervisor engine with the redundant supervisor engine:

```
Router(config)# redundancy
Router(config-r)# main-cpu
Router(config-r-mc)# auto-sync standard
Router(config-r-mc)# auto-sync bootvar
```

```
Router(config-r-mc)# end
Router# copy running-config startup-config
```

**Note**

To manually synchronize only individual elements of the standard auto-sync configuration, disable the default automatic synchronization feature.

This example shows how to disable default automatic synchronization and only allow automatic synchronization of the config-registers of the active supervisor engine to the redundant supervisor engine while disallowing synchronization of the startup configuration:

```
Router(config)# redundancy
Router(config-r)# main-cpu
Router(config-r-mc)# no auto-sync standard
Router(config-r-mc)# auto-sync config-register
Router(config-r-mc)# end
Router# copy running-config startup-config
```

Displaying the Supervisor Engine Redundancy

To display both supervisor engines, perform this task:

Command	Purpose
Router# show module all	Displays the redundancy configuration.

This example shows how to display the supervisor engine redundancy configuration:

```
Router# show module all
Mod Ports Card Type
-----
 1    2 Catalyst 6000 supervisor 2 (Active)
 5   48 48 port 10/100 mb RJ-45 ethernet
Model
-----
WS-X6K-SUP2-2GE
WS-X6248-RJ-45
Serial No.
-----
SAD0620046D
SAD03181291

Mod MAC addresses
-----
 1 0001.c9db.3788 to 0001.c9db.3789
 5 0050.f0ac.3054 to 0050.f0ac.3083
Hw Fw Sw Status
-----
 3.7 6.1(3) 7.5(0.6)HUB6 Ok
 1.0 4.2(0.24)VAI 7.5(0.6)HUB6 Ok

Mod Sub-Module
-----
 1 Policy Feature Card 2
 1 Cat6k MSFC 2 daughterboard
Model
-----
WS-F6K-PFC2
WS-F6K-MSFC2
Serial
-----
SAD06200415
SAD06210067
Hw Status
-----
 3.2 Ok
 2.3 Ok

Mod Online Diag Status
-----
 1 Pass
 5 Pass
Router#
```

Copying Files to the Redundant Supervisor Engine

Use the following command to copy a file to the **slot0:** device on a redundant supervisor engine:

```
Router# copy source_device:source_filename slaveslot0:target_filename
```

Use the following command to copy a file to the **bootflash:** device on a redundant supervisor engine:

```
Router# copy source_device:source_filename slavesup-bootflash:target_filename
```

Use the following command to copy a file to the **bootflash:** device on a redundant MSFC:

```
Router# copy source_device:source_filename slavebootflash:target_filename
```




Configuring RPR and RPR+ Supervisor Engine Redundancy

Release 12.1(13)E and later releases support supervisor engine redundancy with Route Processor Redundancy (RPR) and Route Processor Redundancy Plus (RPR+). This chapter describes how to configure supervisor engine redundancy with RPR and RPR+.



Note

Enhanced high system availability (EHSA) is not supported in Release 12.1(13)E and later releases.

This chapter consists of these sections:

- [Understanding Supervisor Engine Redundancy, page 5-1](#)
- [Supervisor Engine Redundancy Guidelines and Restrictions, page 5-4](#)
- [Configuring Supervisor Engine Redundancy, page 5-6](#)
- [Performing a Fast Software Upgrade, page 5-9](#)
- [Copying Files to an MSFC, page 5-10](#)

Understanding Supervisor Engine Redundancy

These sections describe supervisor engine redundancy:

- [Supervisor Engine Redundancy Overview, page 5-1](#)
- [RPR+ Operation, page 5-2](#)
- [Supervisor Engine Synchronization, page 5-3](#)

Supervisor Engine Redundancy Overview

Catalyst 6500 series switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. RPR supports a switchover time of 2 to 4 minutes and RPR+ supports a switchover time of 30 to 60 seconds.

When RPR+ mode is used, the redundant supervisor engine is fully initialized and configured, which shortens the switchover time. The active supervisor engine checks the image version of the redundant supervisor engine when the redundant supervisor engine comes online. If the image on the redundant supervisor engine does not match the image on the active supervisor engine, RPR redundancy mode is used.

RPR Operation

RPR supports the following features:

- Auto-startup and bootvar synchronization between active and redundant supervisor engines
- Hardware signals that detect and decide the active or redundant status of supervisor engines
- Clock synchronization every 60 seconds from the active to the redundant supervisor engine
- A redundant supervisor engine that is booted but not all subsystems are up: if the active supervisor engine fails, the redundant supervisor engine become fully operational
- An operational supervisor engine present in place of the failed unit becomes the redundant supervisor engine
- Support for fast software upgrade (FSU) (See the [“Performing a Fast Software Upgrade”](#) section on page 5-9).



Note

When a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.

When the switch is powered on, RPR runs between the two supervisor engines. The supervisor engine that boots first, either in slot 1 or 2, becomes the RPR active supervisor engine. The Multilayer Switch Feature Card (MSFC or MSFC2) and Policy Feature Card (PFC or PFC2) become fully operational. The MSFC and PFC on the redundant supervisor engine come out of reset but are not operational.

The following events cause an RPR switchover:

- Clock synchronization failure between supervisor engines
- MSFC or PFC failure on the active supervisor engine
- A manual switchover.

In a switchover, the redundant supervisor engine becomes fully operational and the following occurs:

- All switching modules power up again
- Remaining subsystems on the MSFC (including Layer 2 and Layer 3 protocols) are brought up
- Access control lists (ACLs) are reprogrammed into supervisor engine hardware



Note

In a switchover, there is a disruption of traffic because some address states are lost and then restored after they are dynamically redetermined.

RPR+ Operation

With RPR+, the redundant supervisor engine is fully initialized and configured, which shortens the switchover time if the active supervisor engine fails or if a manual switchover is performed.

When the switch is powered on, RPR+ runs between the two supervisor engines. The supervisor engine that boots first, either in slot 1 or 2, becomes the active supervisor engine. The Multilayer Switch Feature Card (MSFC or MSFC2) and Policy Feature Card (PFC or PFC2) become fully operational. The MSFC and PFC on the redundant supervisor engine come out of reset but are not operational.

RPR+ enhances RPR by providing the following additional benefits:

- Reduced switchover time

Depending on the configuration, the switchover time is in the range of 30 to 60 seconds.

- Installed modules are not reloaded

Because both the startup configuration and the running configuration are continually synchronized from the active to the redundant supervisor engine, installed modules are not reloaded during a switchover.

- Online insertion and removal (OIR) of the redundant supervisor engine

RPR+ allows OIR of the redundant supervisor engine for maintenance. When the redundant supervisor engine is inserted, the active supervisor engine detects its presence and begins to transition the redundant supervisor engine to fully initialized state.

- Synchronization of OIR events
- Manual user-initiated switchover using the **redundancy force-switchover** command

The following events cause an RPR+ switchover:

- Clock synchronization failure between supervisor engines
- MSFC or PFC failure on the active supervisor engine

Supervisor Engine Synchronization

During RPR mode operation, the startup-config and config-registers configuration are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.



Note

Unless auto-synchronization has been disabled, the boot variables are synchronized by default.

When a redundant supervisor engine configuration is running in RPR+ mode, the following operations trigger synchronization:

- When a redundant supervisor engine first comes online, the configuration information is synchronized in bulk from the active supervisor engine to the redundant supervisor engine. This synchronization overwrites any existing startup configuration file on the redundant supervisor engine.
- When configuration changes occur during normal operation, RPR+ performs an incremental synchronization from the active supervisor engine to the redundant supervisor engine. RPR+ synchronizes user-entered CLI commands incrementally line-by-line from the active supervisor engine to the redundant supervisor engine.



Note

- Even though the redundant supervisor engine is fully initialized, it only interacts with the active supervisor engine to receive incremental changes to the configuration files as they occur. You cannot enter CLI commands on the redundant supervisor engine.
 - Synchronization of the startup configuration file is enabled by default in RPR+ mode.
-

Supervisor Engine Redundancy Guidelines and Restrictions

These sections describe supervisor engine redundancy configuration guidelines and restrictions:

- [RPR+ Guidelines and Restrictions, page 5-4](#)
- [Hardware Configuration Guidelines and Restrictions, page 5-5](#)
- [Configuration Mode Restrictions, page 5-6](#)

RPR+ Guidelines and Restrictions

The following guidelines and restrictions apply to RPR+:

Restrictions

- RPR+ redundancy does not support configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy (see [Chapter 9, “Configuring VLANs”](#)).
- Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. Enter a **copy running-config startup-config** command to synchronize the configuration on the redundant supervisor engine.
- Supervisor engine redundancy does not provide supervisor engine mirroring or supervisor engine load balancing. Only one supervisor engine is active. Network services are disrupted until the redundant supervisor engine takes over and the switch recovers.
- With RPR+, both supervisor engines must run the same version of Cisco IOS software. If the supervisor engines are not running the same version of Cisco IOS software, the redundant supervisor engine comes online in RPR mode.
- The Forwarding Information Base (FIB) tables are cleared on a switchover. As a result, routed traffic is interrupted until route tables reconverge.
- Static IP routes are maintained across a switchover because they are configured from entries in the configuration file.
- Information about dynamic states maintained on the active supervisor engine is not synchronized to the redundant supervisor engine and is lost on switchover.

These are examples of dynamic state information that is lost at switchover:

- Frame Relay switched virtual circuits (SVCs)



Note Frame Relay-switched DLCI information is maintained across a switchover because Frame Relay-switched DLCI configuration is in the configuration file.

- All terminated PPP sessions
- All ATM SVC information
- All terminated TCP and other connection-oriented Layer 3 and Layer 4 sessions
- BGP sessions
- All Automatic Protection System (APS) state information

Guidelines

- The two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.
- RPR+ switchover takes place after the failed supervisor engine completes a core dump. A core dump can take up to 15 minutes. To get faster switchover time, disable core dump on the supervisor engines.

Hardware Configuration Guidelines and Restrictions

For redundant operation, the following hardware configuration guidelines and restrictions apply:

Restrictions

- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file** *file_name* command on a switch that has a redundant supervisor engine.
- Cisco IOS running on the supervisor engine and the MSFC supports redundant configurations where the supervisor engines and MSFC routers are identical. If they are not identical, one will boot first and become active and hold the other supervisor engine and MSFC in a reset condition.
- Each supervisor engine must have the resources to run the switch on its own, which means all supervisor engine resources are duplicated. In other words, each supervisor engine has its own Flash device and console port connections.

Guidelines

- Make separate console connections to each supervisor engine. Do not connect a Y cable to the console ports.
- Both supervisor engines must have the same system image (see the [“Copying Files to an MSFC” section on page 5-10](#)).



Note If the redundant supervisor engine is running Catalyst operating system software, remove the active supervisor engine and boot the switch with only the redundant supervisor engine installed. Follow the procedures in the current release notes to convert the redundant supervisor engine from Catalyst software.

- The configuration register in the startup-config must be set to autoboot (see the [“Modifying the Boot Field” section on page 3-23](#)).
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.



Note

There is no support for booting from the network.

Configuration Mode Restrictions

The following configuration restrictions apply during the startup synchronization process:

- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

```
Config mode locked out till standby initializes
```
- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

- [Configuring RPR and RPR+, page 5-6](#)
- [Synchronizing the Supervisor Engine Configurations, page 5-7](#)
- [Displaying the Redundancy States, page 5-8](#)

Configuring RPR and RPR+

To configure RPR or RPR+, perform this task:

	Command	Purpose
Step 1	Router(config)# redundancy	Enters redundancy configuration mode.
Step 2	Router(config-red)# mode {rpr rpr-plus}	Configures RPR or RPR+. When this command is entered, the redundant supervisor engine is reloaded and begins to work in RPR or RPR+ mode.
Step 3	Router# show running-config	Verifies that RPR or RPR+ is enabled.
Step 4	Router# show redundancy states	Displays the operating redundancy mode.

This example shows how to configure the system for RPR+ and display the redundancy state:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# redundancy
Router(config-red)# mode rpr-plus
Router(config-red)# ^Z
Router# show redundancy states
    my state = 13 -ACTIVE
    peer state = 1 -DISABLED
    Mode = Simplex
    Unit = Primary
    Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy Plus
Redundancy Mode (Configured)  = Route Processor Redundancy Plus
    Split Mode = Disabled
    Manual Swact = Disabled Reason: Simplex mode
    Communications = Down Reason: Simplex mode
```

```

client count = 11
client_notification_TMR = 30000 milliseconds
  keep_alive TMR = 4000 milliseconds
  keep_alive count = 0
keep_alive threshold = 7
  RF debug mask = 0x0

Router#

```

Synchronizing the Supervisor Engine Configurations

During normal operation, the startup-config and config-registers configuration are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

To manually synchronize the configurations used by the two supervisor engines, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Router(config)# redundancy	Enters redundancy configuration mode.
Step 2	Router(config-red)# main-cpu	Enters main-cpu configuration submenu.
Step 3	Router(config-r-mc)# auto-sync { startup-config config-register bootvar standard }	Synchronizes the configuration elements.
Step 4	Router(config-r-mc)# end	Returns to privileged EXEC mode.
Step 5	Router# copy running-config startup-config	Forces a manual synchronization of the configuration files in NVRAM. Note This step is not required to synchronize the running configuration file in DRAM.



Note

The **auto-sync standard** command does not synchronize the boot variables.

This example shows how to reenable the default automatic synchronization feature using the **auto-sync standard** command to synchronize the startup-config and config-register configuration of the active supervisor engine with the redundant supervisor engine:

```

Router(config)# redundancy
Router(config-red)# main-cpu
Router(config-r-mc)# auto-sync standard
Router(config-r-mc)# auto-sync bootvar
Router(config-r-mc)# end
Router# copy running-config startup-config

```



Note

To manually synchronize only individual elements of the standard auto-sync configuration, disable the default automatic synchronization feature.

This example shows how to disable default automatic synchronization and only allow automatic synchronization of the config-registers of the active supervisor engine to the redundant supervisor engine while disallowing synchronization of the startup configuration:

```

Router(config)# redundancy
Router(config-red)# main-cpu

```

```

Router(config-r-mc)# no auto-sync standard
Router(config-r-mc)# auto-sync config-register
Router(config-r-mc)# end
Router# copy running-config startup-config

```

Displaying the Redundancy States

To display the redundancy states, perform this task:

Command	Purpose
Router# show redundancy states	Displays the redundancy states.

This example shows how to display the redundancy states:

```

Router# show redundancy states
my state = 13 -ACTIVE
  peer state = 8 -STANDBY HOT
    Mode = Duplex
    Unit = Primary
    Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy Plus
Redundancy Mode (Configured) = Route Processor Redundancy Plus
  Split Mode = Disabled
  Manual Swact = Enabled
  Communications = Up

  client count = 11
  client_notification_TMR = 30000 milliseconds
  keep_alive TMR = 9000 milliseconds
  keep_alive count = 0
  keep_alive threshold = 18
  RF debug mask = 0x0

Router#

my state = 13 -ACTIVE
  peer state = 1 -DISABLED
    Mode = Simplex
    Unit = Primary
    Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy Plus
Redundancy Mode (Configured) = Route Processor Redundancy Plus
  Split Mode = Disabled
  Manual Swact = Disabled Reason: Simplex mode
  Communications = Down Reason: Simplex mode

  client count = 11
  client_notification_TMR = 30000 milliseconds
  keep_alive TMR = 9000 milliseconds
  keep_alive count = 0
  keep_alive threshold = 18
  RF debug mask = 0x0

```

Performing a Fast Software Upgrade

The fast software upgrade (FSU) procedure supported by RPR allows you to upgrade the Cisco IOS image on the supervisor engines without reloading the system.


Note

If you are performing a first-time upgrade to RPR from EHSA, you must reload both supervisor engines. FSU from EHSA is not supported.

To perform an FSU, perform this task:

	Command	Purpose
Step 1	<pre>Router# copy source_device:source_filename {disk0 disk1}:target_filename Or: Router# copy source_device:source_filename sup-bootflash:target_filename Or: Router# copy source_device:source_filename {slavedisk0 slavedisk1}:target_filename Or: Router# copy source_device:source_filename slavesup-bootflash:target_filename</pre>	<p>Copies the new Cisco IOS image to the disk0: device or the disk1: device on the active supervisor engine.</p> <p>Copies the new Cisco IOS image to the bootflash: device on the active supervisor engine.</p> <p>Copies the new Cisco IOS image to the disk0: device or the disk1: device on the redundant supervisor engine.</p> <p>Copies the new Cisco IOS image to the bootflash: device on the redundant supervisor engine.</p>
Step 2	<pre>Router# config terminal Router(config)# config-register 0x2102 Router(config)# boot system flash device:file_name</pre>	Configures the supervisor engines to boot the new image.
Step 3	<pre>Router# copy running-config start-config</pre>	Saves the configuration.
Step 4	<pre>Router# hw-module {module num} reset</pre>	<p>Reloads the redundant supervisor engine and brings it back online (running the new version of the Cisco IOS software).</p> <p>Note Before reloading the redundant supervisor engine, make sure you wait long enough to ensure that all configuration synchronization changes have completed.</p>
Step 5	<pre>Router# redundancy force-switchover</pre>	<p>Conducts a manual switchover to the redundant supervisor engine. The redundant supervisor engine becomes the new active supervisor engine running the new Cisco IOS image. The modules are reloaded and the module software is downloaded from the new active supervisor engine.</p> <p>The old active supervisor engine reboots with the new image and becomes the redundant supervisor engine.</p> <p>Note To perform an EHSA to RPR FSU, use the reload command in Step 5.</p>

This example shows how to perform an FSU:

```
Router# config terminal
Router(config)# config-register 0x2
Router(config)# boot system flash slot0: c6sup22-jsv-mz.121-11.E
Router# copy running-config start-config
Router# hw-module reset
Router# redundancy force-switchover
Router#
```

Copying Files to an MSFC

Use the following command to copy a file to the **bootflash:** device on an active MSFC:

```
Router# copy source_device:source_filename bootflash:target_filename
```

Use the following command to copy a file to the **bootflash:** device on a redundant MSFC:

```
Router# copy source_device:source_filename slavebootflash:target_filename
```




Configuring Interfaces

This chapter describes how to configure interfaces on the Catalyst 6500 series switches. This chapter consists of these sections:

- [Understanding Interface Configuration, page 6-1](#)
- [Using the Interface Command, page 6-2](#)
- [Configuring a Range of Interfaces, page 6-4](#)
- [Defining and Using Interface-Range Macros, page 6-6](#)
- [Configuring Optional Interface Features, page 6-7](#)
- [Understanding Online Insertion and Removal, page 6-17](#)
- [Monitoring and Maintaining Interfaces, page 6-17](#)



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication and the Release 12.1 publications at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/index.htm>

Understanding Interface Configuration

Many features in the software are enabled on a per-interface basis. When you enter the **interface** command, you must specify the following information:

- Interface type:
 - Ethernet (use the **ethernet** keyword)
 - Fast Ethernet (use the **fastethernet** keyword)
 - Gigabit Ethernet (use the **gigabitethernet** keyword)
 - 10-Gigabit Ethernet (use the **tengigabitethernet** keyword)



Note For WAN interfaces, refer to the configuration note for the WAN module.

- Slot number—The slot in which the module is installed. On the Catalyst 6500 series switch, slots are numbered starting with 1, from top to bottom.

- Port number—The physical port number on the module. On the Catalyst 6500 series switch, the port numbers always begin with 1. When facing the rear of the switch, ports are numbered from the left to the right.

You can identify ports from the physical location. You also can use **show** commands to display information about a specific port, or all the ports.

**Note**

With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Using the Interface Command

**Note**

You use the commands described in this section to configure both physical ports and logical interfaces.

These procedures apply to all interface configuration processes. Begin the interface configuration process in global configuration mode.

- Step 1** Enter the **configure terminal** command at the privileged EXEC prompt to enter global configuration mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

- Step 2** In the global configuration mode, enter the **interfaces** command. Identify the interface type and the number of the connector or interface card.

The following example shows how to select Fast Ethernet, slot 5, interface 1:

```
Router(config)# interfaces fastethernet 5/1
Router(config-if)#
```

- Step 3** Enter the **show interfaces EXEC** command to see a list of all interfaces that are installed. A report is provided for each interface that the device supports, as shown in this display:

```
Router# show interfaces fastethernet 5/48
FastEthernet5/48 is up, line protocol is up
  Hardware is C6k 100Mb 802.3, address is 0050.f0ac.3083 (bia 0050.f0ac.3083)
  Internet address is 172.20.52.18/27
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
    4834677 packets input, 329545368 bytes, 0 no buffer
    Received 4796465 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
  51926 packets output, 15070051 bytes, 0 underruns
    0 output errors, 2 collisions, 2 interface resets
```

```

0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
Router#

```

Step 4 Enter the **show hardware EXEC** command to see a list of the system software and hardware:

```

Router# show hardware
Cisco Internetwork Operating System Software
IOS (tm) c6sup2_rp Software (c6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPLOY)
Synced to mainline version: 12.1(5c)
TAC:Home:Software:Ios General:CiscoIOSRoadmap:12.1
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Wed 28-Mar-01 17:52 by hqluong
Image text-base: 0x30008980, data-base: 0x315D0000

ROM: System Bootstrap, Version 12.1(3r)E2, RELEASE SOFTWARE (fc1)
BOOTFLASH: c6sup2_rp Software (c6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPL)

Router uptime is 2 hours, 55 minutes
System returned to ROM by power-on (SP by power-on)
Running default software

cisco Catalyst 6000 (R7000) processor with 114688K/16384K bytes of memory.
Processor board ID SAD04430J9K
R7000 CPU at 300Mhz, Implementation 39, Rev 2.1, 256KB L2, 1024KB L3 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
1 Virtual Ethernet/IEEE 802.3 interface(s)
48 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)
381K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).
Configuration register is 0x2

Router#

```

Step 5 To begin configuring Fast Ethernet port 5/5, enter the **interface** keyword, interface type, and slot number/port number at the privileged EXEC prompt, as shown in the following example:

```

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/5
Router(config-if)#

```



Note You do not need to add a space between the interface type and interface number. For example, in the preceding line you can specify either *fastethernet 5/5* or *fastethernet5/5*.

Step 6 After each **interface** command, enter the interface configuration commands your particular interface requires.

The commands you enter define the protocols and applications that will run on the interface. The commands are collected and applied to the **interface** command until you enter another **interface** command or press **Ctrl-Z** to get out of interface configuration mode and return to privileged EXEC mode.

Step 7 After you configure an interface, check its status by using the EXEC **show** commands listed in “[Monitoring and Maintaining Interfaces](#)” section on page 6-17.

Configuring a Range of Interfaces

The interface-range configuration mode allows you to configure multiple interfaces with the same configuration parameters. After you enter the interface-range configuration mode, all command parameters you enter are attributed to all interfaces within that range until you exit out of the interface-range configuration mode.

To configure a range of interfaces with the same configuration, perform this task:

Command	Purpose
<pre>Router(config)# interface range {{vlan vlan_ID - vlan_ID [, vlan vlan_ID - vlan_ID]} {type slot/port - port [, type slot/port - port]} {macro_name [, macro_name]}}</pre>	Selects the range of interfaces to be configured.
<pre>Router(config)# no interface range {{vlan vlan_ID - vlan_ID [, vlan vlan_ID - vlan_ID]} {macro_name [, macro_name]}}</pre>	Selects the range of VLAN interfaces to remove.

When configuring a range of interfaces, note the following syntax information:

- For information about macros, see the [“Defining and Using Interface-Range Macros” section on page 6-6](#).
- You can enter up to five comma-separated ranges.
- You are not required to enter spaces before or after the comma.
- With releases earlier than Release 12.1(26)E, you must add a space between the interface numbers and the dash when using the **interface range** command. For example, **interface range fastethernet 1 - 5** is valid syntax; **interface range fastethernet 1-5** is invalid.
- With Release 12.1(26)E and later releases, you do not need to add a space between the interface numbers and the dash when using the **interface range** command.
- With releases earlier than Release 12.1(14)E, the **interface range** command supports these interface keywords:
 - **ethernet**
 - **fastethernet**
 - **gigabitethernet**
 - **tengigabitethernet**
- With Release 12.1(14)E and later releases, the **interface range** command supports these additional interface keywords:
 - **pos**
 - **loopback**
 - **tunnel**
- With Release 12.1(14)E and later releases, you can use the **no interface range** command to delete VLAN interfaces.
- With releases earlier than Release 12.1(14)E, you cannot use the **no** keyword with the **range** keyword to delete VLAN interfaces.
- With Release 12.1(14)E and later releases, you can use the **interface range** command to create VLAN interfaces.

- With releases earlier than Release 12.1(26)E, for VLAN interfaces, the **interface range** command supports only those VLAN interfaces for which Layer 2 VLANs have been created with the **interface vlan** command (the **show running-configuration** command displays the configured VLAN interfaces). The **interface range** command does not support VLAN interfaces that are not displayed by the **show running-configuration** command.
- With Release 12.1(26)E and later releases, the **interface range** command supports VLAN interfaces for which Layer 2 VLANs have not been created with the **interface vlan** command.
- With releases earlier than Release 12.1(14)E, you cannot use the **interface range** command to create VLAN interfaces: the **interface range** command is supported only to modify VLAN interfaces that have been created with the **interface vlan** command. You can enter the **show running-configuration** command to display the configured VLAN interfaces.

**Note**

The link state messages (LINK-3-UPDOWN and LINEPROTO-5-UPDOWN) are disabled by default. Enter the **logging event link status** command on each interface where you want the messages enabled.

This example shows how to reenable all Fast Ethernet ports 5/1 to 5/5:

```
Router(config)# interface range fastethernet 5/1 - 5
Router(config-if)# no shutdown
Router(config-if)#
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
5, changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
3, changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
4, changed state to up
Router(config-if)#
```

This example shows how to use a comma to add different interface type strings to the range to reenable all Fast Ethernet ports in the range 5/1 to 5/5 and both Gigabit Ethernet ports (1/1 and 1/2):

```
Router(config-if)# interface range fastethernet 5/1 - 5, gigabitethernet 1/1 - 2
Router(config-if)# no shutdown
Router(config-if)#
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/1, changed state to
up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/2, changed state to
up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
5, changed state to up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
3, changed state to up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
4, changed state to up
Router(config-if)#
```

If you enter multiple configuration commands while you are in interface-range configuration mode, each command is executed as it is entered (they are not batched together and executed after you exit interface-range configuration mode).

If you exit interface-range configuration mode while the commands are being executed, some commands may not be executed on all interfaces in the range. Wait until the command prompt reappears before exiting interface-range configuration mode.

Defining and Using Interface-Range Macros

You can define an interface-range macro to automatically select a range of interfaces for configuration. Before you can use the **macro** keyword in the **interface range macro** command string, you must define the macro.

To define an interface-range macro, perform this task:

Command	Purpose
Router(config)# define interface-range <i>macro_name</i> { vlan <i>vlan_ID</i> - <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> - <i>port</i> } [, { <i>type</i> ¹ <i>slot/port</i> - <i>port</i> }]	Defines the interface-range macro and save it in NVRAM.
Router(config)# no define interface-range <i>macro_name</i>	Deletes a macro.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to define an interface-range macro named `enet_list` to select Fast Ethernet ports 5/1 through 5/4:

```
Router(config)# define interface-range enet_list fastethernet 5/1 - 4
```

To show the defined interface-range macro configuration, perform this task:

Command	Purpose
Router# show running-config	Shows the defined interface-range macro configuration.

This example shows how to display the defined interface-range macro named `enet_list`:

```
Router# show running-config | include define
define interface-range enet_list FastEthernet5/1 - 4
Router#
```

To use an interface-range macro in the **interface range** command, perform this task:

Command	Purpose
Router(config)# interface range macro <i>macro_name</i>	Selects the interface range to be configured using the values saved in a named interface-range macro.

This example shows how to change to the interface-range configuration mode using the interface-range macro `enet_list`:

```
Router(config)# interface range macro enet_list
Router(config-if)#
```

Configuring Optional Interface Features

These sections describe optional interface features:

- [Configuring Ethernet Interface Speed and Duplex Mode](#), page 6-7
- [Configuring Jumbo Frame Support](#), page 6-10
- [Configuring IEEE 802.3Z Flow Control](#), page 6-14
- [Configuring the Port Debounce Timer](#), page 6-15
- [Adding a Description for an Interface](#), page 6-16

Configuring Ethernet Interface Speed and Duplex Mode

These sections describe how to configure Ethernet port speed and duplex mode:

- [Speed and Duplex Mode Configuration Guidelines](#), page 6-7
- [Setting the Ethernet Interface Speed](#), page 6-8
- [Setting the Interface Duplex Mode](#), page 6-8
- [Configuring Link Negotiation on Gigabit Ethernet Ports](#), page 6-8
- [Displaying the Speed and Duplex Mode Configuration](#), page 6-9

Speed and Duplex Mode Configuration Guidelines

You usually configure Ethernet port speed and duplex mode parameters to auto and allow the Catalyst 6500 series switch to negotiate the speed and duplex mode between ports. If you decide to configure the port speed and duplex modes manually, consider the following information:

- If you set the Ethernet port speed to auto, the switch automatically sets the duplex mode to auto.
- If you enter the **no speed** command, the switch automatically configures both speed and duplex to auto.
- If you configure an Ethernet port speed to a value other than auto (for example, 10, 100, or 1000 Mbps), configure the connecting port to match. Do not configure the connecting port to negotiate the speed.
- If you manually configure the Ethernet port speed to either 10 or 100 Mbps, the switch prompts you to also configure the duplex mode on the port.

**Note**

Catalyst 6500 series switches cannot automatically negotiate Ethernet port speed and duplex mode if the connecting port is configured to a value other than auto.

**Caution**

Changing the Ethernet port speed and duplex mode configuration might shut down and reenable the interface during the reconfiguration.

Setting the Ethernet Interface Speed



Note

If you set the Ethernet port speed to **auto** on a 10/100-Mbps or 10/100/1000-Mbps Ethernet port, both speed and duplex are autonegotiated.

To set the port speed for a 10/100 or a 10/100/1000-Mbps Ethernet port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface fastethernet <i>slot/port</i>	Selects the Ethernet port to be configured.
Step 2	Router(config-if)# speed {10 100 1000 auto}	Sets the speed of the Ethernet interface.
	Router(config-if)# no speed	Reverts to the default configuration (speed auto).

This example shows how to set the speed to 100 Mbps on the Fast Ethernet port 5/4:

```
Router(config)# interface fastethernet 5/4
Router(config-if)# speed 100
```

Setting the Interface Duplex Mode



Note

- 10-Gigabit Ethernet and Gigabit Ethernet are full duplex only. You cannot change the duplex mode on 10-Gigabit Ethernet or Gigabit Ethernet ports or on a 10/100/1000-Mbps port configured for Gigabit Ethernet.
- If you set the port speed to auto on a 10/100-Mbps or a 10/100/1000-Mbps Ethernet port, both speed and duplex are autonegotiated. You cannot change the duplex mode of autonegotiation ports.

To set the duplex mode of an Ethernet or Fast Ethernet port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface fastethernet <i>slot/port</i>	Selects the Ethernet port to be configured.
Step 2	Router(config-if)# duplex [auto full half]	Sets the duplex mode of the Ethernet port.
	Router(config-if)# no duplex	Reverts to the default configuration (duplex auto).

This example shows how to set the duplex mode to full on Fast Ethernet port 5/4:

```
Router(config)# interface fastethernet 5/4
Router(config-if)# duplex full
```

Configuring Link Negotiation on Gigabit Ethernet Ports



Note

Link negotiation does not negotiate port speed.

On Gigabit Ethernet ports, link negotiation exchanges flow-control parameters, remote fault information, and duplex information. Link negotiation is enabled by default.

The ports on both ends of a link must have the same setting. The link will not come up if the ports at each end of the link are set inconsistently (link negotiation enabled on one port and disabled on the other port).

Table 6-1 shows the four possible link negotiation configurations and the resulting link status for each configuration.

Table 6-1 Link Negotiation Configuration and Possible Link Status

Link Negotiation State		Link Status	
Local Port	Remote Port	Local Port	Remote Port
Off	Off	Up	Up
On	On	Up	Up
Off	On	Up	Down
On	Off	Down	Up

To configure link negotiation on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface gigabitethernet slot/port	Selects the port to be configured.
Step 2	Router(config-if)# speed nonegotiate Router(config-if)# no speed nonegotiate	Disables link negotiation. Reverts to the default configuration (link negotiation enabled).

This example shows how to enable link negotiation on Gigabit Ethernet port 5/4:

```
Router(config)# interface gigabitethernet 5/4
Router(config-if)# no speed nonegotiate
```

Displaying the Speed and Duplex Mode Configuration

To display the speed and duplex mode configuration for a port, perform this task:

Command	Purpose
Router# show interfaces type¹ slot/port	Displays the speed and duplex mode configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display the speed and duplex mode of Fast Ethernet port 5/4:

```
Router# show interfaces fastethernet 5/4
FastEthernet5/4 is up, line protocol is up
  Hardware is Cat6K 100Mb Ethernet, address is 0050.f0ac.3058 (bia 0050.f0ac.3058)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:33, output never, output hang never
```

```

Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  1238 packets input, 273598 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input packets with dribble condition detected
 1380 packets output, 514382 bytes, 0 underruns
  0 output errors, 0 collisions, 2 interface resets
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out
Router#

```

Configuring Jumbo Frame Support

These sections describe jumbo frame support:

- [Understanding Jumbo Frame Support, page 6-10](#)
- [Configuring MTU Sizes, page 6-13](#)



Caution

The following switching modules support a maximum ingress frame size of 8092 bytes:

- WS-X6516-GE-TX when operating at 100 Mbps
- WS-X6148-RJ-45 and WS-X6148-RJ-45V
- WS-X6148-RJ21 and WS-X6148-RJ21V
- WS-X6248-RJ-45
- WS-X6248-TEL
- WS-X6248A-RJ-45
- WS-X6248A-TEL
- WS-X6348-RJ-45 and WS-X6348-RJ45V
- WS-X6348-RJ-21 and WX-X6348-RJ21V

When jumbo frame support is configured, these modules drop ingress frames larger than 8092 bytes.



Note

The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX do not support jumbo frames.

Understanding Jumbo Frame Support

These sections describe jumbo frame support:

- [Jumbo Frame Support Overview, page 6-11](#)
- [Ethernet Ports, page 6-12](#)
- [VLAN Interfaces, page 6-13](#)

Jumbo Frame Support Overview

A jumbo frame is a frame larger than the default Ethernet frame size. You enable jumbo frame support by configuring a larger-than-default maximum transmission unit (MTU) size on a port or VLAN interface and, with Release 12.1(13)E and later releases, configuring the global LAN port MTU size.

**Note**

- Jumbo frame support fragments routed traffic in software on the MSFC.
- Jumbo frame support does not fragment bridged traffic.

These sections provide an overview of jumbo frame support:

- [Bridged and Routed Traffic Size Check at Ingress 10, 10/100, and 100 Mbps Ethernet and 10 Gigabit Ethernet Ports, page 6-11](#)
- [Bridged and Routed Traffic Size Check at Ingress Gigabit Ethernet Ports, page 6-11](#)
- [Routed Traffic Size Check on the PFC, page 6-11](#)
- [Bridged and Routed Traffic Size Check at Egress 10, 10/100, and 100 Mbps Ethernet Ports, page 6-11](#)
- [Bridged and Routed Traffic Size Check at Egress Gigabit Ethernet and 10 Gigabit Ethernet Ports, page 6-12](#)

Bridged and Routed Traffic Size Check at Ingress 10, 10/100, and 100 Mbps Ethernet and 10 Gigabit Ethernet Ports

Jumbo frame support compares ingress traffic size with the global LAN port MTU size at ingress 10, 10/100, and 100 Mbps Ethernet and 10 Gigabit Ethernet LAN ports that have a nondefault MTU size configured. The port drops traffic that is oversized.

With Release 12.1(13)E and later releases, you can configure the global LAN port MTU size (see the [“Configuring the Global LAN Port MTU Size” section on page 6-14](#)). With earlier releases, the global LAN port MTU size is 9216 bytes and 9216 bytes is the only configurable nondefault MTU size for Layer 2 LAN ports.

Bridged and Routed Traffic Size Check at Ingress Gigabit Ethernet Ports

Gigabit Ethernet LAN ports configured with a nondefault MTU size accept frames containing packets of any size larger than 64 bytes. With a nondefault MTU size configured, Gigabit Ethernet LAN ports do not check for oversize ingress frames.

Routed Traffic Size Check on the PFC

For traffic that needs to be routed Jumbo frame support on the PFC compares traffic sizes to the configured MTU sizes and provides Layer 3 switching for jumbo traffic between interfaces configured with MTU sizes large enough to accommodate the traffic. Between interfaces that are not configured with large enough MTU sizes, if the “do not fragment bit” is not set, the PFC sends the traffic to the MSFC to be fragmented and routed in software. If the “do not fragment bit” is set, the PFC drops the traffic.

Bridged and Routed Traffic Size Check at Egress 10, 10/100, and 100 Mbps Ethernet Ports

10, 10/100, and 100 Mbps Ethernet LAN ports configured with a nondefault MTU size transmit frames containing packets of any size larger than 64 bytes. With a nondefault MTU size configured, 10, 10/100, and 100 Mbps Ethernet LAN ports do not check for oversize egress frames.

Bridged and Routed Traffic Size Check at Egress Gigabit Ethernet and 10 Gigabit Ethernet Ports

Jumbo frame support compares egress traffic size with the global egress LAN port MTU size at egress Gigabit Ethernet and 10 Gigabit Ethernet LAN ports that have a nondefault MTU size configured. The port drops traffic that is oversized.

With Release 12.1(13)E and later releases, you can configure the global LAN port MTU size (see the [“Configuring the Global LAN Port MTU Size” section on page 6-14](#)). With earlier releases, the LAN port MTU size is 9216 bytes and 9216 bytes is the only configurable nondefault MTU size for Layer 2 LAN ports.

Ethernet Ports

These sections describe configuring nondefault MTU sizes on Ethernet ports:

- [Ethernet Port Overview, page 6-12](#)
- [Layer 3 Ethernet Ports, page 6-12](#)
- [Layer 2 Ethernet Ports, page 6-12](#)

Ethernet Port Overview

Configuring a nondefault MTU size on a 10, 10/100, or 100 Mbps Ethernet port limits ingress packets to the global LAN port MTU size and permits egress traffic of any size larger than 64 bytes.

Configuring a nondefault MTU size on a Gigabit Ethernet port permits ingress packets of any size larger than 64 bytes and limits egress traffic to the global LAN port MTU size.

Configuring a nondefault MTU size on a 10 Gigabit Ethernet port limits ingress and egress packets to the global LAN port MTU size.

Configuring a nondefault MTU size on an Ethernet port limits routed traffic to the configured MTU size.

With Release 12.1(11b)E and later releases, you can configure the MTU size on any Ethernet port.

With earlier releases, you can configure the MTU size only on Gigabit Ethernet and 10-Gigabit Ethernet ports.

Layer 3 Ethernet Ports

On a Layer 3 port, you can configure an MTU size that is different than the global LAN port MTU size.

With Release 12.1(13)E and later releases, you can configure a different MTU size on each Layer 3 Ethernet port.

With earlier releases, you can configure only a single larger-than-default MTU size on the switch for Layer 3 Ethernet ports. When you configure a Layer 3 Ethernet port with a nondefault MTU size, the switch automatically configures all other Layer 3 Ethernet ports and Layer 3 EtherChannels with nondefault MTU sizes to the newly configured size. Layer 3 Ethernet ports and EtherChannels that have not been changed from the default are not affected.



Note

Traffic through a Layer 3 Ethernet LAN port that is configured with a nondefault MTU size is also subject to the global LAN port MTU size (see the [“Configuring the Global LAN Port MTU Size” section on page 6-14](#)).

Layer 2 Ethernet Ports

On a Layer 2 port, you can only configure an MTU size that matches the global LAN port MTU size (see the [“Configuring the Global LAN Port MTU Size” section on page 6-14](#)).

Release 12.1(13)E and later releases support nondefault MTU sizes between 1,500 and 9,216 bytes for Layer 2 Ethernet ports, configured per-port with the **mtu** command and globally with the **system jumbomtu** command.

With earlier releases, 9216 bytes is the only supported nondefault MTU size for Layer 2 Ethernet ports, configured per-port with the **mtu** command.

VLAN Interfaces

You can configure a different MTU size on each Layer 3 VLAN interface. Configuring a nondefault MTU size on a VLAN interface limits traffic to the nondefault MTU size.

You can configure the MTU size on VLAN interfaces to support jumbo frames with the following hardware and software:

- Supervisor Engine 1 and Release 12.1(7)E or later
- Supervisor Engine 2 and Release 12.1(8a)E or later

Configuring MTU Sizes

These sections describe how to configure MTU sizes:

- [Configuring MTU Sizes, page 6-13](#)
- [Configuring the Global LAN Port MTU Size, page 6-14](#)

Configuring the MTU Size

To configure the MTU size, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {{type ¹ slot/port} {port-channel port_channel_number} slot/port}}	Selects the interface to configure.
Step 2	Router(config-if)# mtu mtu_size Router(config-if)# no mtu	Configures the MTU size. Reverts to the default MTU size (1500 bytes).
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show running-config interface [{gigabitethernet tengigabitethernet} slot/port]	Displays the running configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

When configuring the MTU size, note the following syntax information:

- For VLAN interfaces and Layer 3 Ethernet ports, supported MTU values are from 64 to 9216 bytes.
- For Layer 2 Ethernet ports with Release 12.1(13)E and later releases, you can configure only the global egress LAN port MTU size (see the “[Configuring the Global LAN Port MTU Size](#)” section on page 6-14).
- For Layer 2 Ethernet ports with earlier releases, the only supported MTU size is 9216 bytes.

This example shows how to configure the MTU size on Gigabit Ethernet port 1/2:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/2
Router(config-if)# mtu 9216
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show interface gigabitethernet 1/2
GigabitEthernet1/2 is administratively down, line protocol is down
  Hardware is C6k 1000Mb 802.3, address is 0030.9629.9f88 (bia 0030.9629.9f88)
  MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
  <...Output Truncated...>
Router#
```

Configuring the Global LAN Port MTU Size

To configure the global LAN port MTU size, perform this task:

	Command	Purpose
Step 1	Router(config)# system jumbomtu <i>mtu_size</i>	Configures the global LAN port MTU size.
	Router(config)# no system jumbomtu	Reverts to the default global LAN port MTU size (9216 bytes).
Step 2	Router(config)# end	Exits configuration mode.

Configuring IEEE 802.3Z Flow Control

Gigabit Ethernet and 10-Gigabit Ethernet ports on the Catalyst 6500 series switches use flow control to stop the transmission of frames to the port for a specified time; other Ethernet ports use flow control to respond to flow-control requests.

If a Gigabit Ethernet or 10-Gigabit Ethernet port receive buffer becomes full, the port transmits an IEEE 802.3Z pause frame that requests remote ports to delay sending frames for a specified time. All Ethernet ports (10 Gbps, 1 Gbps, 100 Mbps, and 10 Mbps) can receive and respond to IEEE 802.3Z pause frames from other devices.

To configure flow control on an Ethernet port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# flowcontrol { receive send } { desired off on }	Configures a port to send or respond to pause frames.
	Router(config-if)# no flowcontrol { receive send }	Reverts to the default flow control settings.
Step 3	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] flowcontrol	Displays the flow-control configuration for all ports.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When configuring flow control, note the following syntax information:

- 10-Gigabit Ethernet ports are permanently configured to respond to pause frames.
- When the configuration of the remote ports is unknown, use the **receive desired** keywords to configure a Gigabit Ethernet port to respond to received pause frames.
- Use the **receive on** keywords to configure a Gigabit Ethernet port to respond to received pause frames.
- Use the **receive off** keywords to configure a Gigabit Ethernet port to ignore received pause frames.
- When configuring transmission of pause frames, note the following information:
 - When the configuration of the remote ports is unknown, use the **send desired** keywords to configure a port to send pause frames.
 - Use the **send on** keywords to configure a port to send pause frames.
 - Use the **send off** keywords to configure a port not to send pause frames.

This example shows how to turn on receive flow control and how to verify the flow-control configuration:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/2
Router(config-if)# flowcontrol receive on
Router(config-if)# end
Router# show interfaces flowcontrol

Interface Send      Receive
Gi1/1      Desired          OFF
Gi1/2      Desired          ON
Fa5/1      Not capable     OFF
<output truncated>
```

Configuring the Port Debounce Timer

The port debounce timer delays notification of a link change, which can decrease traffic loss due to network reconfiguration. Release 12.1(13)E and later releases support the port debounce timer on all LAN ports. You can configure the port debounce timer separately on each LAN port.



Caution

Enabling the port debounce timer causes link up and link down detections to be delayed, resulting in loss of traffic during the debouncing period. This situation might affect the convergence and reconvergence of some Layer 2 and Layer 3 protocols.

Table 6-2 lists the time delay that occurs before notification of a link change.

Table 6-2 Port Debounce Timer Delay Time

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
10BASE-FL ports	300 milliseconds	3100 milliseconds
10/100BASE-TX ports	300 milliseconds	3100 milliseconds
100BASE-FX ports	300 milliseconds	3100 milliseconds
10/100/1000BASE-TX ports	300 milliseconds	3100 milliseconds
1000BASE-TX ports	300 milliseconds	3100 milliseconds

Table 6-2 Port Debounce Timer Delay Time (continued)

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
Fiber Gigabit ports	10 milliseconds	100 through 5000 milliseconds
10-Gigabit ports	Note With Release 12.1(13)E and later releases, you can configure the port debounce timer on 10 Gigabit Ethernet ports, but it has no effect. With Release 12.1(19)E and later releases, you cannot configure the port debounce timer on 10 Gigabit Ethernet ports.	

To configure the debounce timer on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# link debounce [time <i>debounce_time</i>] Router(config-if)# no link debounce	Configures the debounce timer. Note The time keyword is supported only on fiber Gigabit Ethernet ports. Reverts to the default setting.
Step 3	Router# show interfaces debounce	Verifies the configuration.

1. *type* = ethernet, fastethernet, or gigabitethernet

On fiber Gigabit Ethernet ports, you can increase the port debounce timer value in increments of 100 milliseconds up to 5000 milliseconds.

This example shows how to enable the port debounce timer on Fast Ethernet port 5/12:

```
Router(config)# interface fastethernet 5/12
Router(config-if)# link debounce
Router(config-if)# end
```

This example shows how to display the port debounce timer settings:

```
Router# show interfaces debounce | include enable
Fa5/12 enable          3100
```

Adding a Description for an Interface

You can add a description about an interface to help you remember its function. The description appears in the output of the following commands: **show configuration**, **show running-config**, and **show interfaces**.

To add a description for an interface, perform this task:

Command	Purpose
Router(config-if)# description <i>string</i>	Adds a description for an interface.
Router(config-if)# no description	Deletes a description from an interface.

This example shows how to add a description on Fast Ethernet port 5/5:

```
Router(config)# interface fastethernet 5/5
Router(config-if)# description Channel-group to "Marketing"
```

Understanding Online Insertion and Removal

The online insertion and removal (OIR) feature supported on the Catalyst 6500 series switches allows you to remove and replace modules while the system is online. You can shut down the modules before removal and restart it after insertion without causing other software or interfaces to shut down.



Note

Do not remove or install more than one module at a time. After you remove or install a module, check the LEDs before continuing. For module LED descriptions, refer to the *Catalyst 6500 Series Switch Installation Guide*.

When a module has been removed or installed, the Catalyst 6500 series switch stops processing traffic for the module and scans the system for a configuration change. Each interface type is verified against the system configuration, and then the system runs diagnostics on the new module. There is no disruption to normal operation during module insertion or removal.

The switch can bring only an identical replacement module online. If the replacement module is different from the removed module, you must configure it before the switch can bring it online.

Layer 2 MAC addresses are stored in an EEPROM, which allows modules to be replaced online without requiring the system to update switching tables and data structures. Regardless of the types of modules installed, the Layer 2 MAC addresses do not change unless you replace the supervisor engine. If you do replace the supervisor engine, the Layer 2 MAC addresses of *all* ports change to those specified in the address allocator on the new supervisor engine.

Monitoring and Maintaining Interfaces

You can perform the tasks in the following sections to monitor and maintain interfaces:

- [Monitoring Interface Status, page 6-17](#)
- [Clearing Counters on an Interface, page 6-18](#)
- [Resetting an Interface, page 6-19](#)
- [Shutting Down and Restarting an Interface, page 6-19](#)

Monitoring Interface Status

The software contains commands that you can enter at the EXEC prompt to display information about the interface including the version of the software and the hardware and statistics about interfaces. The following table lists some of the interface monitoring commands. (You can display the complete list of **show** commands by using the **show ?** command at the EXEC prompt.) These commands are described in the *Cisco IOS Interface Command Reference* publication.

To display information about the interface, perform these tasks:

Command	Purpose
Router# show ibc	Displays current internal status information.
Router# show eobc	Displays current internal out-of-band information.
Router# show interfaces [<i>type slot/port</i>]	Displays the status and configuration of all or a specific interface.
Router# show running-config	Displays the currently running configuration.
Router# show rif	Displays the current contents of the routing information field (RIF) cache.
Router# show protocols [<i>type slot/port</i>]	Displays the global (system-wide) and interface-specific status of any configured protocol.
Router# show version	Displays the hardware configuration, software version, the names and sources of configuration files, and the boot images.

This example shows how to display the status of Fast Ethernet port 5/5:

```
Router# show protocols fastethernet 5/5
FastEthernet5/5 is up, line protocol is up
Router#
```

Clearing Counters on an Interface

To clear the interface counters shown with the **show interfaces** command, perform this task:

Command	Purpose
Router# clear counters {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>channel_ID</i> }}	Clears interface counters.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to clear and reset the counters on Fast Ethernet port 5/5:

```
Router# clear counters fastethernet 5/5
Clear "show interface" counters on this interface [confirm] y
Router#
*Sep 30 08:42:55: %CLEAR-5-COUNTERS: Clear counter on interface FastEthernet5/5
```

The **clear counters** command clears all the current counters from the interface unless the optional arguments specify a specific interface.



Note

The **clear counters** command clears counters displayed with the EXEC **show interfaces** command, not counters retrieved using SNMP.

Resetting an Interface

To reset an interface, perform this task:

Command	Purpose
Router# clear interface <i>type</i> ¹ <i>slot/port</i>	Resets an interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to reset Fast Ethernet port 5/5:

```
Router# clear interface fastethernet 5/5
Router#
```

Shutting Down and Restarting an Interface

You can shut down an interface, which disables all functions on the specified interface and shows the interface as unavailable on all monitoring command displays. This information is communicated to other network servers through all dynamic routing protocols. The interface is not included in any routing updates.

To shut down an interface and then restart it, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan <i>vlan_ID</i> <i>type</i> ¹ <i>slot/port</i> {port-channel <i>channel_ID</i> }}	Selects the interface to be configured.
Step 2	Router(config-if)# shutdown	Shuts down the interface.
Step 3	Router(config-if)# no shutdown	Reenables the interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to shut down Fast Ethernet port 5/5:

```
Router(config)# interface fastethernet 5/5
Router(config-if)# shutdown
Router(config-if)#
*Sep 30 08:33:47: %LINK-5-CHANGED: Interface FastEthernet5/5, changed state to
administratively down
```

This example shows how to reenabable Fast Ethernet port 5/5:

```
Router(config-if)# no shutdown
Router(config-if)#
*Sep 30 08:36:00: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
```

To check if an interface is disabled, enter the EXEC **show interfaces** command. An interface that has been shut down is shown as administratively down in the **show interfaces** command display.



Configuring LAN Ports for Layer 2 Switching

This chapter describes how to use the command-line interface (CLI) to configure Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet LAN ports for Layer 2 switching on the Catalyst 6500 series switches. The configuration tasks in this chapter apply to LAN ports on LAN switching modules and to the LAN ports on the supervisor engine.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How Layer 2 Switching Works, page 7-1](#)
- [Default Layer 2 LAN Interface Configuration, page 7-5](#)
- [Layer 2 LAN Interface Configuration Guidelines and Restrictions, page 7-6](#)
- [Configuring LAN Interfaces for Layer 2 Switching, page 7-7](#)



Note

To configure Layer 3 interfaces, see [Chapter 12, “Configuring Layer 3 Interfaces.”](#)

Understanding How Layer 2 Switching Works

These sections describe how Layer 2 switching works on the Catalyst 6500 series switches:

- [Understanding Layer 2 Ethernet Switching, page 7-1](#)
- [Understanding VLAN Trunks, page 7-2](#)
- [Layer 2 LAN Port Modes, page 7-4](#)

Understanding Layer 2 Ethernet Switching

These sections describe Layer 2 Ethernet switching:

- [Layer 2 Ethernet Switching Overview, page 7-2](#)
- [Switching Frames Between Segments, page 7-2](#)
- [Building the Address Table, page 7-2](#)

Layer 2 Ethernet Switching Overview

Catalyst 6500 series switches support simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

Catalyst 6500 series switches solve congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own 10-, 100-, or 1000-Mbps collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a properly configured switched environment achieve full access to the bandwidth.

Because collisions are a major bottleneck in Ethernet networks, an effective solution is full-duplex communication. Normally, Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, effective Ethernet bandwidth doubles to 20 Mbps for 10-Mbps ports and to 200 Mbps for Fast Ethernet ports. Gigabit Ethernet ports on Catalyst 6500 series switches are full duplex only, providing 2-Gbps effective bandwidth.

Switching Frames Between Segments

Each LAN port on a Catalyst 6500 series switch can connect to a single workstation or server, or to a hub through which workstations or servers connect to the network.

On a typical Ethernet hub, all ports connect to a common backplane within the hub, and the bandwidth of the network is shared by all devices attached to the hub. If two stations establish a session that uses a significant level of bandwidth, the network performance of all other stations attached to the hub is degraded.

To reduce degradation, the switch considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the switch forwards frames from one LAN port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between LAN ports efficiently, the switch maintains an address table. When a frame enters the switch, it associates the MAC address of the sending station with the LAN port on which it was received.

Building the Address Table

Catalyst 6500 series switches build the address table by using the source address of the frames received. When the switch receives a frame for a destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the switch adds its relevant source address and port ID to the address table. The switch then forwards subsequent frames to a single LAN port without flooding to all LAN ports.

The address table can store at least 16,000 address entries without flooding any entries. The switch uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Understanding VLAN Trunks

These sections describe VLAN trunks on the Catalyst 6500 series switches:

- [Trunking Overview, page 7-3](#)
- [Encapsulation Types, page 7-4](#)

Trunking Overview

**Note**

For information about VLANs, see [Chapter 9, “Configuring VLANs.”](#)

A trunk is a point-to-point link between the switch and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

Two trunking encapsulations are available on all Ethernet ports:

- Inter-Switch Link (ISL)—ISL is a Cisco-proprietary trunking encapsulation.

**Note**

The following switching modules do not support ISL encapsulation:

- WS-X6501-10GEX4
- WS-X6502-10GE
- WS-X6548-GE-TX
- WS-X6548V-GE-TX
- WS-X6148-GE-TX
- WS-X6148V-GE-TX

- 802.1Q—802.1Q is an industry-standard trunking encapsulation.

You can configure a trunk on a single Ethernet port or on an EtherChannel. For more information about EtherChannel, see [Chapter 13, “Configuring EtherChannels.”](#)

Ethernet trunk ports support several trunking modes (see [Table 7-2 on page 7-4](#)). You can specify whether the trunk uses ISL or 802.1Q encapsulation, and if the encapsulation type is autonegotiated.

**Note**

You can configure LAN ports to negotiate the encapsulation type. You cannot configure WAN interfaces to negotiate the encapsulation type.

The Dynamic Trunking Protocol (DTP) manages trunk autonegotiation on LAN ports. DTP supports autonegotiation of both ISL and 802.1Q trunks.

To autonegotiate trunking, the LAN ports must be in the same VTP domain. Use the **trunk** or **nonegotiate** keywords to force LAN ports in different domains to trunk. For more information on VTP domains, see [Chapter 8, “Configuring VTP.”](#)

Encapsulation Types

Table 7-1 lists the Ethernet trunk encapsulation types.

Table 7-1 Ethernet Trunk Encapsulation Types

Encapsulation	Function
<code>switchport trunk encapsulation isl</code>	Specifies ISL encapsulation on the trunk link. Note 10-Gigabit Ethernet ports do not support ISL encapsulation.
<code>switchport trunk encapsulation dot1q</code>	Specifies 802.1Q encapsulation on the trunk link.
<code>switchport trunk encapsulation negotiate</code>	Specifies that the LAN port negotiate with the neighboring LAN port to become an ISL (preferred) or 802.1Q trunk, depending on the configuration and capabilities of the neighboring LAN port.

The trunking mode, the trunk encapsulation type, and the hardware capabilities of the two connected LAN ports determine whether a link becomes an ISL or 802.1Q trunk.

Layer 2 LAN Port Modes

Table 7-2 lists the Layer 2 LAN port modes and describes how they function on LAN ports.

Table 7-2 Layer 2 LAN Port Modes

Mode	Function
<code>switchport mode access</code>	Puts the LAN port into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. The LAN port becomes a nontrunk port even if the neighboring LAN port does not agree to the change.
<code>switchport mode dynamic desirable</code>	Makes the LAN port actively attempt to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to trunk , desirable , or auto mode. This is the default mode for all LAN ports.
<code>switchport mode dynamic auto</code>	Makes the LAN port willing to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to trunk or desirable mode.
<code>switchport mode trunk</code>	Puts the LAN port into permanent trunking mode and negotiates to convert the link into a trunk link. The LAN port becomes a trunk port even if the neighboring port does not agree to the change.
<code>switchport nonegotiate</code>	Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames. You must configure the neighboring port manually as a trunk port to establish a trunk link.

**Note**

DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that LAN ports connected to devices that do not support DTP are configured with the **access** keyword if you do not intend to trunk across those links. To enable trunking to a device that does not support DTP, use the **nonegotiate** keyword to cause the LAN port to become a trunk but not generate DTP frames.

Default Layer 2 LAN Interface Configuration

Table 7-3 shows the Layer 2 LAN port default configuration.

Table 7-3 Layer 2 LAN Interface Default Configuration

Feature	Default
Interface mode: <ul style="list-style-type: none"> • Before entering the switchport command • After entering the switchport command 	Layer 3 (unconfigured) switchport mode dynamic desirable
Trunk encapsulation	switchport trunk encapsulation negotiate
Allowed VLAN range	<ul style="list-style-type: none"> • With Release 12.1(13)E and later releases, VLANs 1 to 4094, except reserved VLANs (see Table 9-1 on page 9-2) • With 12.1 E releases earlier than Release 12.1(13)E, VLANs 1 to 1005
VLAN range eligible for pruning	VLANs 2 to 1001
Default VLAN (for access ports)	VLAN 1
Native VLAN (for 802.1Q trunks)	VLAN 1
Spanning Tree Protocol (STP)	Enabled for all VLANs
STP port priority	128
STP port cost	<ul style="list-style-type: none"> • 100 for 10-Mbps Ethernet LAN ports • 19 for 10/100-Mbps Fast Ethernet LAN ports • 19 for 100-Mbps Fast Ethernet LAN ports • 4 for 1,000-Mbps Gigabit Ethernet LAN ports • 2 for 10,000-Mbps 10-Gigabit Ethernet LAN ports

Layer 2 LAN Interface Configuration Guidelines and Restrictions

When configuring Layer 2 LAN ports, follow these guidelines and restrictions:

Restrictions

- 10-Gigabit Ethernet ports do not support ISL encapsulation.
- Non-Cisco 802.1Q switches maintain only a single instance of spanning tree (the Mono Spanning Tree, or MST) that defines the spanning tree topology for all VLANs. When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the MST of the non-Cisco switch and the native VLAN spanning tree of the Cisco switch combine to form a single spanning tree topology known as the Common Spanning Tree (CST).
- Because Cisco switches transmit BPDUs to the SSTP multicast MAC address on VLANs other than the native VLAN of the trunk, non-Cisco switches do not recognize these frames as BPDUs and flood them on all ports in the corresponding VLAN. Other Cisco switches connected to the non-Cisco 802.1q cloud receive these flooded BPDUs. This allows Cisco switches to maintain a per-VLAN spanning tree topology across a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud separating the Cisco switches is treated as a single broadcast segment between all switches connected to the non-Cisco 802.1q cloud through 802.1q trunks.

Guidelines

- When connecting Cisco switches through an 802.1q trunk, make sure the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
- Disabling spanning tree on the native VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can cause spanning tree loops. We recommend that you leave spanning tree enabled on the native VLAN of an 802.1Q trunk. If this is not possible, disable spanning tree on every VLAN in the network. Make sure your network is free of physical loops before disabling spanning tree.
- When you connect two Cisco switches through 802.1Q trunks, the switches exchange spanning tree BPDUs on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1d spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).
- Make certain that the native VLAN is the same on all of the 802.1Q trunks connecting the Cisco switches to the non-Cisco 802.1Q cloud.
- If you are connecting multiple Cisco switches to a non-Cisco 802.1Q cloud, all of the connections must be through 802.1Q trunks. You cannot connect Cisco switches to a non-Cisco 802.1Q cloud through ISL trunks or through access ports. Doing so will cause the switch to place the ISL trunk port or access port into the spanning tree “port inconsistent” state and no traffic will pass through the port.

Configuring LAN Interfaces for Layer 2 Switching

These sections describe how to configure Layer 2 switching on the Catalyst 6500 series switches:

- [Configuring a LAN Port for Layer 2 Switching, page 7-7](#)
- [Configuring a Layer 2 Switching Port as a Trunk, page 7-8](#)
- [Configuring a LAN Interface as a Layer 2 Access Port, page 7-14](#)



Note

- Use the **default interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* command to revert an interface to its default configuration.
- With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Configuring a LAN Port for Layer 2 Switching

To configure a LAN port for Layer 2 switching, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# shutdown	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching. Note You must enter the switchport command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional switchport commands with keywords.
	Router(config-if)# no switchport	Clears Layer 2 LAN port configuration.
Step 4	Router(config-if)# no shutdown	Activates the interface. (Required only if you shut down the interface.)
Step 5	Router(config-if)# end	Exits configuration mode.
Step 6	Router# show running-config interface [<i>type</i> ¹ <i>slot/port</i>]	Displays the running configuration of the interface.
Step 7	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] switchport	Displays the switch port configuration of the interface.
Step 8	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] trunk	Displays the trunk configuration of the interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

After you enter the **switchport** command, the default mode is **switchport mode dynamic desirable**. If the neighboring port supports trunking and is configured to allow trunking, the link becomes a Layer 2 trunk when you enter the **switchport** command. By default, LAN trunk ports negotiate encapsulation. If the neighboring port supports ISL and 802.1Q encapsulation and both ports are set to negotiate the encapsulation type, the trunk uses ISL encapsulation (10-Gigabit Ethernet ports do not support ISL encapsulation).

Configuring a Layer 2 Switching Port as a Trunk

These sections describe configuring a Layer 2 switching port as a trunk:

- [Preparing a Layer 2 Switching Port for Configuration as a Trunk, page 7-8](#)
- [Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk, page 7-8](#)
- [Configuring the Layer 2 Trunk to Use DTP, page 7-9](#)
- [Configuring the Layer 2 Trunk Not to Use DTP, page 7-9](#)
- [Configuring the Default VLAN, page 7-10](#)
- [Configuring the 802.1Q Native VLAN, page 7-11](#)
- [Configuring the List of VLANs Allowed on a Trunk, page 7-11](#)
- [Configuring the List of Prune-Eligible VLANs, page 7-12](#)
- [Completing Trunk Configuration, page 7-13](#)
- [Verifying Layer 2 Trunk Configuration, page 7-13](#)
- [Configuration and Verification Examples, page 7-13](#)

Preparing a Layer 2 Switching Port for Configuration as a Trunk

To prepare a Layer 2 switching port for configuration as a trunk, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# shutdown	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	Router(config-if)# switchport	(Optional) Configures the LAN port for Layer 2 switching (required only if the LAN port is not already configured for Layer 2 switching; see the “Configuring a LAN Port for Layer 2 Switching” section on page 7-7).

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk



Note

Complete the steps in the [“Preparing a Layer 2 Switching Port for Configuration as a Trunk”](#) section on page 7-8 before performing the tasks in this section.

To configure the Layer 2 switching port as an ISL or 802.1Q trunk, perform this task:

Command	Purpose
Router(config-if)# switchport trunk encapsulation {isl dot1q negotiate}	(Optional) Configures the encapsulation, which configures the Layer 2 switching port as either an ISL or 802.1Q trunk. Note To support the switchport mode trunk command, you must configure the encapsulation.
Router(config-if)# no switchport trunk encapsulation	Reverts to the default trunk encapsulation mode (negotiate).

**Note**

Complete the steps in the “[Completing Trunk Configuration](#)” section on page 7-13 after performing the tasks in this section.

Configuring the Layer 2 Trunk to Use DTP

**Note**

Complete the steps in the “[Preparing a Layer 2 Switching Port for Configuration as a Trunk](#)” section on page 7-8 before performing the tasks in this section.

To configure the Layer 2 trunk to use DTP, perform this task:

Command	Purpose
Router(config-if)# switchport mode dynamic {auto desirable}	(Optional) Configures the trunk to use DTP.
Router(config-if)# no switchport mode	Reverts to the default trunk trunking mode (switchport mode dynamic desirable).

When configuring the Layer 2 trunk to use DTP, note the following syntax information:

- Required only if the interface is a Layer 2 access port or to specify the trunking mode.
- See [Table 7-2 on page 7-4](#) for information about trunking modes.

**Note**

Complete the steps in the “[Completing Trunk Configuration](#)” section on page 7-13 after performing the tasks in this section.

Configuring the Layer 2 Trunk Not to Use DTP

**Note**

Complete the steps in the “[Preparing a Layer 2 Switching Port for Configuration as a Trunk](#)” section on page 7-8 before performing the tasks in this section.

To configure the Layer 2 trunk not to use DTP, perform this task:

	Command	Purpose
Step 1	Router(config-if)# switchport mode trunk	(Optional) Configures the port to trunk unconditionally.
	Router(config-if)# no switchport mode	Reverts to the default trunk trunking mode (switchport mode dynamic desirable).
Step 2	Router(config-if)# switchport nonegotiate	(Optional) Configures the trunk not to use DTP.
	Router(config-if)# no switchport nonegotiate	Enables DTP on the port.

When configuring the Layer 2 trunk not to use DTP, note the following syntax information:

- Before entering the **switchport mode trunk** command, you must configure the encapsulation (see the “[Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk](#)” section on page 7-8).
- To support the **switchport nonegotiate** command, you must enter the **switchport mode trunk** command.
- Enter the **switchport mode dynamic trunk** command. See [Table 7-2 on page 7-4](#) for information about trunking modes.
- Before entering the **switchport nonegotiate** command, you must configure the encapsulation (see the “[Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk](#)” section on page 7-8) and configure the port to trunk unconditionally with the **switchport mode trunk** command (see the “[Configuring the Layer 2 Trunk to Use DTP](#)” section on page 7-9).

**Note**

Complete the steps in the “[Completing Trunk Configuration](#)” section on page 7-13 after performing the tasks in this section.

Configuring the Default VLAN

**Note**

Complete the steps in the “[Preparing a Layer 2 Switching Port for Configuration as a Trunk](#)” section on page 7-8 before performing the tasks in this section.

To configure the default VLAN, perform this task:

Command	Purpose
Router(config-if)# switchport access vlan <i>vlan_ID</i>	(Optional) Configures the default VLAN, which is used if the interface stops trunking. <ul style="list-style-type: none"> • With Release 12.1(13)E and later releases, the <i>vlan_ID</i> value can be 1 to 4094, except for reserved VLANs (see Table 9-1 on page 9-2). • With 12.1 E releases earlier than Release 12.1(13)E, the <i>vlan_ID</i> value can be 1 to 1005.
Router(config-if)# no switchport access vlan	Reverts to the default value (VLAN 1).

**Note**

Complete the steps in the [“Completing Trunk Configuration”](#) section on page 7-13 after performing the tasks in this section.

Configuring the 802.1Q Native VLAN

**Note**

Complete the steps in the [“Preparing a Layer 2 Switching Port for Configuration as a Trunk”](#) section on page 7-8 before performing the tasks in this section.

To configure the 802.1Q native VLAN, perform this task:

Command	Purpose
Router(config-if)# switchport trunk native vlan <i>vlan_ID</i>	(Optional) Configures the 802.1Q native VLAN.
Router(config-if)# no switchport trunk native vlan	Reverts to the default value (VLAN 1).

When configuring the native VLAN, note the following syntax information:

- With Release 12.1(13)E and later releases, the *vlan_ID* value can be 1 to 4094, except for reserved VLANs (see [Table 9-1 on page 9-2](#)).
- With 12.1 E releases earlier than Release 12.1(13)E, the *vlan_ID* value can be 1 to 1005.
- The access VLAN is not automatically used as the native VLAN.

**Note**

Complete the steps in the [“Completing Trunk Configuration”](#) section on page 7-13 after performing the tasks in this section.

Configuring the List of VLANs Allowed on a Trunk

**Note**

Complete the steps in the [“Preparing a Layer 2 Switching Port for Configuration as a Trunk”](#) section on page 7-8 before performing the tasks in this section.

To configure the list of VLANs allowed on a trunk, perform this task:

Command	Purpose
Router(config-if)# switchport trunk allowed vlan { add except none remove } <i>vlan</i> [, <i>vlan</i> [, <i>vlan</i> [, ...]]	(Optional) Configures the list of VLANs allowed on the trunk.
Router(config-if)# no switchport trunk allowed vlan	Reverts to the default value (all VLANs allowed).

When configuring the list of VLANs allowed on a trunk, note the following syntax information:

- The *vlan* parameter is either a single VLAN ID or a range of VLAN IDs described by two VLAN IDs, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated *vlan* parameters or in dash-specified ranges.
- With Release 12.1(13)E and later releases, the VLAN IDs can be 1 to 4094, except for reserved VLANs (see [Table 9-1 on page 9-2](#)).
- With 12.1 E releases earlier than Release 12.1(13)E, the VLAN IDs can be 1 to 1005.
- All VLANs are allowed by default.
- With Release 12.1(13)E and later releases, you can remove the default VLANs (1002–1005) from a trunk. With earlier releases, you cannot remove any of the default VLANs from a trunk.
- With Release 12.1(11b)E or later, you can remove VLAN 1. If you remove VLAN 1 from a trunk, the trunk interface continues to send and receive management traffic, for example, Cisco Discovery Protocol (CDP), VLAN Trunking Protocol (VTP), Port Aggregation Protocol (PAgP), and DTP in VLAN 1.

**Note**

Complete the steps in the [“Completing Trunk Configuration” section on page 7-13](#) after performing the tasks in this section.

Configuring the List of Prune-Eligible VLANs

**Note**

Complete the steps in the [“Preparing a Layer 2 Switching Port for Configuration as a Trunk” section on page 7-8](#) before performing the tasks in this section.

To configure the list of prune-eligible VLANs on the Layer 2 trunk, perform this task:

Command	Purpose
<pre>Router(config-if)# switchport trunk pruning vlan {none {{add except remove} vlan[,vlan[,vlan[,...]]}}</pre>	(Optional) Configures the list of prune-eligible VLANs on the trunk (see the “Understanding VTP Pruning” section on page 8-3).
<pre>Router(config-if)# no switchport trunk pruning vlan</pre>	Reverts to the default value (all VLANs prune-eligible).

When configuring the list of prune-eligible VLANs on a trunk, note the following syntax information:

- The *vlan* parameter is either a single VLAN ID or a range of VLAN IDs described by two VLAN IDs, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated *vlan* parameters or in dash-specified ranges.
- With Release 12.1(13)E and later releases, the VLAN IDs can be 1 to 4094, except for reserved VLANs (see [Table 9-1 on page 9-2](#)).
- With 12.1 E releases earlier than Release 12.1(13)E, the VLAN IDs can be 1 to 1005.
- The default list of VLANs allowed to be pruned contains all VLANs.
- Network devices in VTP transparent mode do not send VTP Join messages. On Catalyst 6500 series switches with trunk connections to network devices in VTP transparent mode, configure the VLANs used by the transparent-mode network devices or that need to be carried across the transparent-mode network devices as pruning ineligible.

**Note**

Complete the steps in the “[Completing Trunk Configuration](#)” section on page 7-13 after performing the tasks in this section.

Completing Trunk Configuration

To complete Layer 2 trunk configuration, perform this task:

	Command	Purpose
Step 1	Router(config-if)# no shutdown	Activates the interface. (Required only if you shut down the interface.)
Step 2	Router(config-if)# end	Exits configuration mode.

Verifying Layer 2 Trunk Configuration

To verify Layer 2 trunk configuration, perform this task:

	Command	Purpose
Step 1	Router# show running-config interface <i>type</i> ¹ <i>slot/port</i>	Displays the running configuration of the interface.
Step 2	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] switchport	Displays the switch port configuration of the interface.
Step 3	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] trunk	Displays the trunk configuration of the interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

Configuration and Verification Examples

This example shows how to configure the Fast Ethernet port 5/8 as an 802.1Q trunk. This example assumes that the neighbor port is configured to support 802.1Q trunking:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/8
Router(config-if)# shutdown
Router(config-if)# switchport
Router(config-if)# switchport mode dynamic desirable
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# no shutdown
Router(config-if)# end
Router# exit
```

This example shows how to verify the configuration:

```
Router# show running-config interface fastethernet 5/8
Building configuration...
Current configuration:
!
interface FastEthernet5/8
  no ip address
  switchport
  switchport trunk encapsulation dot1q
end
```

```

Router# show interfaces fastethernet 5/8 switchport
Name: Fa5/8
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: ALL

Router# show interfaces fastethernet 5/8 trunk

Port      Mode      Encapsulation  Status      Native vlan
Fa5/8     desirable n-802.1q       trunking    1

Port      Vlans allowed on trunk
Fa5/8 1-1005

Port      Vlans allowed and active in management domain
Fa5/8 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-802,850,917,999,1002-1005

Port      Vlans in spanning tree forwarding state and not pruned
Fa5/8 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-802,850,917,999,1002-1005

Router#

```

Configuring a LAN Interface as a Layer 2 Access Port



Note

If you assign a LAN port to a VLAN that does not exist, the port is shut down until you create the VLAN in the VLAN database (see the “[Creating or Modifying an Ethernet VLAN](#)” section on page 9-10).

To configure a LAN port as a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface type ¹ slot/port	Selects the LAN port to configure.
Step 2	Router(config-if)# shutdown	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching. Note You must enter the switchport command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional switchport commands with keywords.
Step 4	Router(config-if)# no switchport	Clears Layer 2 LAN port configuration.
Step 5	Router(config-if)# switchport mode access Router(config-if)# no switchport mode	Configures the LAN port as a Layer 2 access port. Reverts to the default switchport mode (switchport mode dynamic desirable).

	Command	Purpose
Step 6	Router(config-if)# switchport access vlan <i>vlan_ID</i>	Places the LAN port in a VLAN. <ul style="list-style-type: none"> With Release 12.1(13)E and later releases, the <i>vlan_ID</i> value can be 1 to 4094, except for reserved VLANs (see Table 9-1 on page 9-2). With 12.1 E releases earlier than Release 12.1(13)E, the <i>vlan_ID</i> value can be 1 to 1005.
	Router(config-if)# no switchport access vlan	Reverts to the default VLAN (VLAN 1).
Step 7	Router(config-if)# no shutdown	Activates the interface. (Required only if you shut down the interface.)
Step 8	Router(config-if)# end	Exits configuration mode.
Step 9	Router# show running-config interface [<i>type</i> ¹ <i>slot/port</i>]	Displays the running configuration of the interface.
Step 10	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] switchport	Displays the switch port configuration of the interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure the Fast Ethernet port 5/6 as an access port in VLAN 200:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/6
Router(config-if)# shutdown
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 200
Router(config-if)# no shutdown
Router(config-if)# end
Router# exit
```

This example shows how to verify the configuration:

```
Router# show running-config interface fastethernet 5/6
Building configuration...
!
Current configuration:
interface FastEthernet5/6
  no ip address
  switchport access vlan 200
  switchport mode access
end

Router# show interfaces fastethernet 5/6 switchport
Name: Fa5/6
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Enabled
Access Mode VLAN: 200 (VLAN0200)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: ALL

Router#
```

Configuring a Custom IEEE 802.1Q EtherType Field Value

With Release 12.1(20)E and later releases, you can configure a custom EtherType field value on a port to support network devices that do not use the standard 0x8100 EtherType field value on 802.1Q-tagged or 802.1p-tagged frames.

To configure a custom value for the EtherType field, perform this task:

Command	Purpose
Router(config-if)# switchport dot1q ether-type <i>value</i>	Configures the 802.1Q EtherType field value for the port.
Router(config-if)# no switchport dot1q ether-type	Reverts to the default 802.1Q EtherType field value (0x8100).

When configuring a custom EtherType field value, note the following:

- To use a custom EtherType field value, all network devices in the traffic path across the network must support the custom EtherType field value.
- You can configure a custom EtherType field value on trunk ports, access ports, and tunnel ports.
- Each port supports only one EtherType field value. A port that is configured with a custom EtherType field value does not recognize frames that have any other EtherType field value as tagged frames. For example, a trunk port that is configured with a custom EtherType field value does not recognize the standard 0x8100 EtherType field value on 802.1Q-tagged frames and cannot put the frames into the VLAN to which they belong.



Caution

A port that is configured with a custom EtherType field value considers frames that have any other EtherType field value to be untagged frames. A trunk port with a custom EtherType field value places frames with any other EtherType field value into the native VLAN. An access port or tunnel port with a custom EtherType field value places frames that are tagged with any other EtherType field value into the access VLAN. If you misconfigure a custom EtherType field value, frames might be placed into the wrong VLAN.

- You can configure a custom EtherType field value on these modules:
 - Supervisor engines
 - WS-X6516A-GBIC
 - WS-X6516-GBIC



Note The WS-X6516A-GBIC and WS-X6516-GBIC modules apply a configured custom EtherType field value to all ports supported by each port ASIC (1 through 8 and 9 through 16).

- WS-X6516-GE-TX
- WS-X6748-GE-TX
- WS-X6724-SFP
- WS-X6704-10GE
- WS-X6816-GBIC

- You cannot configure a custom EtherType field value on the ports in an EtherChannel.
- You cannot form an EtherChannel from ports that are configured with custom EtherType field values.

This example shows how to configure the EtherType field value to 0x1234:

```
Router (config-if)# switchport dot1q ethertype 1234  
Router (config-if)#
```




Configuring VTP

This chapter describes how to configure the VLAN Trunking Protocol (VTP) on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How VTP Works, page 8-1](#)
- [VTP Default Configuration, page 8-5](#)
- [VTP Configuration Guidelines and Restrictions, page 8-5](#)
- [Configuring VTP, page 8-6](#)

Understanding How VTP Works

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs within a VTP domain. A VTP domain (also called a VLAN management domain) is made up of one or more network devices that share the same VTP domain name and that are interconnected with trunks. VTP minimizes misconfigurations and configuration inconsistencies that can result in a number of problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations. Before you create VLANs, you must decide whether to use VTP in your network. With VTP, you can make configuration changes centrally on one or more network devices and have those changes automatically communicated to all the other network devices in the network.



Note

For complete information on configuring VLANs, see [Chapter 9, “Configuring VLANs.”](#)

These sections describe how VTP works:

- [Understanding the VTP Domain, page 8-2](#)
- [Understanding VTP Modes, page 8-2](#)
- [Understanding VTP Advertisements, page 8-2](#)
- [Understanding VTP Version 2, page 8-3](#)
- [Understanding VTP Pruning, page 8-3](#)

Understanding the VTP Domain

A VTP domain (also called a VLAN management domain) is made up of one or more interconnected network devices that share the same VTP domain name. A network device can be configured to be in one and only one VTP domain. You make global VLAN configuration changes for the domain using either the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

By default, the Catalyst 6500 series switch is in VTP server mode and is in the no-management domain state until the switch receives an advertisement for a domain over a trunk link or you configure a management domain.

If the switch receives a VTP advertisement over a trunk link, it inherits the management domain name and the VTP configuration revision number. The switch ignores advertisements with a different management domain name or an earlier configuration revision number.

If you configure the switch as VTP transparent, you can create and modify VLANs but the changes affect only the individual switch.

When you make a change to the VLAN configuration on a VTP server, the change is propagated to all network devices in the VTP domain. VTP advertisements are transmitted out all trunk connections, including Inter-Switch Link (ISL), IEEE 802.1Q, IEEE 802.10, and ATM LAN Emulation (LANE).

VTP maps VLANs dynamically across multiple LAN types with unique names and internal index associations. Mapping eliminates excessive device administration required from network administrators.

Understanding VTP Modes

You can configure a Catalyst 6500 series switch to operate in any one of these VTP modes:

- **Server**—In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters (such as VTP version and VTP pruning) for the entire VTP domain. VTP servers advertise their VLAN configuration to other network devices in the same VTP domain and synchronize their VLAN configuration with other network devices based on advertisements received over trunk links. VTP server is the default mode.
- **Client**—VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- **Transparent**—VTP transparent network devices do not participate in VTP. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements. However, in VTP version 2, transparent network devices do forward VTP advertisements that they receive out their trunking LAN ports.

**Note**

Catalyst 6500 series switches automatically change from VTP server mode to VTP client mode if the switch detects a failure while writing configuration to NVRAM. If this happens, the switch cannot be returned to VTP server mode until the NVRAM is functioning.

Understanding VTP Advertisements

Each network device in the VTP domain sends periodic advertisements out each trunking LAN port to a reserved multicast address. VTP advertisements are received by neighboring network devices, which update their VTP and VLAN configurations as necessary.

The following global configuration information is distributed in VTP advertisements:

- VLAN IDs (ISL and 802.1Q)
- Emulated LAN names (for ATM LANE)
- 802.10 SAID values (FDDI)
- VTP domain name
- VTP configuration revision number
- VLAN configuration, including maximum transmission unit (MTU) size for each VLAN
- Frame format

Understanding VTP Version 2

If you use VTP in your network, you must decide whether to use VTP version 1 or version 2.



Note

If you are using VTP in a Token Ring environment, you must use version 2.

VTP version 2 supports the following features not supported in version 1:

- Token Ring support—VTP version 2 supports Token Ring LAN switching and VLANs (Token Ring Bridge Relay Function [TrBRF] and Token Ring Concentrator Relay Function [TrCRF]). For more information about Token Ring VLANs, see the [“Understanding How VLANs Work” section on page 9-1](#).
- Unrecognized Type-Length-Value (TLV) Support—A VTP server or client propagates configuration changes to its other trunks, even for TLVs it is not able to parse. The unrecognized TLV is saved in NVRAM.
- Version-Dependent Transparent Mode—In VTP version 1, a VTP transparent network device inspects VTP messages for the domain name and version, and forwards a message only if the version and domain name match. Because only one domain is supported in the supervisor engine software, VTP version 2 forwards VTP messages in transparent mode without checking the version.
- Consistency Checks—In VTP version 2, VLAN consistency checks (such as VLAN names and values) are performed only when you enter new information through the CLI or SNMP. Consistency checks are not performed when new information is obtained from a VTP message, or when information is read from NVRAM. If the digest on a received VTP message is correct, its information is accepted without consistency checks.

Understanding VTP Pruning

VTP pruning enhances network bandwidth use by reducing unnecessary flooded traffic, such as broadcast, multicast, unknown, and flooded unicast packets. VTP pruning increases available bandwidth by restricting flooded traffic to those trunk links that the traffic must use to access the appropriate network devices. By default, VTP pruning is disabled.

For VTP pruning to be effective, all devices in the management domain must support VTP pruning. On devices that do not support VTP pruning, you must manually configure the VLANs allowed on trunks.

Figure 8-1 shows a switched network without VTP pruning enabled. Interface 1 on network Switch 1 and port 2 on Switch 4 are assigned to the Red VLAN. A broadcast is sent from the host connected to Switch 1. Switch 1 floods the broadcast, and every network device in the network receives it, even though Switches 3, 5, and 6 have no ports in the Red VLAN.

You enable pruning globally on the Catalyst 6500 series switch (see the “Enabling VTP Pruning” section on page 8-7). You configure pruning on Layer 2 trunking LAN ports (see the “Configuring a Layer 2 Switching Port as a Trunk” section on page 7-8).

Figure 8-1 Flooding Traffic without VTP Pruning

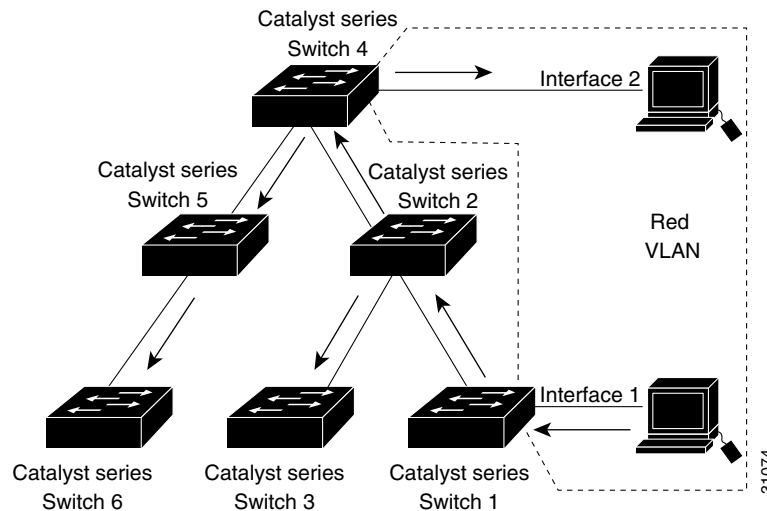
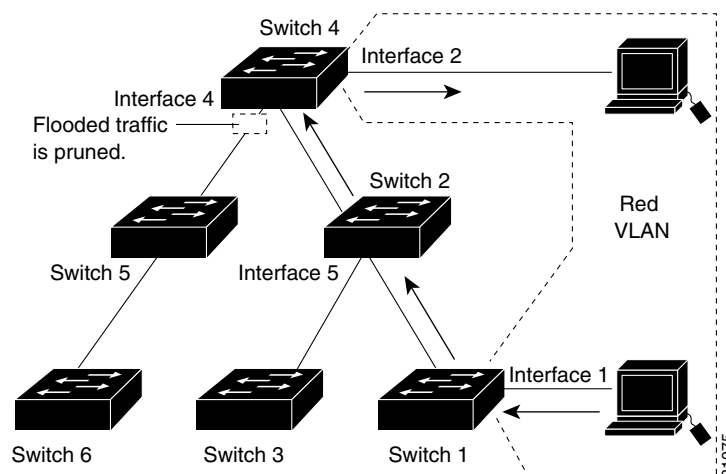


Figure 8-2 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (port 5 on Switch 2 and port 4 on Switch 4).

Figure 8-2 Flooding Traffic with VTP Pruning



Enabling VTP pruning on a VTP server enables pruning for the entire management domain. VTP pruning takes effect several seconds after you enable it. By default, VLANs 2 through 1000 are pruning eligible. VTP pruning does not prune traffic from pruning-eligible VLANs. VLAN 1 is always pruning ineligible; traffic from VLAN 1 cannot be pruned.

To configure VTP pruning on a trunking LAN port, use the **switchport trunk pruning vlan** command (see the “Configuring a Layer 2 Switching Port as a Trunk” section on page 7-8). VTP pruning operates when a LAN port is trunking. You can set VLAN pruning eligibility when VTP pruning is enabled or disabled for the VTP domain, when any given VLAN exists or not, and when the LAN port is currently trunking or not.

VTP Default Configuration

Table 8-1 shows the default VTP configuration.

Table 8-1 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP mode	Server
VTP version 2 enable state	Version 2 is disabled
VTP password	None
VTP pruning	Disabled

VTP Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when implementing VTP in your network:

- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file file_name** command on a switch that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- All network devices in a VTP domain must run the same VTP version.
- You must configure a password on each network device in the management domain when in secure mode.



Caution

If you configure VTP in secure mode, the management domain will not function properly if you do not assign a management domain password to each network device in the domain.

- A VTP version 2-capable network device can operate in the same VTP domain as a network device running VTP version 1 provided VTP version 2 is disabled on the VTP version 2-capable network device (VTP version 2 is disabled by default).
- Do not enable VTP version 2 on a network device unless all of the network devices in the same VTP domain are version 2-capable. When you enable VTP version 2 on a network device, all of the version 2-capable network devices in the domain enable VTP version 2.
- In a Token Ring environment, you must enable VTP version 2 for Token Ring VLAN switching to function properly.

- When you enable or disable VTP pruning on a VTP server, VTP pruning for the entire management domain is enabled or disabled.
- The pruning-eligibility configuration applies globally to all trunks on the switch. You cannot configure pruning-eligibility separately for each trunk.
- When you configure VLANs as pruning eligible or pruning ineligible on a Catalyst 6500 series switch, pruning eligibility for those VLANs is affected on that switch only, not on all network devices in the VTP domain.
- If there is insufficient DRAM available for use by VTP, the VTP mode changes to transparent.
- Network devices in VTP transparent mode do not send VTP Join messages. On Catalyst 6500 series switches with trunk connections to network devices in VTP transparent mode, configure the VLANs that are used by the transparent-mode network devices or that need to be carried across trunks as pruning ineligible. For information about configuring prune eligibility, see the [“Configuring the List of Prune-Eligible VLANs” section on page 7-12](#).

Configuring VTP

These sections describe how to configure VTP:

- [Configuring VTP Global Parameters, page 8-6](#)
- [Configuring the VTP Mode, page 8-8](#)
- [Displaying VTP Statistics, page 8-10](#)

Configuring VTP Global Parameters

These sections describe configuring the VTP global parameters:

- [Configuring a VTP Password, page 8-6](#)
- [Enabling VTP Pruning, page 8-7](#)
- [Enabling VTP Version 2, page 8-7](#)



Note

- In Release 12.1(13)E and later releases, you can enter the VTP global parameters in either global configuration mode or in EXEC mode.
- In earlier releases, you can enter the VTP global parameters only in EXEC mode.

Configuring a VTP Password

To configure the VTP global parameters, perform this task:

Command	Purpose
Router(config)# vtp password <i>password_string</i>	Sets a password, which can be from 8 to 64 characters long, for the VTP domain.
Router(config)# no vtp password	Clears the password.

This example shows one way to configure a VTP password with Release 12.1(13)E and later releases:

```
Router# configure terminal
Router(config)# vtp password WATER
Setting device VLAN database password to WATER.
Router#
```

This example shows how to configure a VTP password with any release:

```
Router# vtp password WATER
Setting device VLAN database password to WATER.
Router#
```



Note

The password is not stored in the running-config file.

Enabling VTP Pruning

To enable VTP pruning in the management domain, perform this task:

	Command	Purpose
Step 1	Router(config)# vtp pruning	Enables VTP pruning in the management domain.
	Router(config)# no vtp pruning	Disables VTP pruning in the management domain.
Step 2	Router# show vtp status	Verifies the configuration.

This example shows one way to enable VTP pruning in the management domain with Release 12.1(13)E and later releases:

```
Router# configure terminal
Router(config)# vtp pruning
Pruning switched ON
```

This example shows how to enable VTP pruning in the management domain with any release:

```
Router# vtp pruning
Pruning switched ON
```

This example shows how to verify the configuration:

```
Router# show vtp status | include Pruning
VTP Pruning Mode: Enabled
Router#
```

For information about configuring prune eligibility, see the [“Configuring the List of Prune-Eligible VLANs”](#) section on page 7-12.

Enabling VTP Version 2

VTP version 2 is disabled by default on VTP version 2-capable network devices. When you enable VTP version 2 on a network device, every VTP version 2-capable network device in the VTP domain enables version 2.



Caution

VTP version 1 and VTP version 2 are not interoperable on network devices in the same VTP domain. Every network device in the VTP domain must use the same VTP version. Do not enable VTP version 2 unless every network device in the VTP domain supports version 2.

**Note**

In a Token Ring environment, you must enable VTP version 2 for Token Ring VLAN switching to function properly on devices that support Token Ring interfaces.

To enable VTP version 2, perform this task:

	Command	Purpose
Step 1	Router(config)# vtp version {1 2}	Enables VTP version 2.
	Router(config)# no vtp version	Reverts to the default (VTP version 1).
Step 2	Router# show vtp status	Verifies the configuration.

This example shows one way to enable VTP version 2 with Release 12.1(13)E and later releases:

```
Router# configure terminal
Router(config)# vtp version 2
V2 mode enabled.
Router(config)#
```

This example shows how to enable VTP version 2 with any release:

```
Router# vtp version 2
V2 mode enabled.
Router#
```

This example shows how to verify the configuration:

```
Router# show vtp status | include v2
VTP V2 Mode: Enabled
Router#
```

Configuring the VTP Mode

To configure the VTP mode, perform this task:

	Command	Purpose
Step 1	Router(config)# vtp mode {client server transparent}	Configures the VTP mode.
	Router(config)# no vtp mode	Reverts to the default VTP mode (server).
Step 2	Router(config)# vtp domain <i>domain_name</i>	(Optional for server mode) Defines the VTP domain name, which can be up to 32 characters long. VTP server mode requires a domain name. If the switch has a trunk connection to a VTP domain, the switch learns the domain name from the VTP server in the domain. Note You cannot clear the domain name.
Step 3	Router(config)# end	Exits VLAN configuration mode.
Step 4	Router# show vtp status	Verifies the configuration.

**Note**

When VTP is disabled, you can enter VLAN configuration commands in configuration mode instead of the VLAN database mode and the VLAN configuration is stored in the startup configuration file.

This example shows how to configure the switch as a VTP server:

```
Router# configure terminal
Router(config)# vtp mode server
Setting device to VTP SERVER mode.
Router(config)# vtp domain Lab_Network
Setting VTP domain name to Lab_Network
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show vtp status
VTP Version                : 2
Configuration Revision     : 255
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 35
VTP Operating Mode         : Server
VTP Domain Name            : Lab_Network
VTP Pruning Mode           : Enabled
VTP V2 Mode                 : Enabled
VTP Traps Generation       : Disabled
MD5 digest                  : 0x08 0x7E 0x54 0xE2 0x5A 0x79 0xA9 0x2D
Configuration last modified by 127.0.0.12 at 8-7-02 11:21:43
Local updater ID is 127.0.0.12 on interface E00/0 (first interface found)

Router#
```

This example shows how to configure the switch as a VTP client:

```
Router# configure terminal
Router(config)# vtp mode client
Setting device to VTP CLIENT mode.
Router(config)# exit
Router#
```

This example shows how to verify the configuration:

```
Router# show vtp status
VTP Version                : 2
Configuration Revision     : 255
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 35
VTP Operating Mode         : Client
VTP Domain Name            : Lab_Network
VTP Pruning Mode           : Enabled
VTP V2 Mode                 : Enabled
VTP Traps Generation       : Disabled
MD5 digest                  : 0x08 0x7E 0x54 0xE2 0x5A 0x79 0xA9 0x2D
Configuration last modified by 127.0.0.12 at 8-7-02 11:21:43

Router#
```

This example shows how to disable VTP on the switch:

```
Router# configure terminal
Router(config)# vtp transparent
Setting device to VTP TRANSPARENT mode.
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show vtp status
VTP Version           : 2
Configuration Revision : 247
Maximum VLANs supported locally : 1005
Number of existing VLANs : 33
VTP Operating Mode    : Transparent
VTP Domain Name       : Lab_Network
VTP Pruning Mode      : Enabled
VTP V2 Mode           : Disabled
VTP Traps Generation  : Disabled
MD5 digest            : 0x45 0x52 0xB6 0xFD 0x63 0xC8 0x49 0x80
Configuration last modified by 0.0.0.0 at 8-12-99 15:04:49
Router#
```

Displaying VTP Statistics

To display VTP statistics, including VTP advertisements sent and received and VTP errors, perform this task:

Command	Purpose
Router# show vtp counters	Displays VTP statistics.

This example shows how to display VTP statistics:

```
Router# show vtp counters
VTP statistics:
Summary advertisements received : 7
Subset advertisements received  : 5
Request advertisements received  : 0
Summary advertisements transmitted : 997
Subset advertisements transmitted : 13
Request advertisements transmitted : 3
Number of config revision errors : 0
Number of config digest errors   : 0
Number of V1 summary errors      : 0

VTP pruning statistics:

Trunk          Join Transmitted Join Received  Summary advts received from
-----          -----
Fa5/8          43071          42766          5
non-pruning-capable device
```




Configuring VLANs

This chapter describes how to configure VLANs on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How VLANs Work, page 9-1](#)
- [VLAN Default Configuration, page 9-6](#)
- [VLAN Configuration Guidelines and Restrictions, page 9-8](#)
- [Configuring VLANs, page 9-9](#)

Understanding How VLANs Work

The following sections describe how VLANs work:

- [VLAN Overview, page 9-1](#)
- [VLAN Ranges, page 9-2](#)
- [Configurable VLAN Parameters, page 9-3](#)
- [Understanding Token Ring VLANs, page 9-3](#)

VLAN Overview

A VLAN is a group of end stations with a common set of requirements, independent of physical location. VLANs have the same attributes as a physical LAN but allow you to group end stations even if they are not located physically on the same LAN segment.

VLANs are usually associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. LAN port VLAN membership is assigned manually on a port-by-port basis.

VLAN Ranges


Note

You must enable the extended system ID to use 4096 VLANs (see the [“Understanding the Bridge ID” section on page 15-3](#)).

With Release 12.1(13)E and later releases, Catalyst 6500 series switches support 4096 VLANs in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges; you use each range slightly differently. Some of these VLANs are propagated to other switches in the network when you use the VLAN Trunking Protocol (VTP). The extended-range VLANs are not propagated, so you must configure extended-range VLANs manually on each network device.

[Table 9-1](#) describes the VLAN ranges.

Table 9-1 VLAN Ranges

VLANs	Range	Usage	Propagated by VTP
0, 4095	Reserved	For system use only. You cannot see or use these VLANs.	—
1	Normal	Cisco default. You can use this VLAN but you cannot delete it.	Yes
2–1001	Normal	For Ethernet VLANs; you can create, use, and delete these VLANs.	Yes
1002–1005	Normal	Cisco defaults for FDDI and Token Ring. You cannot delete VLANs 1002–1005.	Yes
1006–4094	Extended	For Ethernet VLANs only.	No

The following information applies to VLAN ranges:

- Layer 3 LAN ports, WAN interfaces and subinterfaces, and some software features use internal VLANs in the extended range. You cannot use an extended range VLAN that has been allocated for internal use.
- With Release 12.1(13)E and later releases, to display the VLANs used internally, enter the **show vlan internal usage** command. With earlier releases, enter the **show vlan internal usage** and **show cwan vlans** commands.
- With Release 12.1(13)E and later releases, you can configure ascending internal VLAN allocation (from 1006 and up) or descending internal VLAN allocation (from 4094 and down). In previous 12.1EX releases that support 4096 VLANs, internal VLANs are allocated from 1006 and up.
- Switches running the Catalyst operating system do not support configuration of VLANs 1006–1024. If you configure VLANs 1006–1024, ensure that the VLANs do not extend to any switches running Catalyst software.
- You must enable the extended system ID to use extended range VLANs (see the [“Understanding the Bridge ID” section on page 15-3](#)).

Configurable VLAN Parameters

**Note**

- Ethernet VLAN 1 uses only default values.
- Except for the VLAN name, Ethernet VLANs 1006 through 4094 use only default values.
- With Release 12.1(13)E and later releases, you can configure the VLAN name for Ethernet VLANs 1006 through 4094.

You can configure the following parameters for VLANs 2 through 1001:

- VLAN name
- VLAN type (Ethernet, FDDI, FDDI network entity title [NET], TrBRF, or TrCRF)
- VLAN state (active or suspended)
- Security Association Identifier (SAID)
- Bridge identification number for TrBRF VLANs
- Ring number for FDDI and TrCRF VLANs
- Parent VLAN number for TrCRF VLANs
- Spanning Tree Protocol (STP) type for TrCRF VLANs

Understanding Token Ring VLANs

The following section describes the two Token Ring VLAN types supported on network devices running VTP version 2:

- [Token Ring TrBRF VLANs, page 9-3](#)
- [Token Ring TrCRF VLANs, page 9-4](#)

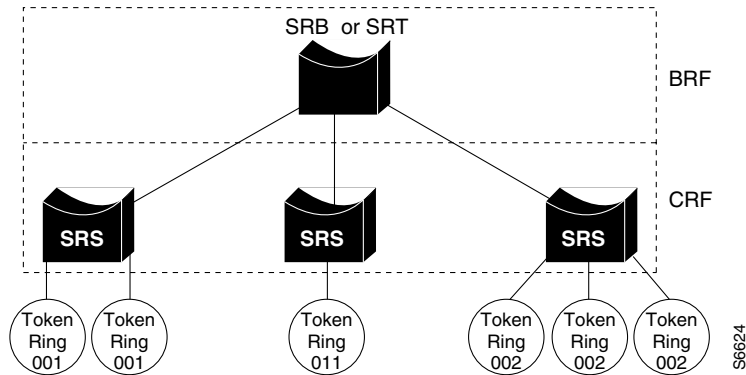
**Note**

Catalyst 6500 series switches do not support Inter-Switch Link (ISL)-encapsulated Token Ring frames. When a Catalyst 6500 series switch is configured as a VTP server, you can configure Token Ring VLANs from the switch.

Token Ring TrBRF VLANs

Token Ring Bridge Relay Function (TrBRF) VLANs interconnect multiple Token Ring Concentrator Relay Function (TrCRF) VLANs in a switched Token Ring network (see [Figure 9-1](#)). The TrBRF can be extended across network devices interconnected with trunk links. The connection between the TrCRF and the TrBRF is referred to as a *logical port*.

Figure 9-1 Interconnected Token Ring TrBRF and TrCRF VLANs



For source routing, the Catalyst 6500 series switch appears as a single bridge between the logical rings. The TrBRF can function as a source-route bridge (SRB) or a source-route transparent (SRT) bridge running either the IBM or IEEE STP. If an SRB is used, you can define duplicate MAC addresses on different logical rings.

The Token Ring software runs an instance of STP for each TrBRF VLAN and each TrCRF VLAN. For TrCRF VLANs, STP removes loops in the logical ring. For TrBRF VLANs, STP interacts with external bridges to remove loops from the bridge topology, similar to STP operation on Ethernet VLANs.

**Caution**

Certain parent TrBRF STP and TrCRF bridge mode configurations can place the logical ports (the connection between the TrBRF and the TrCRF) of the TrBRF in a blocked state. For more information, see the “[VLAN Configuration Guidelines and Restrictions](#)” section on page 9-8.

To accommodate IBM System Network Architecture (SNA) traffic, you can use a combination of SRT and SRB modes. In a mixed mode, the TrBRF determines that some ports (logical ports connected to TrCRFs) operate in SRB mode while other ports operate in SRT mode.

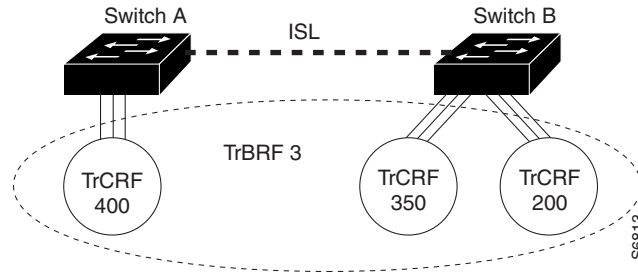
Token Ring TrCRF VLANs

Token Ring Concentrator Relay Function (TrCRF) VLANs define port groups with the same logical ring number. You can configure two types of TrCRFs in your network: undistributed and backup.

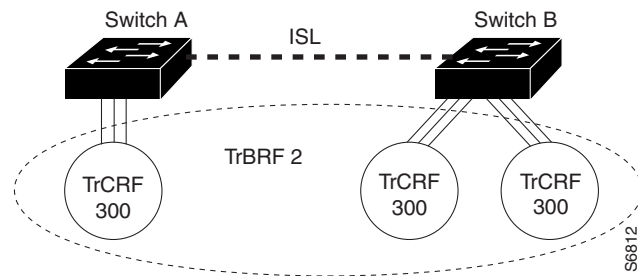
TrCRFs typically are undistributed, which means each TrCRF is limited to the ports on a single network device. Multiple undistributed TrCRFs on the same or separate network devices can be associated with a single parent TrBRF (see [Figure 9-2](#)). The parent TrBRF acts as a multiport bridge, forwarding traffic between the undistributed TrCRFs.

**Note**

To pass data between rings located on separate network devices, you can associate the rings to the same TrBRF and configure the TrBRF for an SRB.

Figure 9-2 Undistributed TrCRFs

By default, Token Ring ports are associated with the default TrCRF (VLAN 1003, trcrf-default), which has the default TrBRF (VLAN 1005, trbrf-default) as its parent. In this configuration, a distributed TrCRF is possible (see [Figure 9-3](#)), and traffic is passed between the default TrCRFs located on separate network devices if the network devices are connected through an ISL trunk.

Figure 9-3 Distributed TrCRF

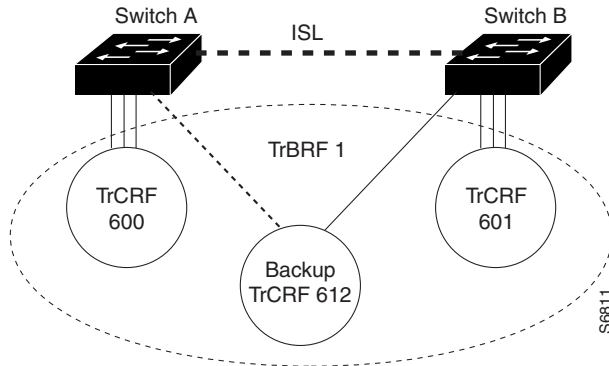
Within a TrCRF, source-route switching forwards frames based on either MAC addresses or route descriptors. The entire VLAN can operate as a single ring, with frames switched between ports within a single TrCRF.

You can specify the maximum hop count for All-Routes and Spanning Tree Explorer frames for each TrCRF. When you specify the maximum hop count, you limit the maximum number of hops an explorer is allowed to traverse. If a port determines that the explorer frame it is receiving has traversed more than the number of hops specified, it does not forward the frame. The TrCRF determines the number of hops an explorer has traversed by the number of bridge hops in the route information field.

If the ISL connection between network devices fails, you can use a backup TrCRF to configure an alternate route for traffic between undistributed TrCRFs. Only one backup TrCRF for a TrBRF is allowed, and only one port per network device can belong to a backup TrCRF.

If the ISL connection between the network devices fails, the port in the backup TrCRF on each affected network device automatically becomes active, rerouting traffic between the undistributed TrCRFs through the backup TrCRF. When the ISL connection is reestablished, all but one port in the backup TrCRF is disabled. [Figure 9-4](#) illustrates the backup TrCRF.

Figure 9-4 Backup TrCRF



VLAN Default Configuration

Tables 9-2 through 9-6 show the default configurations for the different VLAN media types.

Table 9-2 Ethernet VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1	1–4094
VLAN name	“default” for VLAN 1 “VLANvlan_ID” for other Ethernet VLANs	—
802.10 SAID	10vlan_ID	100001–104094
MTU size	1500	1500–18190
Translational bridge 1	0	0–1005
Translational bridge 2	0	0–1005
VLAN state	active	active, suspend
Pruning eligibility	VLANs 2–1001 are pruning eligible; VLANs 1006–4094 are not pruning eligible.	—

Table 9-3 FDDI VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1002	1–1005
VLAN name	“fddi-default”	—
802.10 SAID	101002	1–4294967294
MTU size	1500	1500–18190
Ring number	0	1–4095
Parent VLAN	0	0–1005
Translational bridge 1	0	0–1005

Table 9-3 FDDI VLAN Defaults and Ranges (continued)

Parameter	Default	Range
Translational bridge 2	0	0–1005
VLAN state	active	active, suspend

Table 9-4 Token Ring (TrCRF) VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1003	1–1005
VLAN name	“token-ring-default”	—
802.10 SAID	101003	1–4294967294
Ring Number	0	1–4095
MTU size	VTPv1 default 1500 VTPv2 default 4472	1500–18190
Translational bridge 1	0	0–1005
Translational bridge 2	0	0–1005
VLAN state	active	active, suspend
Bridge mode	srb	srb, srt
ARE max hops	7	0–13
STE max hops	7	0–13
Backup CRF	disabled	disable; enable

Table 9-5 FDDI-Net VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1004	1–1005
VLAN name	“fddinet-default”	—
802.10 SAID	101004	1–4294967294
MTU size	1500	1500–18190
Bridge number	1	0–15
STP type	ieee	auto, ibm, ieee
VLAN state	active	active, suspend

Table 9-6 Token Ring (TrBRF) VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1005	1–1005
VLAN name	“trnet-default”	—
802.10 SAID	101005	1–4294967294

Table 9-6 Token Ring (TrBRF) VLAN Defaults and Ranges (continued)

Parameter	Default	Range
MTU size	VTPv1 1500; VTPv2 4472	1500–18190
Bridge number	1	0–15
STP type	ibm	auto, ibm, ieee
VLAN state	active	active, suspend

VLAN Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when creating and modifying VLANs in your network:

Restrictions

- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file** *file_name* command on a switch that has a redundant supervisor engine.
- RPR+ redundancy (see [Chapter 5, “Configuring RPR and RPR+ Supervisor Engine Redundancy”](#)) does not support a configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode. See the [“VLAN Configuration Options” section on page 9-9](#).
- The Cisco IOS **end** command is not supported in VLAN database mode.
- You cannot enter **Ctrl-Z** to exit VLAN database mode.
- Catalyst 6500 series switches do not support Token Ring or FDDI media. The switch does not forward FDDI, FDDI-Net, TrCRF, or TrBRF traffic, but it can propagate the VLAN configuration through VTP.
- In a Token Ring environment, the logical interfaces (the connection between the TrBRF and the TrCRF) of the TrBRF are placed in a blocked state if either of these conditions exists:
 - The TrBRF is running the IBM STP, and the TrCRF is in SRT mode.
 - The TrBRF is running the IEEE STP, and the TrCRF is in SRB mode.

Guidelines

- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- Before you can create a VLAN, the Catalyst 6500 series switch must be in VTP server mode or VTP transparent mode. For information on configuring VTP, see [Chapter 8, “Configuring VTP.”](#)
- The VLAN configuration is stored in the *vlan.dat* file, which is stored in nonvolatile memory. You can cause inconsistency in the VLAN database if you manually delete the *vlan.dat* file. If you want to modify the VLAN configuration or VTP, use the commands described in this guide and in the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.
- To do a complete backup of your configuration, include the *vlan.dat* file in the backup.

- When a Catalyst 6500 series switch is configured as a VTP server, you can configure FDDI and Token Ring VLANs from the switch.
- You must configure a TrBRF before you configure the TrCRF (the parent TrBRF VLAN you specify must exist).

Configuring VLANs

These sections describe how to configure VLANs:

- [VLAN Configuration Options, page 9-9](#)
- [Creating or Modifying an Ethernet VLAN, page 9-10](#)
- [Assigning a Layer 2 LAN Interface to a VLAN, page 9-12](#)
- [Configuring the Internal VLAN Allocation Policy, page 9-12](#)
- [Mapping 802.1Q VLANs to ISL VLANs, page 9-12](#)



Note

- With releases 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode commands by entering the **do** keyword before the EXEC mode command.
- VLANs support a number of parameters that are not discussed in detail in this section. For complete information, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

VLAN Configuration Options

These sections describe the VLAN configuration options:

- [VLAN Configuration in Global Configuration Mode, page 9-9](#)
- [VLAN Configuration in VLAN Database Mode, page 9-10](#)

VLAN Configuration in Global Configuration Mode



Note

Releases 12.1(11b)E and later support VLAN configuration in global configuration mode.

If the switch is in VTP server or transparent mode (see the “[Configuring VTP](#)” section on page 8-6), you can configure VLANs in global and config-vlan configuration modes. When you configure VLANs in global and config-vlan configuration modes, the VLAN configuration is saved in the vlan.dat files. To display the VLAN configuration, enter the **show vlan** command.

If the switch is in VLAN transparent mode, use the copy **running-config startup-config** command to save the VLAN configuration to the startup-config file. After you save the running configuration as the startup configuration, use the **show running-config** and **show startup-config** commands to display the VLAN configuration.



Note

- When the switch boots, if the VTP domain name and VTP mode in the startup-config and vlan.dat files do not match, the switch uses the configuration in the vlan.dat file.

- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode.

VLAN Configuration in VLAN Database Mode



Note

You cannot configure extended-range VLANs in VLAN database mode. You can configure extended-range VLANs only in global configuration mode. RPR+ redundancy does not support configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.

If the switch is in VTP server or transparent mode, you can configure VLANs in the VLAN database mode. When you configure VLANs in VLAN database mode, the VLAN configuration is saved in the `vlan.dat` files. To display the VLAN configuration, enter the **show vlan** command.

You use the interface configuration command mode to define the port membership mode and add and remove ports from a VLAN. The results of these commands are written to the running-config file, and you can display the file by entering the **show running-config** command.

Creating or Modifying an Ethernet VLAN

User-configured VLANs have unique IDs from 1 to 4094, except for reserved VLANs (see [Table 9-1 on page 9-2](#)). Enter the **vlan** command with an unused ID to create a VLAN. Enter the **vlan** command for an existing VLAN to modify the VLAN (you cannot modify an existing VLAN that is being used by a Layer 3 port or a software feature).

See the “[VLAN Default Configuration](#)” section on [page 9-6](#) for the list of default parameters that are assigned when you create a VLAN. If you do not specify the VLAN type with the **media** keyword, the VLAN is an Ethernet VLAN.

To create or modify a VLAN, perform this task:

	Command	Purpose
Step 1	Router# configure terminal OR Router# vlan database	Enters VLAN configuration mode.
Step 2	Router(config)# vlan <i>vlan_ID</i> { [- <i>vlan_ID</i>] [, <i>vlan_ID</i>]} Router(config-vlan)# OR Router(vlan)# vlan <i>vlan_ID</i> Router(config)# no vlan <i>vlan_ID</i> Router(config-vlan)# OR Router(vlan)# no vlan <i>vlan_ID</i>	Creates or modifies an Ethernet VLAN, a range of Ethernet VLANs, or several Ethernet VLANs specified in a comma-separated list (do not enter space characters). Deletes a VLAN.
Step 3	Router(config-vlan)# end OR Router(vlan)# exit	Updates the VLAN database and returns to privileged EXEC mode.
Step 4	Router# show vlan [<i>id</i> <i>name</i>] <i>vlan</i>	Verifies the VLAN configuration.

When you create or modify an Ethernet VLAN, note the following syntax information:

- Releases 12.1(11b)E and later support VLAN configuration in global configuration mode.
- Releases 12.1(13)E and later support extended-range VLANs.
- RPR+ redundancy does not support a configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.
- Because Layer 3 ports and some software features require internal VLANs allocated from 1006 and up, configure extended-range VLANs starting with 4094.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode.
- Layer 3 ports and some software features use extended-range VLANs. If the VLAN you are trying to create or modify is being used by a Layer 3 port or a software feature, the switch displays a message and does not modify the VLAN configuration.

When deleting VLANs, note the following syntax information:

- You cannot delete the default VLANs for the different media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005.
- When you delete a VLAN, any LAN ports configured as access ports assigned to that VLAN become inactive. The ports remain associated with the VLAN (and inactive) until you assign them to a new VLAN.

This example shows how to create an Ethernet VLAN in global configuration mode and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 3
Router(config-vlan)# end
Router# show vlan id 3
```

VLAN Name	Status	Ports
3 VLAN0003	active	

VLAN Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
3	enet	100003	1500	-	-	-	-	0	0

Primary	Secondary	Type	Interfaces

This example shows how to create an Ethernet VLAN in VLAN database mode:

```
Router# vlan database
Router(vlan)# vlan 3
VLAN 3 added:
    Name: VLAN0003
Router(vlan)# exit
APPLY completed.
Exiting....
```

This example shows how to verify the configuration:

```
Router# show vlan name VLAN0003
```

VLAN Name	Status	Ports
3 VLAN0003	active	

VLAN Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	Trans1	Trans2
3	enet	100003	1500	-	-	-	0	0

```
Router#
```

Assigning a Layer 2 LAN Interface to a VLAN

A VLAN created in a management domain remains unused until you assign one or more LAN ports to the VLAN.



Note

Make sure you assign LAN ports to a VLAN of the appropriate type. Assign Ethernet ports to Ethernet-type VLANs.

To assign one or more LAN ports to a VLAN, complete the procedures in the [“Configuring LAN Interfaces for Layer 2 Switching”](#) section on page 7-7.

Configuring the Internal VLAN Allocation Policy


Internal VLAN allocation policy is supported in Release 12.1(13)E and later releases. For more information about VLAN allocation, see the [“VLAN Ranges”](#) section on page 9-2.



Note

The internal VLAN allocation policy is applied only following a reload.

To configure the internal VLAN allocation policy, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan internal allocation policy {ascending descending}	Configures the internal VLAN allocation policy.
	Router(config)# no vlan internal allocation policy	Returns to the default (ascending).
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# reload	Applies the new internal VLAN allocation policy.
		 Caution You do not need to enter the reload command immediately. Enter the reload command during a planned maintenance window.

When you configure the internal VLAN allocation policy, note the following syntax information:

- Enter the **ascending** keyword to allocate internal VLANs from 1006 and up.
- Enter the **descending** keyword to allocate internal VLAN from 4094 and down.

This example shows how to configure descending as the internal VLAN allocation policy:

```
Router# configure terminal
Router(config)# vlan internal allocation policy descending
```

Mapping 802.1Q VLANs to ISL VLANs

The valid range of user-configurable ISL VLANs is 1 through 1001 and 1006 through 4094. The valid range of VLANs specified in the IEEE 802.1Q standard is 1 to 4094. You can map 802.1Q VLAN numbers to ISL VLAN numbers.

802.1Q VLANs in the range 1 through 1001 and 1006 through 4094 are automatically mapped to the corresponding ISL VLAN. 802.1Q VLAN numbers corresponding to reserved VLAN numbers must be mapped to an ISL VLAN in order to be recognized and forwarded by Cisco network devices.

These restrictions apply when mapping 802.1Q VLANs to ISL VLANs:

- You can configure up to eight 802.1Q-to-ISL VLAN mappings on the Catalyst 6500 series switch.
- You can only map 802.1Q VLANs to Ethernet-type ISL VLANs.
- Do not enter the native VLAN of any 802.1Q trunk in the mapping table.
- When you map an 802.1Q VLAN to an ISL VLAN, traffic on the 802.1Q VLAN corresponding to the mapped ISL VLAN is blocked. For example, if you map 802.1Q VLAN 1007 to ISL VLAN 200, traffic on 802.1Q VLAN 200 is blocked.
- VLAN mappings are local to each Catalyst 6500 series switch. Make sure you configure the same VLAN mappings on all appropriate network devices.

To map an 802.1Q VLAN to an ISL VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan mapping dot1q <i>dot1q_vlan</i> isl <i>isl_vlan</i>	Maps an 802.1Q VLAN to an ISL Ethernet VLAN. The valid range for <i>dot1q_vlan</i> is 1001 to 4094. The valid range for <i>isl_vlan</i> is the same.
	Router(config)# no vlan mapping dot1q { all <i>dot1q_vlan</i> }	Deletes the mapping.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show vlan	Verifies the VLAN mapping.

This example shows how to map 802.1Q VLAN 1003 to ISL VLAN 200:

```
Router# configure terminal
Router(config)# vlan mapping dot1q 1003 isl 200
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show vlan
<...output truncated...>
802.1Q Trunk Remapped VLANs:
802.1Q VLAN      ISL VLAN
-----
      1003          200
```




Configuring Private VLANs

This chapter describes how to configure private VLANs on the Catalyst 6500 series switches. Release 12.1 E supports private VLANs with Release 12.1(11b)E and later.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How Private VLANs Work, page 10-1](#)
- [Private VLAN Configuration Restrictions and Guidelines, page 10-2](#)
- [Configuring Private VLANs, page 10-5](#)

Understanding How Private VLANs Work



Note

To configure private VLANs, the switch must be in VTP transparent mode.

Private VLANs provide Layer 2 isolation between ports within the same private VLAN. There are three types of private VLAN ports:

- **Promiscuous**—A promiscuous port can communicate with all interfaces, including the community and isolated ports within a private VLAN.
- **Isolated**—An isolated port has complete Layer 2 separation from other ports within the same private VLAN except for the promiscuous port. Private VLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports.
- **Community**—Community ports communicate among themselves and with their promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities or isolated ports within their private VLAN.



Note

Because trunks can support the VLANs carrying traffic between isolated, community, and promiscuous ports, isolated and community port traffic might enter or leave the switch through a trunk interface.

Private VLAN ports are associated with a set of supporting VLANs that are used to create the private VLAN structure. A private VLAN uses VLANs three ways:

- Primary VLAN—Carries traffic from promiscuous ports to isolated, community, and other promiscuous ports.
- Isolated VLAN—Carries traffic from isolated ports to promiscuous ports.
- Community VLAN—Carries traffic between community ports and to promiscuous ports. You can configure multiple community VLANs in a private VLAN.

**Note**

Isolated and community VLANs are both called secondary VLANs.

You can extend private VLANs across multiple devices by trunking the primary, isolated, and community VLANs to other devices that support private VLANs.

In a switched environment, you can assign an individual private VLAN and associated IP subnet to each individual or common group of end stations. The end stations only need to communicate with a default gateway to gain access outside the private VLAN. With end stations in a private VLAN, you can do the following:

- Designate selected ports connected to end stations (for example, interfaces connected to servers) as isolated to prevent any communication at Layer 2. (For example, if the end stations were servers, this configuration would prevent Layer 2 communication between the servers.)
- Designate the interfaces to which the default gateway(s) and selected end stations (for example, backup servers or LocalDirector) are attached as promiscuous to allow all end stations access.
- Reduce VLAN and IP subnet consumption by preventing traffic between end stations even though they are in the same VLAN and IP subnet.

A promiscuous port can serve only one primary VLAN.

A promiscuous port can serve as many isolated or community VLANs as desired.

With a promiscuous port, you can connect a wide range of devices as “access points” to a private VLAN. For example, you can connect a promiscuous port to the “server port” of LocalDirector to connect an isolated VLAN or a number of community VLANs to the server so that LocalDirector can load balance the servers present in the isolated or community VLANs, or you can use a promiscuous port to monitor or back up all the private VLAN servers from an administration workstation.

Private VLAN Configuration Restrictions and Guidelines

Follow these restrictions and guidelines to configure private VLANs:

- Set VTP to transparent mode. After you configure a private VLAN, you cannot change the VTP mode to client or server. See [Chapter 8, “Configuring VTP.”](#)
- You cannot include VLAN 1 or VLANs 1002 to 1005 in the private VLAN configuration.
- Use only the private VLAN configuration commands to assign ports to primary, isolated, or community VLANs. Layer 2 access ports assigned to the VLANs that you configure as primary, isolated, or community VLANs are inactive while the VLAN is part of the private VLAN configuration. Layer 2 trunk interfaces remain in the STP forwarding state.
- Configure Layer 3 VLAN interfaces only for primary VLANs. Layer 3 VLAN interfaces for isolated and community VLANs are inactive while the VLAN is configured as an isolated or community VLAN.

- Do not configure private VLAN ports as EtherChannels. While a port is part of the private VLAN configuration, any EtherChannel configuration for it is inactive.
- Destination SPAN configuration supersedes private VLAN configuration. While a port is a destination SPAN port, any private VLAN configuration for it is inactive.
- Private VLANs support the following SPAN features:
 - You can configure a private VLAN port as a SPAN source port.
 - You can use VLAN-based SPAN (VSPAN) on primary, isolated, and community VLANs, or use SPAN on only one VLAN to separately monitor egress or ingress traffic.

For more information about SPAN, see [Chapter 34, “Configuring Local SPAN and RSPAN.”](#)

- A primary VLAN can have one isolated VLAN and multiple community VLANs associated with it.
- An isolated or community VLAN can have only one primary VLAN associated with it.
- Enable PortFast and BPDU guard on isolated and community ports to prevent STP loops due to misconfigurations and to speed up STP convergence (see [Chapter 16, “Configuring Optional STP Features”](#)). When enabled, STP applies the BPDU guard feature to all PortFast-configured Layer 2 LAN ports.
- If you delete a VLAN used in the private VLAN configuration, the private VLAN ports associated with the VLAN become inactive.

- **12-Port Restriction:**

- In all releases, the “12-port restriction” applies to these 10 Mb, 10/100 Mb, and 100 Mb Ethernet switching modules: WS-X6324-100FX, WS-X6348-RJ-45, WS-X6348-RJ-45V, WS-X6348-RJ-21V, WS-X6248-RJ-45, WS-X6248A-TEL, WS-X6248-TEL, WS-X6148-RJ-45, WS-X6148-RJ-45V, WS-X6148-45AF, WS-X6148-RJ-21, WS-X6148-RJ-21V, WS-X6148-21AF, WS-X6024-10FL-MT.
- In releases earlier than Release 12.1(19)E, the “12-port restriction” applies to these Ethernet switching modules: WS-X6548-RJ-45, WS-X6548-RJ-21, WS-X6524-100FX-MM.
- In Release 12.1(19)E and later releases, the “12-port restriction” does not apply to these Ethernet switching modules: WS-X6548-RJ-45, WS-X6548-RJ-21, WS-X6524-100FX-MM (CSCea67876).

Within groups of 12 ports (1–12, 13–24, 25–36, and 37–48), do not configure ports as isolated or community VLAN ports when one port within the 12 ports is a trunk or a SPAN destination or a promiscuous private VLAN port. While one port within the 12 ports is a trunk or a SPAN destination or a promiscuous private VLAN port, any isolated or community VLAN configuration for other ports within the 12 ports is inactive. To reactivate the ports, remove the isolated or community VLAN port configuration and enter **shutdown** and **no shutdown** commands.

- **24-Port Restriction:**

- In all releases, this “24-port restriction” applies to the WS-X6548-GE-TX and WS-X6148-GE-TX 10/100/1000 Mb Ethernet switching modules: within groups of 24 ports (1–24, 25–48), do not configure ports as isolated or community VLAN ports when one port within the 24 ports is a trunk or a SPAN destination or a promiscuous private VLAN port. While one port within the 24 ports is a trunk or a SPAN destination or a promiscuous private VLAN port, any isolated or community VLAN configuration for other ports within the 24 ports is inactive. To reactivate the ports, remove the isolated or community VLAN port configuration and enter **shutdown** and **no shutdown** commands.
- Private VLAN ports can be on different network devices as long as the devices are trunk connected and the primary and secondary VLANs have not been removed from the trunk.

- VTP does not support private VLANs. You must configure private VLANs on each device where you want private VLAN ports.
- To maintain the security of your private VLAN configuration and avoid other use of the VLANs configured as private VLANs, configure private VLANs on all intermediate devices, including devices that have no private VLAN ports.
- We recommend that you prune the private VLANs from the trunks on devices that carry no traffic in the private VLANs.
- In networks with some devices using MAC address reduction, and others not using MAC address reduction, STP parameters do not necessarily propagate to ensure that the spanning tree topologies match. You should manually check the STP configuration to ensure that the primary, isolated, and community VLANs' spanning tree topologies match.
- If you enable MAC address reduction on the switch, we recommend that you enable MAC address reduction on all the devices in your network to ensure that the STP topologies of the private VLANs match.
- In a network where private VLANs are configured, if you enable MAC address reduction on some devices and disable it on others (mixed environment), use the default bridge priorities to make sure that the root bridge is common to the primary VLAN and to all its associated isolated and community VLANs. Be consistent with the ranges employed by the MAC address reduction feature regardless of whether it is enabled on the system. MAC address reduction allows only discrete levels and uses all intermediate values internally as a range. You should disable a root bridge with private VLANs and MAC address reduction, and configure the root bridge with any priority higher than the highest priority range used by any nonroot bridge.
- You can apply different quality of service (QoS) configuration to primary, isolated, and community VLANs (see [Chapter 31, “Configuring PFC QoS”](#)).
- You cannot apply VACLs to secondary VLANs (see the [“Configuring VLAN ACLs” section on page 23-8](#)).
- To apply Cisco IOS output ACLs to all outgoing private VLAN traffic, configure them on the Layer 3 VLAN interface of the primary VLAN (see [Chapter 23, “Configuring Network Security”](#)).
- Cisco IOS ACLs applied to the Layer 3 VLAN interface of a primary VLAN automatically apply to the associated isolated and community VLANs.
- Do not apply Cisco IOS ACLs to isolated or community VLANs. Cisco IOS ACL configuration applied to isolated and community VLANs is inactive while the VLANs are part of the private VLAN configuration.
- Do not apply dynamic access control entries (ACEs) to primary VLANs. Cisco IOS dynamic ACL configuration applied to a primary VLAN is inactive while the VLAN are part of the private VLAN configuration.
- ARP entries learned on Layer 3 private VLAN interfaces are sticky ARP entries (we recommend that you display and verify private VLAN interface ARP entries).
- For security reasons, private VLAN port sticky ARP entries do not age out. Connecting a device with a different MAC address but with the same IP address generates a message and the ARP entry is not created.

- Because the private VLAN port sticky ARP entries do not age out, you must manually remove private VLAN port ARP entries if a MAC address changes. You can add or remove private VLAN ARP entries manually as follows:

```
Router(config)# no arp 11.1.3.30
IP ARP:Deleting Sticky ARP entry 11.1.3.30
```

```
Router(config)# arp 11.1.3.30 0000.5403.2356 arpa
IP ARP:Overwriting Sticky ARP entry 11.1.3.30, hw:00d0.bb09.266e by hw:0000.5403.2356
```

Configuring Private VLANs

These sections describe how to configure private VLANs:

- [Configuring a VLAN as a Private VLAN, page 10-5](#)
- [Associating Secondary VLANs with a Primary VLAN, page 10-6](#)
- [Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN, page 10-7](#)
- [Configuring a Layer 2 Interface as a Private VLAN Host Port, page 10-8](#)
- [Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port, page 10-9](#)



Note

If the VLAN is not defined already, the private VLAN configuration process defines it.



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Configuring a VLAN as a Private VLAN

To configure a VLAN as a private VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan <i>vlan_ID</i>	Enters VLAN configuration submode.
Step 2	Router(config-vlan)# private-vlan { community isolated primary } Router(config-vlan)# no private-vlan { community isolated primary }	Configures a VLAN as a private VLAN. Clears the private VLAN configuration. Note These commands do not take effect until you exit VLAN configuration submode.
Step 3	Router(config-vlan)# end	Exits configuration mode.
Step 4	Router# show vlan private-vlan [type]	Verifies the configuration.

This example shows how to configure VLAN 202 as a primary VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan primary
Router(config-vlan)# end
Router# show vlan private-vlan
```

```

Primary Secondary Type          Interfaces
-----
202                primary

```

This example shows how to configure VLAN 303 as a community VLAN and verify the configuration:

```

Router# configure terminal
Router(config)# vlan 303
Router(config-vlan)# private-vlan community
Router(config-vlan)# end
Router# show vlan private-vlan

```

```

Primary Secondary Type          Interfaces
-----
202                primary
                   303        community

```

This example shows how to configure VLAN 440 as an isolated VLAN and verify the configuration:

```

Router# configure terminal
Router(config)# vlan 440
Router(config-vlan)# private-vlan isolated
Router(config-vlan)# end
Router# show vlan private-vlan

```

```

Primary Secondary Type          Interfaces
-----
202                primary
                   303        community
                   440        isolated

```

Associating Secondary VLANs with a Primary VLAN

To associate secondary VLANs with a primary VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan <i>primary_vlan_ID</i>	Enters VLAN configuration submode for the primary VLAN.
Step 2	Router(config-vlan)# private-vlan association { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> } Router(config-vlan)# no private-vlan association	Associates the secondary VLANs with the primary VLAN. Clears all secondary VLAN associations.
Step 3	Router(config-vlan)# end	Exits VLAN configuration mode.
Step 4	Router# show vlan private-vlan [<i>type</i>]	Verifies the configuration.

When you associate secondary VLANs with a primary VLAN, note the following syntax information:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs.
- The *secondary_vlan_list* parameter can contain multiple community VLAN IDs.
- The *secondary_vlan_list* parameter can contain only one isolated VLAN ID.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to associate secondary VLANs with a primary VLAN.

- Use the **remove** keyword with a *secondary_vlan_list* to clear the association between secondary VLANs and a primary VLAN.
- The command does not take effect until you exit VLAN configuration submode.

This example shows how to associate community VLANs 303 through 307 and 309 and isolated VLAN 440 with primary VLAN 202 and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan association 303-307,309,440
Router(config-vlan)# end
Router# show vlan private-vlan
```

Primary	Secondary	Type	Interfaces
202	303	community	
202	304	community	
202	305	community	
202	306	community	
202	307	community	
202	309	community	
202	440	isolated	
	308	community	

Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN



Note

Isolated and community VLANs are both called secondary VLANs.

To map secondary VLANs to the Layer 3 VLAN interface of a primary VLAN to allow Layer 3 switching of private VLAN ingress traffic, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>primary_vlan_ID</i>	Enters interface configuration mode for the primary VLAN.
Step 2	Router(config-if)# private-vlan mapping { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	Maps the secondary VLANs to the Layer 3 VLAN interface of a primary VLAN to allow Layer 3 switching of private VLAN ingress traffic.
	Router(config-if)# [no] private-vlan mapping	Clears the mapping between the secondary VLANs and the primary VLAN.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show interface private-vlan mapping	Verifies the configuration.

When you map secondary VLANs to the Layer 3 VLAN interface of a primary VLAN, note the following syntax information:

- The **private-vlan mapping** interface configuration command only affects private VLAN ingress traffic that is Layer 3 switched.
- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs.

- Enter a *secondary_vlan_list* parameter or use the **add** keyword with a *secondary_vlan_list* parameter to map the secondary VLANs to the primary VLAN.
- Use the **remove** keyword with a *secondary_vlan_list* parameter to clear the mapping between secondary VLANs and the primary VLAN.

This example shows how to permit routing of secondary VLAN ingress traffic from private VLANs 303 through 307, 309, and 440 and verify the configuration:

```
Router# configure terminal
Router(config)# interface vlan 202
Router(config-if)# private-vlan mapping add 303-307,309,440
Router(config-if)# end
Router# show interfaces private-vlan mapping
Interface Secondary VLAN Type
-----
vlan202    303          community
vlan202    304          community
vlan202    305          community
vlan202    306          community
vlan202    307          community
vlan202    309          community
vlan202    440          isolated

Router#
```

Configuring a Layer 2 Interface as a Private VLAN Host Port

To configure a Layer 2 interface as a private VLAN host port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching: <ul style="list-style-type: none"> • You must enter the switchport command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional switchport commands with keywords. • Required only if you have not entered the switchport command already for the interface.
Step 3	Router(config-if)# switchport mode private-vlan { host promiscuous }	Configures the Layer 2 port as a private VLAN host port.
	Router(config-if)# no switchport mode private-vlan	Clears private VLAN port configuration.
Step 4	Router(config-if)# switchport private-vlan host-association <i>primary_vlan_ID</i> <i>secondary_vlan_ID</i>	Associates the Layer 2 port with a private VLAN.
	Router(config-if)# no switchport private-vlan host-association	Clears the association.
Step 5	Router(config-if)# end	Exits configuration mode.
Step 6	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] switchport	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure interface FastEthernet 5/1 as a private VLAN host port and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport mode private-vlan host
Router(config-if)# switchport private-vlan host-association 202 303
Router(config-if)# end
Router# show interfaces fastethernet 5/1 switchport
Name: Fa5/1
Switchport: Enabled
→ Administrative Mode: private-vlan host
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
→ Administrative private-vlan host-association: 202 (VLAN0202) 303 (VLAN0303)
Administrative private-vlan mapping: none
→ Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port

To configure a Layer 2 interface as a private VLAN promiscuous port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN interface to configure.
Step 2	Router(config-if)# switchport	Configures the LAN interface for Layer 2 switching: <ul style="list-style-type: none"> You must enter the switchport command once without any keywords to configure the LAN interface as a Layer 2 interface before you can enter additional switchport commands with keywords. Required only if you have not entered the switchport command already for the interface.
Step 3	Router(config-if)# switchport mode private-vlan { host promiscuous }	Configures the Layer 2 port as a private VLAN promiscuous port.
	Router(config-if)# no switchport mode private-vlan	Clears the private VLAN port configuration.
Step 4	Router(config-if)# switchport private-vlan mapping primary_vlan_ID { <i>secondary_vlan_list</i> add secondary_vlan_list remove secondary_vlan_list }	Maps the private VLAN promiscuous port to a primary VLAN and to selected secondary VLANs.
	Router(config-if)# no switchport private-vlan mapping	Clears all mapping between the private VLAN promiscuous port and the primary VLAN and any secondary VLANs.
Step 5	Router(config-if)# end	Exits configuration mode.
Step 6	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] switchport	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When you configure a Layer 2 interface as a private VLAN promiscuous port, note the following syntax information:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs.
- Enter a *secondary_vlan_list* value or use the **add** keyword with a *secondary_vlan_list* value to map the secondary VLANs to the private VLAN promiscuous port.
- Use the **remove** keyword with a *secondary_vlan_list* value to clear the mapping between secondary VLANs and the private VLAN promiscuous port.

This example shows how to configure interface FastEthernet 5/2 as a private VLAN promiscuous port and map it to a private VLAN:

```
Router# configure terminal
Router(config)# interface fastethernet 5/2
Router(config-if)# switchport mode private-vlan promiscuous
Router(config-if)# switchport private-vlan mapping 202 303,440
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show interfaces fastethernet 5/2 switchport
Name: Fa5/2
Switchport: Enabled
→ Administrative Mode: private-vlan promiscuous
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none ((Inactive))
→ Administrative private-vlan mapping: 202 (VLAN0202) 303 (VLAN0303) 440 (VLAN0440)
→ Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```




Configuring Cisco IP Phone Support

This chapter describes how to configure support for Cisco IP Phones on the Catalyst 6500 series switches. Release 12.1(13)E and later releases support Cisco IP Phones.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication for this release.

This chapter consists of these sections:

- [Understanding Cisco IP Phone Support, page 11-1](#)
- [Default Cisco IP Phone Support Configuration, page 11-4](#)
- [Cisco IP Phone Support Configuration Guidelines and Restrictions, page 11-4](#)
- [Configuring Cisco IP Phone Support, page 11-5](#)

Understanding Cisco IP Phone Support

These sections describe Cisco IP Phone support:

- [Cisco IP Phone Connections, page 11-1](#)
- [Cisco IP Phone Voice Traffic, page 11-2](#)
- [Cisco IP Phone Data Traffic, page 11-3](#)
- [Cisco IP Phone Power Configurations, page 11-3](#)

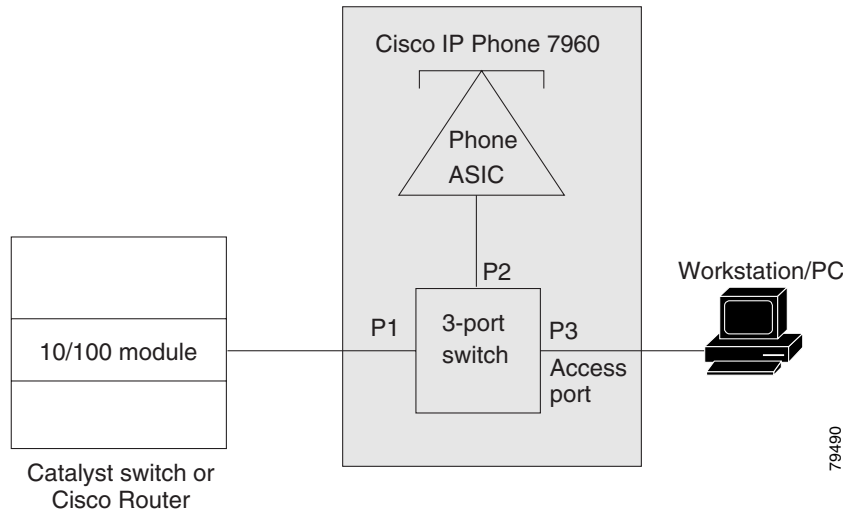
Cisco IP Phone Connections

The Cisco IP Phone contains an integrated 3-port 10/100 switch. The ports are dedicated connections to these devices:

- Port 1 connects to the switch.
- Port 2 is an internal 10/100 interface that carries the Cisco IP Phone traffic.
- Port 3 connects to a PC or other device.

[Figure 11-1](#) shows a Cisco IP Phone connected between a switch and a PC.

Figure 11-1 Cisco IP Phone Connected to a Switch



Cisco IP Phone Voice Traffic

The Cisco IP Phone transmits voice traffic with Layer 3 IP precedence and Layer 2 CoS values, which are both set to 5 by default. The sound quality of a Cisco IP Phone call can deteriorate if the voice traffic is transmitted unevenly. To provide more predictable voice traffic flow, you can configure QoS to trust the Layer 3 IP precedence or Layer 2 CoS value in the voice traffic (refer to [Chapter 31, “Configuring PFC QoS”](#)).



Note

You can configure the ports on WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules to trust received Layer 2 CoS values (QoS port architecture 1p1q0t/1p3q1t). The WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules cannot supply power to Cisco IP Phones. Configure QoS policies that use the Layer 3 IP precedence value on other switching modules.

You can configure a Layer 2 access port with an attached Cisco IP Phone to use one VLAN for voice traffic and another VLAN for data traffic from a device attached to the Cisco IP Phone.

You can configure Layer 2 access ports on the switch to send Cisco Discovery Protocol version 2 (CDPv2) packets that instruct an attached Cisco IP Phone to transmit voice traffic to the switch in any of the following ways:

- In the voice VLAN, tagged with a Layer 2 CoS priority value
- In the access VLAN, tagged with a Layer 2 CoS priority value
- In the access VLAN, untagged (no Layer 2 CoS priority value)



Note

In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

You cannot use Cisco IOS software commands to configure the frame type used by data traffic sent from a device attached to the access port on the Cisco IP Phone.

Cisco IP Phone Data Traffic

**Note**

Untagged traffic from the device attached to the Cisco IP Phone passes through the Cisco IP Phone unchanged, regardless of the trust state of the access port on the Cisco IP Phone.

To process tagged data traffic (traffic in 802.1Q or 802.1p frame types) from the device attached to the access port on the Cisco IP Phone (see [Figure 11-1](#)), you can configure Layer 2 access ports on the switch to send CDPv2 packets that instruct an attached Cisco IP Phone to configure the access port on the Cisco IP Phone to either of these two modes:

- Trusted mode—All traffic received through the access port on the Cisco IP Phone passes through the Cisco IP Phone unchanged.
- Untrusted mode—All traffic in 802.1Q or 802.1p frames received through the access port on the Cisco IP Phone is marked with a configured Layer 2 CoS value. The default Layer 2 CoS value is 0. Untrusted mode is the default.

Cisco IP Phone Power Configurations

These sections describe Cisco IP Phone power configurations:

- [Locally Powered Cisco IP Phones, page 11-3](#)
- [Inline-Powered Cisco IP Phones, page 11-3](#)

Locally Powered Cisco IP Phones

There are two varieties of local power:

- From a power supply connected to the Cisco IP Phone
- From a power supply through a patch panel over the twisted-pair Ethernet cable to the Cisco IP Phone

When a locally powered Cisco IP Phone is present on a switching module port, the switching module cannot detect its presence. The supervisor engine discovers the Cisco IP Phone through CDPv2 messaging with the Cisco IP Phone.

If a locally powered Cisco IP Phone loses local power and the mode is set to **auto**, the switching module discovers the Cisco IP Phone and informs the supervisor engine, which then supplies inline power to the Cisco IP Phone.

Inline-Powered Cisco IP Phones

Inline power is from switching modules that support an inline power daughtercard. Inline power is sent over the twisted-pair Ethernet cable to the Cisco IP Phone.

**Note**

For information about switching modules that support inline power, refer to the *Release Notes for Cisco IOS Release 12.1E on the Catalyst 6000 and Cisco 7600 Supervisor Engine and MSFC* publication at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/12_1e/ol_2310.htm

When a switching module port detects an unpowered Cisco IP Phone, the switching module reports to the supervisor engine that an unpowered Cisco IP Phone is present and on which module and port. If the port is configured in **auto** mode, the supervisor engine determines if there is enough system power available to power up the Cisco IP Phone. If there is sufficient power available, the supervisor engine removes the default-allocated power required by a Cisco IP Phone from the total available system power and sends a message to the switching module instructing it to provide power to the port. If there is not enough available power for the Cisco IP Phone, the supervisor engine sends a message to the switching module indicating that power is denied to the port.

Cisco IP Phones may have different power requirements. The supervisor engine initially allocates the configured default of 7 W (167 mA at 42V) to the Cisco IP Phone. When the correct amount of power is determined from the CDPv2 messaging with the Cisco IP Phone, the supervisor engine reduces or increases the allocated power.

For example, the default allocated power is 7 W. A Cisco IP Phone requiring 6.3W is plugged into a port. The supervisor engine allocates 7 W for the Cisco IP Phone and powers it up. Once the Cisco IP Phone is operational, it sends a CDPv2 message with the actual power requirement to the supervisor engine. The supervisor engine then decreases the allocated power to the required amount.

When you power off the Cisco IP Phone through the CLI or SNMP or remove it, the supervisor engine sends a message to the switching module to turn off the power on the port. That power is then returned to the available system power.

**Caution**

When a Cisco IP Phone cable is plugged into a port and the power is turned on, the supervisor engine has a 4-second timeout waiting for the link to go up on the line. During those 4 seconds, if the Cisco IP Phone cable is unplugged and a network device is plugged in, the network device could be damaged. We recommend that you wait at least 10 seconds between unplugging a network device and plugging in another network device.

Default Cisco IP Phone Support Configuration

Cisco IP Phone support is disabled by default.

When the voice VLAN feature is enabled, all untagged traffic is sent with the default CoS priority of the port.

The CoS is not trusted for 802.1P or 802.1Q tagged traffic.

Cisco IP Phone Support Configuration Guidelines and Restrictions

When configuring IP phone supports, follow these guidelines and restrictions:

- You must enable the Cisco Discovery Protocol version 2 (CDPv2) on the Catalyst 6500 series switch port connected to the Cisco IP Phone to send configuration information to the Cisco IP Phone.
- You can configure a voice VLAN only on a Layer 2 LAN port.
- You can configure the ports on WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules to trust received Layer 2 CoS values (QoS port architecture 1p1q0t/1p3q1t). The WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules cannot supply power to Cisco IP Phones.

- You cannot configure 10/100 Mbps ports with QoS port architecture 1p4t/2q2t to trust received Layer 2 CoS values. Configure policies to trust the Layer 3 IP precedence value on switching modules with QoS port architecture 1p4t/2q2t.
- The following conditions indicate that the Cisco IP Phone and a device attached to the Cisco IP Phone are in the same VLAN and must be in the same IP subnet:
 - If they both use 802.1p or untagged frames
 - If the Cisco IP Phone uses 802.1p frames and the device uses untagged frames
 - If the Cisco IP Phone uses untagged frames and the device uses 802.1p frames
 - If the Cisco IP Phone uses 802.1Q frames and the voice VLAN is the same as the access VLAN
- The Cisco IP Phone and a device attached to the Cisco IP Phone cannot communicate if they are in the same VLAN and subnet but use different frame types, because traffic between devices in the same subnet is not routed (routing would eliminate the frame type difference).
- You cannot use Cisco IOS software commands to configure the frame type used by traffic sent from a device attached to the access port on the Cisco IP Phone.
- If you enable port security on a port configured with a voice VLAN and if there is a PC connected to the Cisco IP Phone, set the maximum allowed secure addresses on the port to at least 3.
- You cannot configure static secure MAC addresses in the voice VLAN.
- Ports configured with a voice VLAN can be secure ports (refer to [Chapter 26, “Configuring Port Security”](#)).
- In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

Configuring Cisco IP Phone Support

These sections describe how to configure Cisco IP Phone support:

- [Configuring Voice Traffic Support, page 11-5](#)
- [Configuring Data Traffic Support, page 11-7](#)
- [Configuring Inline Power Support, page 11-8](#)



Note

Voice VLANs are referred to as *auxiliary VLANs* in the Catalyst software publications.

Configuring Voice Traffic Support

To configure the way in which the Cisco IP Phone transmits voice traffic, perform this task:

	Command	Purpose
Step 1	Router(config)# interface fastethernet <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# switchport voice vlan { <i>voice_vlan_ID</i> dot1p none untagged }	Configures the way in which the Cisco IP Phone transmits voice traffic.
	Router(config-if)# no switchport voice vlan	Clears the configuration.

	Command	Purpose
Step 3	Router(config)# end	Exits configuration mode.
Step 4	Router# show interfaces fastethernet slot/port switchport Router# show running-config interface fastethernet slot/port	Verifies the configuration.

When configuring the way in which the Cisco IP Phone transmits voice traffic, note the following syntax information:

- Enter a voice VLAN ID to send CDPv2 packets that configure the Cisco IP Phone to transmit voice traffic in 802.1Q frames, tagged with the voice VLAN ID and a Layer 2 CoS value (the default is 5). Valid VLAN IDs are from 1 to 4094. The switch puts the 802.1Q voice traffic into the voice VLAN.
- Enter the **dot1p** keyword to send CDPv2 packets that configure the Cisco IP Phone to transmit voice traffic in 802.1p frames, tagged with VLAN ID 0 and a Layer 2 CoS value (the default is 5 for voice traffic and 3 for voice control traffic). The switch puts the 802.1p voice traffic into the access VLAN.
- Enter the **untagged** keyword to send CDPv2 packets that configure the Cisco IP Phone to transmit untagged voice traffic. The switch puts the untagged voice traffic into the access VLAN.
- Enter the **none** keyword to allow the Cisco IP Phone to use its own configuration and transmit untagged voice traffic. The switch puts the untagged voice traffic into the access VLAN.
- In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5).
- Refer to [Chapter 31, “Configuring PFC QoS,”](#) for information about how to configure QoS.
- Refer to the [“Configuring a LAN Interface as a Layer 2 Access Port”](#) section on page 7-14 for information about how to configure the port as a Layer 2 access port and configure the access VLAN.

This example shows how to configure Fast Ethernet port 5/1 to send CDPv2 packets that tell the Cisco IP Phone to use VLAN 101 as the voice VLAN:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport voice vlan 101
Router(config-if)# exit
```

This example shows how to verify the configuration of Fast Ethernet port 5/1:

```
Router# show interfaces fastethernet 5/1 switchport
Name: Fa5/1
Switchport: Enabled
Administrative Mode: access
Operational Mode: access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: off
Access Mode VLAN: 100
Voice VLAN: 101
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none
Administrative private-vlan mapping: 900 ((Inactive)) 901 ((Inactive))
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

Configuring Data Traffic Support

To configure the way in which the Cisco IP Phone transmits data traffic, perform this task:

	Command	Purpose
Step 1	Router(config)# interface fastethernet <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# mls qos trust extend [cos <i>cos_value</i>]	Configures the way in which the Cisco IP Phone transmits data traffic.
	Router(config-if)# no mls qos trust extend	Clears the configuration.
Step 3	Router(config)# end	Exits configuration mode.
Step 4	Router# show interfaces fastethernet <i>slot/port</i> switchport Router# show running-config interface fastethernet <i>slot/port</i>	Verifies the configuration.

When configuring the way in which the Cisco IP Phone transmits data traffic, note the following syntax information:

- To send CDPv2 packets that configure the Cisco IP Phone to trust tagged traffic received from a device connected to the access port on the Cisco IP Phone, do not enter the **cos** keyword and CoS value.
- To send CDPv2 packets that configure the Cisco IP Phone to mark tagged ingress traffic received from a device connected to the access port on the Cisco IP Phone, enter the **cos** keyword and CoS value (valid values are 0 through 7).
- You cannot use Cisco IOS software commands to configure whether or not traffic sent from a device attached to the access port on the Cisco IP Phone is tagged.

This example shows how to configure Fast Ethernet port 5/1 to send CDPv2 packets that tell the Cisco IP Phone to configure its access port as untrusted and to mark all tagged traffic received from a device connected to the access port on the Cisco IP Phone with CoS 3:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# mls qos trust extend cos 3
```

This example shows how to configure Fast Ethernet port 5/1 to send CDPv2 packets that tell the Cisco IP Phone to configure its access port as trusted:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# mls qos trust extend
```

This example shows how to verify the configuration on Fast Ethernet port 5/1:

```
Router# show queueing interface fastethernet 5/1 | include Extend
Extend trust state: trusted
```

Configuring Inline Power Support

To configure inline power support, perform this task:

	Command	Purpose
Step 1	Router(config)# interface fastethernet <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# power inline { auto never }	Configures inline power support.
	Router(config-if)# no power inline	Clears the configuration.
Step 3	Router(config)# end	Exits configuration mode.
Step 4	Router# show power inline [fastethernet <i>slot/port</i>]	Verifies the configuration.

When configuring inline power support, note the following syntax information:

- To configure auto-detection of a Cisco IP Phone, enter the **auto** keyword.
- To disable auto-detection of a Cisco IP Phone, enter the **never** keyword.

This example shows how to disable inline power on Fast Ethernet port 5/1:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# power inline never
```

This example shows how to enable inline power on Fast Ethernet port 5/1:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# power inline auto
```

This example shows how to verify the inline power configuration on Fast Ethernet port 5/1:

```
Router# show power inline fastethernet 5/1
Interface  Admin    Oper    Power      Device
              (Watts)
-----  -
Fa5/1      auto  on      6.3      cisco phone device
```




Configuring Layer 3 Interfaces

This chapter contains information about how to configure Layer 3 interfaces on the Catalyst 6500 series switches, which supplements the information and procedures in the Release 12.1 publications at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

This chapter consists of these sections:

- [Configuring IP Routing and Addresses, page 12-2](#)
- [Configuring IPX Routing and Network Numbers, page 12-6](#)
- [Configuring AppleTalk Routing, Cable Ranges, and Zones, page 12-7](#)
- [Configuring Other Protocols on Layer 3 Interfaces, page 12-8](#)

**Note**

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication and the Release 12.1 publications at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>
- Release 12.1(13)E and later releases support configuration of 4,096 Layer 3 VLAN interfaces.
 - We recommend that you configure a combined total of no more than 2,000 Layer 3 VLAN interfaces and Layer 3 ports on an MSFC2 with either Supervisor Engine 1 or Supervisor Engine 2.
 - We recommend that you configure a combined total of no more than 1,000 Layer 3 VLAN interfaces and Layer 3 ports on an MSFC.
- With releases earlier than Release 12.1(13)E, an MSFC2 with either Supervisor Engine 1 or Supervisor Engine 2 supports a combined maximum of 1,000 Layer 3 VLAN interfaces and Layer 3 ports.
- With releases earlier than Release 12.1(13)E, an MSFC with Supervisor Engine 1 supports a maximum of 256 Layer 3 VLAN interfaces.
- To support VLAN interfaces, create and configure VLANs and assign VLAN membership to Layer 2 LAN ports. For more information, see [Chapter 9, “Configuring VLANs”](#) and [Chapter 8, “Configuring VTP.”](#)
- Catalyst 6500 series switches support Layer 3 trunks only on the 4-port Gigabit Ethernet WAN module (OSM-4GE-WAN and OSM-2+4GE-WAN+). You cannot configure subinterfaces or use the **encapsulation** keyword on LAN ports. Catalyst 6500 series switches support Layer 2 trunks and Layer 3 VLAN interfaces, which provide equivalent capabilities for LAN ports. See [Chapter 7, “Configuring LAN Ports for Layer 2 Switching”](#) and the “[Configuring IP Routing and Addresses](#)” section on page 12-2.
- With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Configuring IP Routing and Addresses

For complete information and procedures, refer to these publications:

- *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/index.htm
- *Cisco IOS IP and IP Routing Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/index.htm

For information about the **maximum paths** command in Release 12.1 E, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

The Policy Feature Card 2 (PFC2) and any Distributed Feature Cards (DFCs) provide hardware support for policy-based routing (PBR) for route-map sequences that use the **match ip address** and **set ip next-hop** keywords.

With Release 12.1(11b)E and later, the PFC2 and any DFCs provide hardware support for the **ip default next-hop** PBR keywords.

The Multilayer Switch Feature Card 2 (MSFC2) provides processing in software for route-map sequences that use the **match length** and **set interface** keywords.

To configure PBR, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1, “Classification,” “Configuring Policy-Based Routing,” at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt1/qcdpbr.htm

To configure IP routing and an IP address on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# ip routing	Enables IP routing. (Required only if IP routing is disabled.)
Step 2	Router(config)# router <i>ip_routing_protocol</i>	Specifies an IP routing protocol.
Step 3	Router(config-router)# <i>ip_routing_protocol_commands</i>	Configures the IP routing protocol.
Step 4	Router(config-router)# exit	Exits IP routing protocol configuration mode.
Step 5	Router(config)# interface { <i>vlan vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>port_channel_number</i> }	Selects an interface to configure.
Step 6	Router(config-if)# ip address <i>ip_address subnet_mask</i>	Configures the IP address and IP subnet.
Step 7	Router(config-if)# no shutdown	Enables the interface.
Step 8	Router(config-if)# end	Exits configuration mode.
Step 9	Router# show interfaces [{ <i>vlan vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>port_channel_number</i> }] Router# show ip interfaces [{ <i>vlan vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>port_channel_number</i> }] Router# show running-config interfaces [{ <i>vlan vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>port_channel_number</i> }]	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

This example shows how to enable IP Routing Information Protocol (RIP) routing:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip routing
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# end
Router#
```

This example shows how to configure an IP address on Fast Ethernet port 5/4:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/4
Router(config-if)# ip address 172.20.52.106 255.255.255.248
Router(config-if)# no shutdown
Router(config-if)# end
Router#
```

This example uses the **show interfaces** command to display the interface IP address configuration and status of Fast Ethernet port 5/4:

```

Router# show interfaces fastethernet 5/4
FastEthernet5/4 is up, line protocol is up
  Hardware is Cat6K 100Mb Ethernet, address is 0050.f0ac.3058 (bia 0050.f0ac.3058)
  Internet address is 172.20.52.106/29
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    7 packets input, 871 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    8 packets output, 1658 bytes, 0 underruns
    0 output errors, 0 collisions, 4 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
Router#

```

This example uses the **show ip interface** command to display the detailed configuration and status of Fast Ethernet port 5/4:

```

Router# show ip interface fastethernet 5/4
FastEthernet5/4 is up, line protocol is up
  Internet address is 172.20.52.106/29
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP Fast switching turbo vector
  IP Normal CEF switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled

```

```
WCCP Redirect exclude is disabled
BGP Policy Mapping is disabled
IP multicast multilayer switching is disabled
IP mls switching is enabled
Router#
```

This example uses the **show running-config** command to display the interface IP address configuration of Fast Ethernet port 5/4:

```
Router# show running-config interfaces fastethernet 5/4
Building configuration...
```

```
Current configuration:
!
interface FastEthernet5/4
  description "Router port"
  ip address 172.20.52.106 255.255.255.248
  no ip directed-broadcast
!
```

Configuring IPX Routing and Network Numbers

For complete information and procedures, refer to these publications:

- *Cisco IOS AppleTalk and Novell IPX Configuration Guide*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/atipx_c/index.htm
- *Cisco IOS AppleTalk and Novell IPX Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/atipx_r/index.htm

To configure routing for Internetwork Packet Exchange (IPX) and configure IPX on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# ipx routing	Enables IPX routing.
Step 2	Router(config)# router ipx_routing_protocol	Specifies an IP routing protocol. This step might include other commands, such as specifying the networks to route with the network command.
Step 3	Router(config)# interface {vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 4	Router(config-if)# ipx network [network unnumbered] encapsulation encapsulation_type	Configures the IPX network number. This enables IPX routing on the interface. When you enable IPX routing on the interface, you can also specify an encapsulation type.
Step 5	Router(config-if)# no shutdown	Enables the interface.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}] Router# show ipx interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}] Router# show running-config interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}]	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

This example shows how to enable IPX routing and assign an IPX network address to interface VLAN 100:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ipx routing
Router(config)# ipx router rip
Router(config-ipx-router)# network all
Router(config-ipx-router)# interface vlan 100
Router(config-if)# ipx network 100 encapsulation snap
Router(config-if)# no shutdown
Router(config-if)# end
Router# copy running-config startup-config
```

Configuring AppleTalk Routing, Cable Ranges, and Zones

For complete information and procedures, refer to these publications:

- *Cisco IOS AppleTalk and Novell IPX Configuration Guide*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/atipx_c/index.htm
- *Cisco IOS AppleTalk and Novell IPX Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/atipx_r/index.htm

To configure routing for AppleTalk, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# appletalk routing	Enables AppleTalk routing.
Step 2	Router(config)# interface {vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 3	Router(config-if)# appletalk cable-range cable_range	Assigns a cable range to the interface.
Step 4	Router(config-if)# appletalk zone zone_name	Assigns a zone name to the interface.
Step 5	Router(config-if)# no shutdown	Enables the interface.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}] Router# show appletalk interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}] Router# show running-config interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}]	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

This example shows how to enable AppleTalk routing and assign an AppleTalk cable-range and zone name to interface VLAN 100:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# appletalk routing
Router(config)# interface vlan 100
Router(config-if)# appletalk cable-range 100-100
Router(config-if)# appletalk zone Engineering
Router(config-if)# no shutdown
Router(config-if)# end
Router# copy running-config startup-config
```

Configuring Other Protocols on Layer 3 Interfaces

Refer to these publications for information about configuring other protocols on Layer 3 interfaces:

- *Cisco IOS Apollo Domain, VINES, DECnet, ISO CLNS, and XNS Configuration Guide*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/apollo_c/index.htm
- *Cisco IOS Apollo Domain, VINES, DECnet, ISO CLNS, and XNS Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/apollo_r/index.htm



Configuring EtherChannels

This chapter describes how to configure EtherChannels on the Catalyst 6500 series switch Layer 2 or Layer 3 LAN ports.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.
 - The commands in the following sections can be used on all LAN ports in Catalyst 6500 series switches, including the ports on the supervisor engine and a redundant supervisor engine.
 - Release 12.1(13)E and later releases support the IEEE 802.3ad Link Aggregation Control Protocol (LACP).
 - The WS-X6548-GE-TX and WS-X6548V-GE-TX fabric-enabled switching modules do not support more than 1 Gbps of traffic per EtherChannel, except when the switch is operating in truncated mode.
 - The WS-X6148-GE-TX and WS-X6148V-GE-TX switching modules do not support more than 1 Gbps of traffic per EtherChannel.
-

This chapter consists of these sections:

- [Understanding How EtherChannels Work, page 13-1](#)
- [EtherChannel Feature Configuration Guidelines and Restrictions, page 13-5](#)
- [Configuring EtherChannels, page 13-6](#)

Understanding How EtherChannels Work

These sections describe how EtherChannels work:

- [EtherChannel Feature Overview, page 13-2](#)
- [Understanding How EtherChannels Are Configured, page 13-2](#)
- [Understanding Port Channel Interfaces, page 13-5](#)
- [Understanding Load Balancing, page 13-5](#)

EtherChannel Feature Overview

An EtherChannel bundles individual Ethernet links into a single logical link that provides the aggregate bandwidth of up to eight physical links.

A Catalyst 6500 series switch supports a maximum of 64 EtherChannels (256 with Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6500 series switch. All LAN ports in each EtherChannel must be the same speed and must all be configured as either Layer 2 or Layer 3 LAN ports.

**Note**

The network device to which a Catalyst 6500 series switch is connected may impose its own limits on the number of ports in an EtherChannel.

If a segment within an EtherChannel fails, traffic previously carried over the failed link switches to the remaining segments within the EtherChannel. When a failure occurs, the EtherChannel feature sends a trap that identifies the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one segment in an EtherChannel are blocked from returning on any other segment of the EtherChannel.

Understanding How EtherChannels Are Configured

These sections describe how EtherChannels are configured:

- [EtherChannel Configuration Overview, page 13-2](#)
- [Understanding Manual EtherChannel Configuration, page 13-3](#)
- [Understanding PAgP EtherChannel Configuration, page 13-3](#)
- [Understanding IEEE 802.3ad LACP EtherChannel Configuration, page 13-3](#)

EtherChannel Configuration Overview

You can configure EtherChannels manually or you can use the Port Aggregation Control Protocol (PAgP) or, with Release 12.1(13)E and later, the Link Aggregation Control Protocol (LACP) to form EtherChannels. The EtherChannel protocols allow ports with similar characteristics to form an EtherChannel through dynamic negotiation with connected network devices. PAgP is a Cisco-proprietary protocol and LACP is defined in IEEE 802.3ad.

PAgP and LACP do not interoperate with each other. Ports configured to use PAgP cannot form EtherChannels with ports configured to use LACP. Ports configured to use LACP cannot form EtherChannels with ports configured to use PAgP.

[Table 13-1](#) lists the user-configurable EtherChannel modes.

Table 13-1 EtherChannel Modes

Mode	Description
on	Mode that forces the LAN port to channel unconditionally. In the on mode, a usable EtherChannel exists only when a LAN port group in the on mode is connected to another LAN port group in the on mode. Because ports configured in the on mode do not negotiate, there is no negotiation traffic between the ports. You cannot configure the on mode with an EtherChannel protocol.
auto	PAgP mode that places a LAN port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not initiate PAgP negotiation. (Default)
desirable	PAgP mode that places a LAN port into an active negotiating state, in which the port initiates negotiations with other LAN ports by sending PAgP packets.
passive	LACP mode that places a port into a passive negotiating state, in which the port responds to LACP packets it receives but does not initiate LACP negotiation. (Default)
active	LACP mode that places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending LACP packets.

Understanding Manual EtherChannel Configuration

Manually configured EtherChannel ports do not exchange EtherChannel protocol packets. A manually configured EtherChannel forms only when you enter configure all ports in the EtherChannel compatibly.

Understanding PAgP EtherChannel Configuration

PAgP supports the automatic creation of EtherChannels by exchanging PAgP packets between LAN ports. PAgP packets are exchanged only between ports in **auto** and **desirable** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once PAgP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **auto** and **desirable** modes allow PAgP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different PAgP modes if the modes are compatible. For example:

- A LAN port in **desirable** mode can form an EtherChannel successfully with another LAN port that is in **desirable** mode.
- A LAN port in **desirable** mode can form an EtherChannel with another LAN port in **auto** mode.
- A LAN port in **auto** mode cannot form an EtherChannel with another LAN port that is also in **auto** mode, because neither port will initiate negotiation.

Understanding IEEE 802.3ad LACP EtherChannel Configuration

Release 12.1(13)E and later releases support IEEE 802.3ad LACP EtherChannels. LACP supports the automatic creation of EtherChannels by exchanging LACP packets between LAN ports. LACP packets are exchanged only between ports in **passive** and **active** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **passive** and **active** modes allow LACP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different LACP modes as long as the modes are compatible. For example:

- A LAN port in **active** mode can form an EtherChannel successfully with another LAN port that is in **active** mode.
- A LAN port in **active** mode can form an EtherChannel with another LAN port in **passive** mode.
- A LAN port in **passive** mode cannot form an EtherChannel with another LAN port that is also in **passive** mode, because neither port will initiate negotiation.

LACP uses the following parameters:

- LACP system priority—You must configure an LACP system priority on each switch running LACP. The system priority can be configured automatically or through the CLI (see the [“Configuring the LACP System Priority and System ID” section on page 13-10](#)). LACP uses the system priority with the switch MAC address to form the system ID and also during negotiation with other systems.



Note The LACP system ID is the combination of the LACP system priority value and the MAC address of the switch.

- LACP port priority—You must configure an LACP port priority on each port configured to use LACP. The port priority can be configured automatically or through the CLI (see the [“Configuring Channel Groups” section on page 13-8](#)). LACP uses the port priority with the port number to form the port identifier. LACP uses the port priority to decide which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.
- LACP administrative key—LACP automatically configures an administrative key value equal to the channel group identification number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port’s ability to aggregate with other ports is determined by these factors:
 - Port physical characteristics, such as data rate, duplex capability, and point-to-point or shared medium
 - Configuration restrictions that you establish

On ports configured to use LACP, LACP tries to configure the maximum number of compatible ports in an EtherChannel, up to the maximum allowed by the hardware (eight ports). If LACP cannot aggregate all the ports that are compatible (for example, the remote system might have more restrictive hardware limitations), then all the ports that cannot be actively included in the channel are put in hot standby state and are used only if one of the channeled ports fails. You can configure an additional 8 standby ports (total of 16 ports associated with the EtherChannel).

Understanding Port Channel Interfaces

Each EtherChannel has a numbered port channel interface. Release 12.1(5)E and later releases support a maximum of 64 port channel interfaces, numbered from 1 to 256.

**Note**

Releases 12.1(4)E1, 12.1(3a)E4, and 12.1(3a)E3 support a maximum of 64 port channel interfaces, numbered from 1 to 64. Releases 12.1(2)E and earlier support a maximum of 256 port channel interfaces, numbered from 1 to 256.

The configuration that you apply to the port channel interface affects all LAN ports assigned to the port channel interface.

After you configure an EtherChannel, the configuration that you apply to the port channel interface affects the EtherChannel; the configuration that you apply to the LAN ports affects only the LAN port where you apply the configuration. To change the parameters of all ports in an EtherChannel, apply the configuration commands to the port channel interface, for example, Spanning Tree Protocol (STP) commands or commands to configure a Layer 2 EtherChannel as a trunk.

Understanding Load Balancing

An EtherChannel balances the traffic load across the links in an EtherChannel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel.

EtherChannel load balancing can use MAC addresses or IP addresses. With a PFC2, EtherChannel load balancing can also use Layer 4 port numbers. EtherChannel load balancing can use either source or destination or both source and destination addresses or ports. The selected mode applies to all EtherChannels configured on the switch.

Use the option that provides the balance criteria with the greatest variety in your configuration. For example, if the traffic on an EtherChannel is going only to a single MAC address and you use the destination MAC address as the basis of EtherChannel load balancing, the EtherChannel always chooses the same link in the EtherChannel; using source addresses or IP addresses might result in better load balancing.

EtherChannel Feature Configuration Guidelines and Restrictions

When EtherChannel interfaces are configured improperly, they are disabled automatically to avoid network loops and other problems. To avoid configuration problems, observe these guidelines and restrictions:

- All Ethernet LAN ports on all modules, including those on a redundant supervisor engine, support EtherChannels (maximum of eight LAN ports) with no requirement that the LAN ports be physically contiguous or on the same module.
- Configure all LAN ports in an EtherChannel to use the same EtherChannel protocol; you cannot run two EtherChannel protocols in one EtherChannel.
- Configure all LAN ports in an EtherChannel to operate at the same speed and in the same duplex mode.
- LACP does not support half-duplex. Half-duplex ports in an LACP EtherChannel are put in the suspended state.

- Enable all LAN ports in an EtherChannel. If you shut down a LAN port in an EtherChannel, it is treated as a link failure and its traffic is transferred to one of the remaining ports in the EtherChannel.
- An EtherChannel will not form if one of the LAN ports is a Switched Port Analyzer (SPAN) destination port.
- For Layer 3 EtherChannels, assign Layer 3 addresses to the port channel logical interface, not to the LAN ports in the channel.
- For Layer 2 EtherChannels:
 - Assign all LAN ports in the EtherChannel to the same VLAN or configure them as trunks.
 - If you configure an EtherChannel from trunking LAN ports, verify that the trunking mode is the same on all the trunks. LAN ports in an EtherChannel with different trunk modes can operate unpredictably.
 - An EtherChannel supports the same allowed range of VLANs on all the LAN ports in a trunking Layer 2 EtherChannel. If the allowed range of VLANs is not the same, the LAN ports do not form an EtherChannel.
 - LAN ports with different STP port path costs can form an EtherChannel as long they are compatibly configured with each other. If you set different STP port path costs, the LAN ports are not incompatible for the formation of an EtherChannel.
 - An EtherChannel will not form if protocol filtering is set differently on the LAN ports.
- After you configure an EtherChannel, the configuration that you apply to the port channel interface affects the EtherChannel. The configuration that you apply to the LAN ports affects only the LAN port where you apply the configuration.
- With Release 12.1(12c)E1 and later releases, when QoS is enabled, enter the **no mls qos channel-consistency** port-channel interface command to support EtherChannels that have ports with and without strict-priority queues.

Configuring EtherChannels

These sections describe how to configure EtherChannels:

- [Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels, page 13-7](#)
- [Configuring Channel Groups, page 13-8](#)
- [Configuring EtherChannel Load Balancing, page 13-11](#)



Note

- Make sure that the LAN ports are configured correctly (see the “[EtherChannel Feature Configuration Guidelines and Restrictions](#)” section on page 13-5).
- With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode commands by entering the **do** keyword before the EXEC mode command.

Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels



Note

- When configuring Layer 2 EtherChannels, you cannot put Layer 2 LAN ports into manually created port channel logical interfaces. If you are configuring a Layer 2 EtherChannel, do not perform the procedures in this section (see the “[Configuring Channel Groups](#)” section on page 13-8).
- When configuring Layer 3 EtherChannels, you must manually create the port channel logical interface as described in this section, and then put the Layer 3 LAN ports into the channel group (see the “[Configuring Channel Groups](#)” section on page 13-8).
- To move an IP address from a Layer 3 LAN port to an EtherChannel, you must delete the IP address from the Layer 3 LAN port before configuring it on the port channel logical interface.

To create a port channel interface for a Layer 3 EtherChannel, perform this task:

	Command	Purpose
Step 1	Router(config)# interface port-channel <i>number</i>	Creates the port channel interface.
	Router(config)# no interface port-channel <i>number</i>	Deletes the port channel interface.
Step 2	Router(config-if)# ip address <i>ip_address mask</i>	Assigns an IP address and subnet mask to the EtherChannel.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show running-config interface port-channel <i>number</i>	Verifies the configuration.

When creating the port channel interface, the *group* number can be one of the following:

- Release 12.1(5)E and later—1 through 256, up to a maximum of 64 port channel interfaces
- Releases 12.1(4)E1, 12.1(3a)E4, and 12.1(3a)E3—1 through 64
- Release 12.1(2)E and earlier—1 through 256

This example shows how to create port channel interface 1:

```
Router# configure terminal
Router(config)# interface port-channel 1
Router(config-if)# ip address 172.32.52.10 255.255.255.0
Router(config-if)# end
```

This example shows how to verify the configuration of port channel interface 1:

```
Router# show running-config interface port-channel 1
Building configuration...

Current configuration:
!
interface Port-channell
 ip address 172.32.52.10 255.255.255.0
 no ip directed-broadcast
end
Router#
```

Configuring Channel Groups



Note

- When configuring Layer 3 EtherChannels, you must manually create the port channel logical interface first (see the “[Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels](#)” section on page 13-7), and then put the Layer 3 LAN ports into the channel group as described in this section.
- When configuring Layer 2 EtherChannels, configure the LAN ports with the **channel-group** command as described in this section, which automatically creates the port channel logical interface. You cannot put Layer 2 LAN ports into a manually created port channel interface.
- For Cisco IOS to create port channel interfaces for Layer 2 EtherChannels, the Layer 2 LAN ports must be connected and functioning.

To configure channel groups, perform this task for each LAN port:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects a LAN port to configure.
Step 2	Router(config-if)# no ip address	Ensures that there is no IP address assigned to the LAN port.
Step 3	Router(config-if)# channel-protocol (lACP pagp) Router(config-if)# no channel-protocol	(Optional) On the selected LAN port, restricts the channel-group command to the EtherChannel protocol configured with the channel-protocol command. Removes the restriction.
Step 4	Router(config-if)# channel-group <i>number</i> mode { active auto desirable on passive } Router(config-if)# no channel-group	Configures the LAN port in a port channel and specifies the mode (see Table 13-1 on page 13-3). PAGP supports only the auto and desirable modes. LACP supports only the active and passive modes. Removes the LAN port from the channel group.
Step 5	Router(config-if)# lACP port-priority <i>priority_value</i> Router(config-if)# no lACP port-priority	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768. Reverts to the default.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show running-config interface <i>type</i> ¹ <i>slot/port</i> Router# show interfaces <i>type</i> ¹ <i>slot/port</i> etherchannel	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure Fast Ethernet ports 5/6 and 5/7 into port channel 2 with PAGP mode **desirable**:

```
Router# configure terminal
Router(config)# interface range fastethernet 5/6 -7
Router(config-if)# channel-group 2 mode desirable
Router(config-if)# end
```


**Note**

See the “Configuring a Range of Interfaces” section on page 6-4 for information about the **range** keyword.

This example shows how to verify the configuration of port channel interface 2:

```
Router# show running-config interface port-channel 2
Building configuration...

Current configuration:
!
interface Port-channel2
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
end
Router#
```

This example shows how to verify the configuration of Fast Ethernet port 5/6:

```
Router# show running-config interface fastethernet 5/6
Building configuration...

Current configuration:
!
interface FastEthernet5/6
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
 channel-group 2 mode desirable
end
Router# show interfaces fastethernet 5/6 etherchannel
Port state      = Down Not-in-Bndl
Channel group = 12          Mode = Desirable-S1          Gcchange = 0
Port-channel   = null      GC      = 0x00000000          Pseudo port-channel = Po1
2
Port index     = 0          Load = 0x00          Protocol = PAgP

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.         P - Device learns on physical port.
        d - PAgP is down.

Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.     I - Interface timer is running.

Local information:

Port      Flags State  Timers  Hello  Partner  PAgP  Learning  Group
Fa5/2    d    U1/S1  1s      1s     0        128   Any       0

Age of the port in the current state: 04d:18h:57m:19s
```

This example shows how to verify the configuration of port channel interface 2 after the LAN ports have been configured:

```
Router# show etherchannel 12 port-channel
      Port-channels in the group:
      -----

Port-channel: Po12
-----

Age of the Port-channel   = 04d:18h:58m:50s
Logical slot/port        = 14/1           Number of ports = 0
GC                       = 0x00000000    HotStandBy port = null
Port state               = Port-channel Ag-Not-Inuse
Protocol                 = PAgP

Router#
```

Configuring the LACP System Priority and System ID

The LACP system ID is the combination of the LACP system priority value and the MAC address of the switch.

To configure the LACP system priority and system ID, perform this task:

	Command	Purpose
Step 1	Router(config)# lacp system-priority <i>priority_value</i>	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
	Router(config)# no lacp system-priority	Reverts to the default.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show lacp sys-id	Verifies the configuration.

This example shows how to configure the LACP system priority:

```
Router# configure terminal
Router(config)# lacp system-priority 23456
Router(config)# end
Router(config)#
```

This example shows how to verify the configuration:

```
Router# show lacp sys-id
23456,0050.3e8d.6400
Router#
```

The system priority is displayed first, followed by the MAC address of the switch.

Configuring EtherChannel Load Balancing

To configure EtherChannel load balancing, perform this task:

	Command	Purpose
Step 1	Router(config)# port-channel load-balance { src-mac dst-mac src-dst-mac src-ip dst-ip src-dst-ip src-port dst-port src-dst-port }	Configures EtherChannel load balancing.
	Router(config)# no port-channel load-balance	Reverts to default EtherChannel load balancing.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show etherchannel load-balance	Verifies the configuration.

The load-balancing keywords indicate the following information:

- With a PFC2:
 - **src-port**—Source Layer 4 port
 - **dst-port**—Destination Layer 4 port
 - **src-dst-port**—Source and destination Layer 4 port
- With a PFC or PFC2:
 - **src-ip**—Source IP addresses
 - **dst-ip**—Destination IP addresses
 - **src-dst-ip**—Source and destination IP addresses
 - **src-mac**—Source MAC addresses
 - **dst-mac**—Destination MAC addresses
 - **src-dst-mac**—Source and destination MAC addresses

This example shows how to configure EtherChannel to use source and destination IP addresses:

```
Router# configure terminal
Router(config)# port-channel load-balance src-dst-ip
Router(config)# end
Router(config)#
```

This example shows how to verify the configuration:

```
Router# show etherchannel load-balance
Source XOR Destination IP address
Router#
```




Configuring IEEE 802.1Q Tunneling and Layer 2 Protocol Tunneling

With Release 12.1(13)E and later, the Catalyst 6500 series switches support IEEE 802.1Q tunneling and Layer 2 protocol tunneling. This chapter describes how to configure IEEE 802.1Q tunneling and Layer 2 protocol tunneling on the Catalyst 6500 series switches.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.
- The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX switching modules do not support IEEE 802.1Q tunneling or Layer 2 protocol tunneling.

This chapter consists of these sections:

- [Understanding How 802.1Q Tunneling Works, page 14-1](#)
- [802.1Q Tunneling Configuration Guidelines and Restrictions, page 14-4](#)
- [Configuring 802.1Q Tunneling, page 14-5](#)
- [Understanding How Layer 2 Protocol Tunneling Works, page 14-7](#)
- [Configuring Support for Layer 2 Protocol Tunneling, page 14-8](#)

Understanding How 802.1Q Tunneling Works

802.1Q tunneling enables service providers to use a single VLAN to support customers who have multiple VLANs, while preserving customer VLAN IDs and keeping traffic in different customer VLANs segregated.

A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that you dedicate to tunneling, which then becomes a tunnel VLAN. To keep customer traffic segregated, each customer requires a separate tunnel VLAN, but that one tunnel VLAN supports all of the customer's VLANs.

802.1Q tunneling is not restricted to point-to-point tunnel configurations. Any tunnel port in a tunnel VLAN is a tunnel entry and exit point. An 802.1Q tunnel can have as many tunnel ports as are needed to connect customer switches.

The customer switches are trunk connected, but with 802.1Q tunneling, the service provider switches only use one service provider VLAN to carry all the customer VLANs, instead of directly carrying all the customer VLANs.

With 802.1Q tunneling, tagged customer traffic comes from an 802.1Q trunk port on a customer device and enters the service-provider edge switch through a tunnel port. The link between the 802.1Q trunk port on a customer device and the tunnel port is called an asymmetrical link because one end is configured as an 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port to an access VLAN ID unique to each customer. See [Figure 14-1 on page 14-2](#) and [Figure 14-2 on page 14-3](#).

Figure 14-1 IEEE 802.1Q Tunnel Ports in a Service-Provider Network

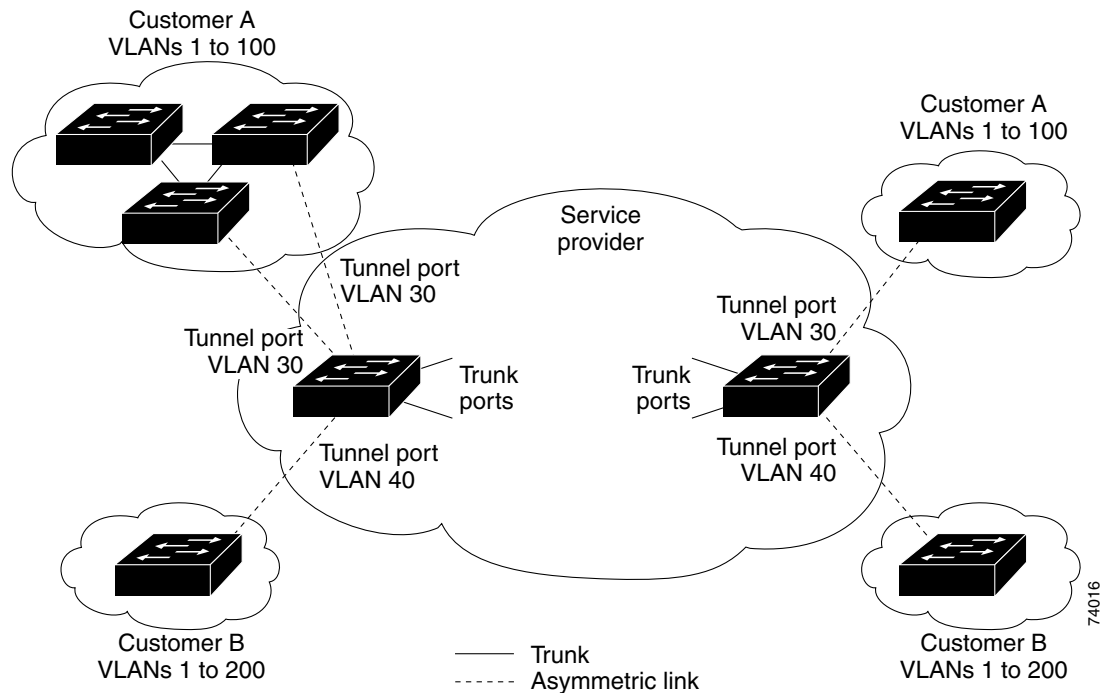
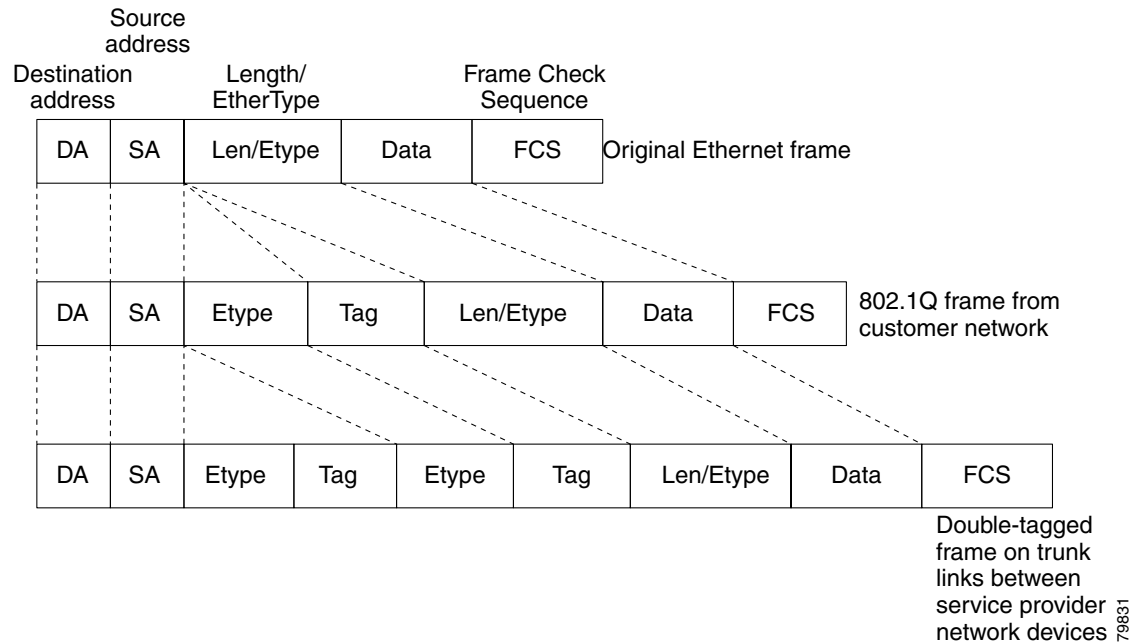


Figure 14-2 Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames

When a tunnel port receives tagged customer traffic from an 802.1Q trunk port, it does not strip the received 802.1Q tag from the frame header; instead, the tunnel port leaves the 802.1Q tag intact, adds a 2-byte Ethertype field (0x8100) followed by a 2-byte field containing the priority (CoS) and the VLAN. The received customer traffic is then put into the VLAN to which the tunnel port is assigned. This Ethertype 0x8100 traffic, with the received 802.1Q tag intact, is called tunnel traffic.

A VLAN carrying tunnel traffic is an 802.1Q tunnel. The tunnel ports in the VLAN are the tunnel's ingress and egress points.

The tunnel ports do not have to be on the same network device. The tunnel can cross other network links and other network devices before reaching the egress tunnel port. A tunnel can have as many tunnel ports as required to support the customer devices that need to communicate through the tunnel.

An egress tunnel port strips the 2-byte Ethertype field (0x8100) and the 2-byte length field and transmits the traffic with the 802.1Q tag still intact to an 802.1Q trunk port on a customer device. The 802.1Q trunk port on the customer device strips the 802.1Q tag and puts the traffic into the appropriate customer VLAN.

**Note**

Tunnel traffic carries a second 802.1Q tag only when it is on a trunk link between service-provider network devices, with the outer tag containing the service-provider-assigned VLAN ID and the inner tag containing the customer-assigned VLAN IDs.

802.1Q Tunneling Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring 802.1Q tunneling in your network:

Restrictions

- Because tunnel traffic has the added ethertype and length field and retains the 802.1Q tag within the switch, the following restrictions exist:
 - The Layer 3 packet within the Layer 2 frame cannot be identified in tunnel traffic.
 - Layer 3 and higher parameters cannot be identified in tunnel traffic (for example, Layer 3 destination and source addresses).
 - Because the Layer 3 addresses cannot be identified within the packet, tunnel traffic cannot be routed.
 - The switch can provide only MAC-layer filtering for tunnel traffic (VLAN IDs and source and destination MAC addresses).
 - The switch can provide only MAC-layer access control and QoS for tunnel traffic.
 - QoS cannot detect the received CoS value in the 802.1Q 2-byte Tag Control Information field.
- Tunnel ports learn customer MAC addresses.
- On an asymmetrical link, the Cisco Discovery Protocol (CDP) reports a native VLAN mismatch if the VLAN of the tunnel port does not match the native VLAN of the 802.1Q trunk. The 802.1Q tunnel feature does not require that the VLANs match. Ignore the messages if your configuration requires nonmatching VLANs.
- Asymmetrical links do not support the Dynamic Trunking Protocol (DTP), because only one port on the link is a trunk. Configure the 802.1Q trunk port on an asymmetrical link to trunk unconditionally.
- Jumbo frames can be tunneled as long as the jumbo frame length combined with the 802.1Q tag does not exceed the maximum frame size.
- The 802.1Q tunneling feature cannot be configured on ports configured to support private VLANs
- VLAN Trunk Protocol (VTP) does not work between the following devices:
 - Devices connected by an asymmetrical link
 - Devices communicating through a tunnel



Note VTP works between tunneled devices if Layer 2 protocol tunneling is enabled. See the [“Configuring Support for Layer 2 Protocol Tunneling”](#) section on page 14-8 for configuration details.

Guidelines

- Use asymmetrical links to put traffic into a tunnel or to remove traffic from a tunnel.
- Configure tunnel ports only to form an asymmetrical link.
- Dedicate one VLAN for each tunnel.
- Assign only tunnel ports to VLANs used for tunneling.

- Trunks require no special configuration to carry tunnel VLANs.
- We recommend that you use ISL trunks to carry tunnel traffic between devices that do not have tunnel ports. Because of the 802.1Q native VLAN feature, using 802.1Q trunks requires that you be very careful when you configure tunneling: a mistake might direct tunnel traffic to a non-tunnel port.
- Ensure that the native VLAN of the 802.1Q trunk port in an asymmetrical link carries no traffic. Because traffic in the native VLAN is untagged, it cannot be tunneled correctly. Alternatively, you can enter the global **vlan dot1q tag native** command to tag native VLAN egress traffic and drop untagged native VLAN ingress traffic.
- The following Layer 2 protocols work between devices connected by an asymmetrical link:
 - CDP
 - UniDirectional Link Detection (UDLD)
 - Port Aggregation Protocol (PAgP)
 - Link Aggregation Control Protocol (LACP)
- With Release 12.1(13)E and later releases, PortFast BPDU filtering is enabled automatically on tunnel ports. With releases earlier than Release 12.1(13)E, you can manually enable PortFast BPDU filtering on tunnel ports (see the “[Enabling PortFast BPDU Filtering](#)” section on page 16-10).
- With Release 12.1(13)E and later releases, CDP is automatically disabled on tunnel ports. With releases earlier than Release 12.1(13)E, you can manually disable CDP when you enable 802.1Q tunneling (see the “[Enabling CDP on a Port](#)” section on page 30-2).
- To configure an EtherChannel as an asymmetrical link, all ports in the EtherChannel must have the same tunneling configuration. Because the Layer 3 packet within the Layer 2 frame cannot be identified, you must configure the EtherChannel to use MAC-address-based frame distribution.
- Because all the BPDUs are being dropped, spanning tree PortFast can be enabled on Layer 2 protocol tunnel ports as follows:

```
Router(config-if)# spanning-tree portfast trunk
```

- If the service provider does not want the customer to see its switches, CDP should be disabled on the 802.1Q tunnel port as follows:

```
Router(config-if)# no cdp enable
```

Configuring 802.1Q Tunneling

These sections describe 802.1Q tunneling configuration:

- [Preconfiguration Tasks, page 14-6](#)
- [Configuring 802.1Q Tunnel Ports, page 14-6](#)
- [Configuring the Switch to Tag Native VLAN Traffic, page 14-7](#)



Caution

Ensure that only the appropriate tunnel ports are in any VLAN used for tunneling and that one VLAN is used for each tunnel. Incorrect assignment of tunnel ports to VLANs can forward traffic inappropriately.

Preconfiguration Tasks

Before you can configure Layer 2 protocol tunneling, you must perform these tasks:

- Step 1** On all the service provider edge switches, PortFast BPDU filtering must be enabled on the 802.1Q tunnel ports as follows:

```
Router(config-if)# spanning-tree bpdupfilter enable
Router(config-if)# spanning-tree portfast
```



Note With Release 12.1(13)E and later releases, PortFast BPDU filtering is enabled automatically on tunnel ports. With releases earlier than Release 12.1(13)E, you must manually enable PortFast BPDU filtering on tunnel ports.

- Step 2** At least one VLAN must be available for Native VLAN tagging (**vlan dot1q tag native** option). If you use all the available VLANs and then try to enable the **vlan dot1q tag native** option, the option will not be enabled.

- Step 3** On all the service provider core switches, tag native VLAN egress traffic and drop untagged native VLAN ingress traffic by entering the following command:

```
Router(config)# vlan dot1q tag native
```

- Step 4** On all the customer switches, *either* enable or disable the global **vlan dot1q tag native** option.



Note If this option is enabled on one switch and disabled on another switch, all traffic is dropped; all customer switches must have this option configured the same on each switch.

Configuring 802.1Q Tunnel Ports

To configure 802.1Q tunneling on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching: <ul style="list-style-type: none"> You must enter the switchport command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional switchport commands with keywords. Required only if you have not entered the switchport command already for the interface.
Step 3	Router(config-if)# switchport mode dot1qtunnel	Configures the Layer 2 port as a tunnel port.
	Router(config-if)# no switchport mode dot1qtunnel	Clears the tunnel port configuration.

	Command	Purpose
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show dot1q-tunnel [{ <i>interface type interface-number</i> }]	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure tunneling on port 4/1 and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 4/1
Router(config-if)# switchport mode dot1qtunnel
Router(config-if)# end
Router# show dot1q-tunnel interface
```

Configuring the Switch to Tag Native VLAN Traffic

The `vlan dot1q tag native` command is a global command that configures the switch to tag native VLAN traffic, and admit only 802.1Q tagged frames on 802.1Q trunks, dropping any untagged traffic, including untagged traffic in the native VLAN.

To configure the switch to tag traffic in the native VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan dot1q tag native	Configures the switch to tag native VLAN traffic.
	Router(config)# no vlan dot1q tag native	Clears the configuration.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show vlan dot1q tag native	Verifies the configuration.

This example shows how to configure the switch to tag native VLAN traffic and verify the configuration:

```
Router# configure terminal
Router(config)# vlan dot1q tag native
Router(config)# end
Router# show vlan dot1q tag native
```

Understanding How Layer 2 Protocol Tunneling Works

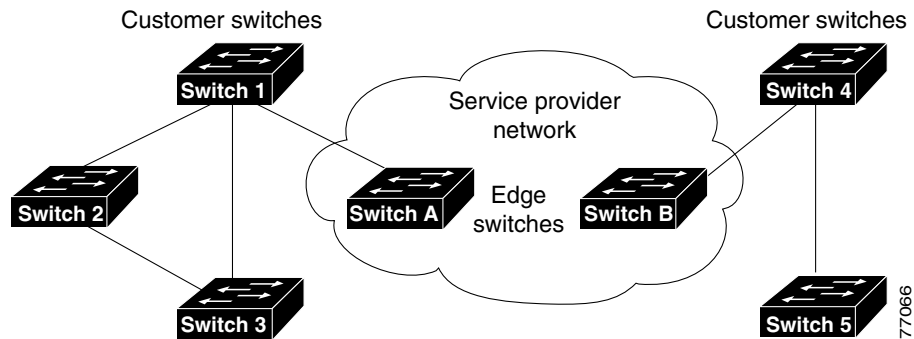
Layer 2 protocol tunneling allows Layer 2 protocol data units (PDUs) (CDP, STP, and VTP) to be tunneled through a network. This section uses the following terminology:

- Edge switch—The switch connected to the customer switch and placed on the boundary of the service provider network (see [Figure 14-3](#)).
- Layer 2 protocol tunnel port—A port on the edge switch on which a specific tunneled protocol can be encapsulated or deencapsulated. The Layer 2 protocol tunnel port is configured through CLI commands.
- Tunneled PDU—A CDP, STP, or VTP PDU.

Without Layer 2 protocol tunneling, tunnel ports drop STP and VTP packets and process CDP packets. This handling of the PDUs creates different spanning tree domains (different spanning tree roots) for the customer switches. For example, STP for a VLAN on switch 1 (see [Figure 14-3](#)) builds a spanning tree

topology on switches 1, 2, and 3 without considering convergence parameters based on switches 4 and 5. To provide a single spanning tree domain for the customer, a generic scheme to tunnel BPDUs was created for control protocol PDUs (CDP, STP, and VTP). This process is referred to as Generic Bridge PDU Tunneling (GBPT).

Figure 14-3 Layer 2 Protocol Tunneling Network Configuration



GBPT provides a scalable approach to PDU tunneling by software encapsulating the PDUs in the ingress edge switches and then multicasting them in hardware. All switches inside the service provider network treat these encapsulated frames as data packets and forward them to the other end. The egress edge switch listens for these special encapsulated frames and deencapsulates them; they are then forwarded out of the tunnel.

The encapsulation involves rewriting the destination media access control (MAC) address in the PDU. An ingress edge switch rewrites the destination MAC address of the PDUs received on a Layer 2 tunnel port with the Cisco proprietary multicast address (01-00-0c-cd-cd-d0). The PDU is then flooded to the native VLAN of the Layer 2 tunnel port. If you enable Layer 2 protocol tunneling on a port, PDUs of an enabled protocol are not sent out. If you disable Layer 2 protocol tunneling on a port, the disabled protocols behave the same way they were behaving before Layer 2 protocol tunneling was disabled on the port.

Configuring Support for Layer 2 Protocol Tunneling



Note

Encapsulated PDUs received by an 802.1Q tunnel port are transmitted from other tunnel ports in the same VLAN on the switch.

To configure Layer 2 protocol tunneling on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching: <ul style="list-style-type: none"> You must enter the switchport command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional switchport commands with keywords. Required only if you have not entered the switchport command already for the interface.
Step 3	Router(config-if)# l2protocol-tunnel [cdp drop-threshold [<i>packets</i>] shutdown-threshold [<i>packets</i>] stp vtp]	Configures the Layer 2 port as a Layer 2 protocol tunnel port for the protocol(s) specified.
	Router(config-if)# no l2protocol-tunnel [cdp drop-threshold shutdown-threshold stp vtp]	Clears the configuration.
Step 4	Router(config)# end	Exits configuration mode.
Step 5	Router# show l2protocol-tunnel [interface <i>type</i> ¹ <i>slot/port</i> summary]	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When you configure a Layer 2 port as a Layer 2 protocol tunnel port, note the following syntax information:

- Optionally, you may specify a drop-threshold for the port. The drop-threshold value, from 1 to 4096, determines the number of packets to be processed for that protocol on that interface in one second. When the drop threshold is exceeded, PDUs for the specified protocol are dropped for the remainder of the 1-second period. If a shutdown threshold is not specified, the value is 0 (shutdown threshold disabled).
- Optionally, you may specify a shutdown-threshold for the port. The drop-threshold value, from 1 to 4096, determines the number of packets to be processed for that protocol on that interface in one second. When the shutdown threshold is exceeded, the port is put in errdisable state. If a shutdown threshold is not specified, the value is 0 (shutdown threshold disabled).



Note

A new keyword, **l2ptguard**, has been added to the following commands:

- errdisable detect cause**
- errdisable recovery cause**

Refer to the *Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide—Release 12.1 E* publication for more information.

This example shows how to configure Layer 2 protocol tunneling and shutdown thresholds on port 5/1 for CDP, STP, and VTP, and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport
Router(config-if)# l2protocol-tunnel shutdown-threshold cdp 10
Router(config-if)# l2protocol-tunnel shutdown-threshold stp 10
Router(config-if)# l2protocol-tunnel shutdown-threshold vtp 10
Router(config-if)# end
Router# show l2protocol-tunnel summary
Port   Protocol          Threshold
              (cos/cdp/stp/vtp)
-----
Fa5/1  cdp stp vtp      0/10 /10 /10      down trunk
Router#
```

This example shows how to display counter information for port 5/1:

```
Router# show l2protocol-tunnel interface fastethernet 5/1
Port   Protocol          Threshold          Counters
              (cos/cdp/stp/vtp)  (cdp/stp/vtp/decap)
-----
Router#
```

This example shows how to clear the Layer 2 protocol tunneling configuration from port 5/1:

```
Router(config-if)# no l2protocol-tunnel shutdown-threshold cdp 10
Router(config-if)# no l2protocol-tunnel shutdown-threshold stp 10
Router(config-if)# no l2protocol-tunnel shutdown-threshold vtp 10
Router(config-if)# no l2protocol-tunnel cdp
Router(config-if)# no l2protocol-tunnel stp
Router(config-if)# no l2protocol-tunnel vtp
Router(config-if)# end
Router# show l2protocol-tunnel summary
Port   Protocol          Threshold
              (cos/cdp/stp/vtp)
-----
Router#
```

This example shows how to clear Layer 2 protocol tunneling port counters:

```
Router# clear l2protocol-tunnel counters
Router#
```



Configuring STP and IEEE 802.1s MST

This chapter describes how to configure the Spanning Tree Protocol (STP) and the IEEE 802.1s Multiple Spanning Tree (MST) protocol on Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How STP Works, page 15-2](#)
- [Understanding How IEEE 802.1w RSTP Works, page 15-13](#)
- [Understanding How IEEE 802.1s MST Works, page 15-14](#)
- [Default STP Configuration, page 15-21](#)
- [STP and MST Configuration Guidelines, page 15-21](#)
- [Configuring STP, page 15-22](#)
- [Configuring IEEE 802.1s MST, page 15-34](#)



Note

-
- For information on configuring the PortFast, UplinkFast, and BackboneFast STP enhancements, see [Chapter 16, “Configuring Optional STP Features.”](#)
 - Release 12.1(13)E and later releases support IEEE 802.1s MST and IEEE 802.1w, rapid reconfiguration of spanning tree.
-

Understanding How STP Works

These sections describe how STP works:

- [STP Overview, page 15-2](#)
- [Understanding the Bridge ID, page 15-3](#)
- [Understanding Bridge Protocol Data Units, page 15-4](#)
- [Election of the Root Bridge, page 15-4](#)
- [STP Protocol Timers, page 15-5](#)
- [Creating the Spanning Tree Topology, page 15-5](#)
- [STP Port States, page 15-6](#)
- [STP and IEEE 802.1Q Trunks, page 15-12](#)

STP Overview

STP is a Layer 2 link management protocol that provides path redundancy while preventing undesirable loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. STP operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

Catalyst 6500 series switches use STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single instance of STP runs on each configured VLAN (provided you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched Layer 2 network. Layer 2 LAN ports send and receive STP frames at regular intervals. Network devices do not forward these frames, but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and network devices might learn end station MAC addresses on multiple Layer 2 LAN ports. These conditions result in an unstable network.

STP defines a tree with a root bridge and a loop-free path from the root to all network devices in the Layer 2 network. STP forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the standby path.

When two Layer 2 LAN ports on a network device are part of a loop, the STP port priority and port path cost setting determine which port is put in the forwarding state and which port is put in the blocking state. The STP port priority value represents the location of a port in the network topology and how well located it is to pass traffic. The STP port path cost value represents media speed.

Understanding the Bridge ID

Each VLAN on each network device has a unique 64-bit bridge ID consisting of a bridge priority value, an extended system ID, and an STP MAC address allocation.

This section contains these topics:

- [Bridge Priority Value, page 15-3](#)
- [Extended System ID, page 15-3](#)
- [STP MAC Address Allocation, page 15-3](#)

Bridge Priority Value

With Release 12.1(8a)E and later releases, the bridge priority is a 4-bit value when the extended system ID is enabled (see [Table 15-2 on page 15-3](#)). With earlier releases, the bridge priority is a 16-bit value (see [Table 15-1 on page 15-3](#)). See the “[Configuring the Bridge Priority of a VLAN](#)” section on [page 15-30](#).

Extended System ID

Release 12.1(8a)E and later releases support a 12-bit extended system ID field as part of the bridge ID (see [Table 15-2 on page 15-3](#)). Chassis that support only 64 MAC addresses always use the 12-bit extended system ID. On chassis that support 1024 MAC addresses, you can enable use of the extended system ID. STP uses the VLAN ID as the extended system ID. See the “[Enabling the Extended System ID](#)” section on [page 15-24](#).

Table 15-1 Bridge Priority Value with the Extended System ID Disabled

Bridge Priority Value															
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Table 15-2 Bridge Priority Value and Extended System ID with the Extended System ID Enabled

Bridge Priority Value				Extended System ID (Set Equal to the VLAN ID)											
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

STP MAC Address Allocation

Catalyst 6500 series switch chassis have either 64 or 1024 MAC addresses available to support software features such as STP. To view the MAC address range on your chassis, enter the **show catalyst6000 chassis-mac-address** command.

Release 12.1(8a)E and later releases support chassis with 64 or 1024 MAC addresses. For chassis with 64 MAC addresses, STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

Earlier releases support chassis with 1024 MAC addresses. With earlier releases, STP uses one MAC address per VLAN to make the bridge ID unique for each VLAN.

If you have a network device in your network with MAC address reduction enabled, you should also enable MAC address reduction on all other Layer-2 connected network devices to avoid undesirable root bridge election and spanning tree topology issues.

When MAC address reduction is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. With MAC address reduction enabled, a switch bridge ID (used by the spanning-tree algorithm to determine the identity of the root bridge, the lowest being preferred) can only be specified as a multiple of 4096. Only the following values are possible: 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440.

If another bridge in the same spanning-tree domain does not run the MAC address reduction feature, it could win root bridge ownership because of the finer granularity in the selection of its bridge ID.

Understanding Bridge Protocol Data Units

Bridge protocol data units (BPDUs) are transmitted in one direction from the root bridge. Each network device sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the network device that the transmitting network device believes to be the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- Message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timers

When a network device transmits a BPDU frame, all network devices connected to the LAN on which the frame is transmitted receive the BPDU. When a network device receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU, and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One network device is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each network device based on the path cost.
- A designated bridge for each LAN segment is selected. This is the network device closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. This is the port providing the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.

Election of the Root Bridge

For each VLAN, the network device with the highest bridge ID (the lowest numerical ID value) is elected as the root bridge. If all network devices are configured with the default priority (32768), the network device with the lowest MAC address in the VLAN becomes the root bridge. The bridge priority value occupies the most significant bits of the bridge ID.

When you change the bridge priority value, you change the probability that the switch will be elected as the root bridge. Configuring a higher value increases the probability; a lower value decreases the probability.

The STP root bridge is the logical center of the spanning tree topology in a Layer 2 network. All paths that are not needed to reach the root bridge from anywhere in the Layer 2 network are placed in STP blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the Layer 2 network, to elect the root port leading to the root bridge, and to determine the designated port for each Layer 2 segment.

STP Protocol Timers

Table 15-3 describes the STP protocol timers that affect STP performance.

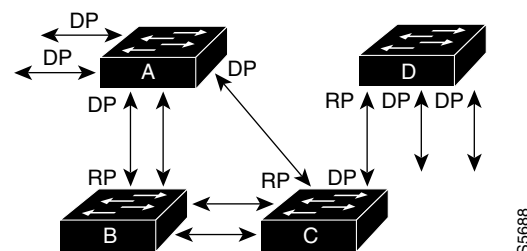
Table 15-3 STP Protocol Timers

Variable	Description
Hello timer	Determines how often the network device broadcasts hello messages to other network devices.
Forward delay timer	Determines how long each of the listening and learning states last before the port begins forwarding.
Maximum age timer	Determines the amount of time protocol information received on an port is stored by the network device.

Creating the Spanning Tree Topology

In Figure 15-1, Switch A is elected as the root bridge because the bridge priority of all the network devices is set to the default (32768) and Switch A has the lowest MAC address. However, due to traffic patterns, number of forwarding ports, or link types, Switch A might not be the ideal root bridge. By increasing the priority (lowering the numerical value) of the ideal network device so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal network device as the root.

Figure 15-1 Spanning Tree Topology



RP = Root Port
DP = Designated Port

When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the STP port priority on the fiber-optic port to a higher priority (lower numerical value) than the root port, the fiber-optic port becomes the new root port.

STP Port States

These sections describe the STP port states:

- [STP Port State Overview, page 15-6](#)
- [Blocking State, page 15-8](#)
- [Listening State, page 15-9](#)
- [Learning State, page 15-10](#)
- [Forwarding State, page 15-11](#)
- [Disabled State, page 15-12](#)

STP Port State Overview

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a Layer 2 LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames. They must allow the frame lifetime to expire for frames that have been forwarded using the old topology.

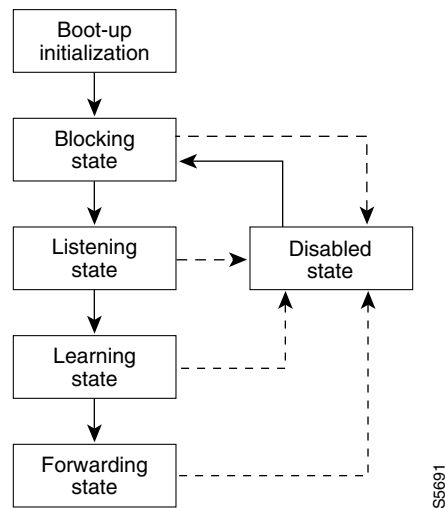
Each Layer 2 LAN port on a Catalyst 6500 series switch using STP exists in one of the following five states:

- **Blocking**—The Layer 2 LAN port does not participate in frame forwarding.
- **Listening**—First transitional state after the blocking state when STP determines that the Layer 2 LAN port should participate in frame forwarding.
- **Learning**—The Layer 2 LAN port prepares to participate in frame forwarding.
- **Forwarding**—The Layer 2 LAN port forwards frames.
- **Disabled**—The Layer 2 LAN port does not participate in STP and is not forwarding frames.

A Layer 2 LAN port moves through these five states as follows:

- From initialization to blocking
- From blocking to listening or to disabled
- From listening to learning or to disabled
- From learning to forwarding or to disabled
- From forwarding to disabled

Figure 15-2 illustrates how a Layer 2 LAN port moves through the five states.

Figure 15-2 STP Layer 2 LAN Interface States

When you enable STP, every port in the Catalyst 6500 series switch, VLAN, and network goes through the blocking state and the transitory states of listening and learning at power up. If properly configured, each Layer 2 LAN port stabilizes to the forwarding or blocking state.

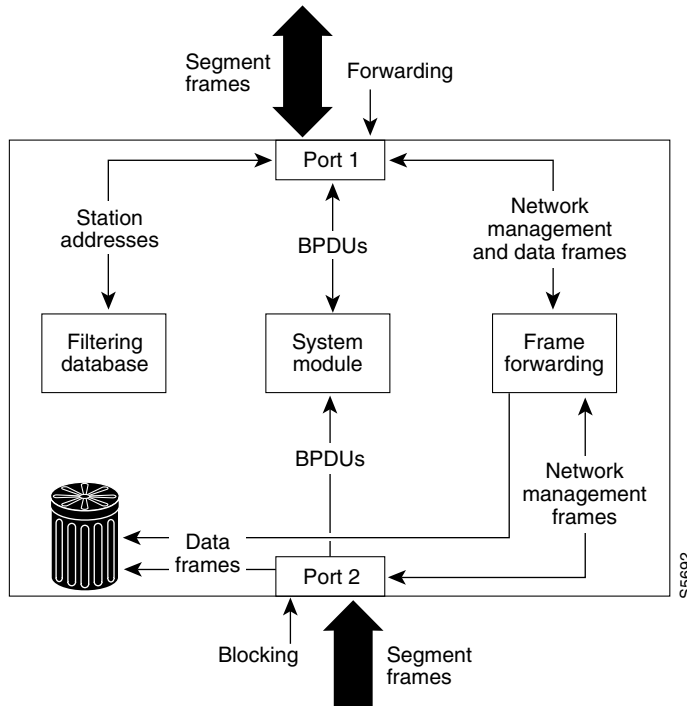
When the STP algorithm places a Layer 2 LAN port in the forwarding state, the following process occurs:

1. The Layer 2 LAN port is put into the listening state while it waits for protocol information that suggests it should go to the blocking state.
2. The Layer 2 LAN port waits for the forward delay timer to expire, moves the Layer 2 LAN port to the learning state, and resets the forward delay timer.
3. In the learning state, the Layer 2 LAN port continues to block frame forwarding as it learns end station location information for the forwarding database.
4. The Layer 2 LAN port waits for the forward delay timer to expire and then moves the Layer 2 LAN port to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A Layer 2 LAN port in the blocking state does not participate in frame forwarding, as shown in Figure 15-3. After initialization, a BPDU is sent out to each Layer 2 LAN port. A network device initially assumes it is the root until it exchanges BPDUs with other network devices. This exchange establishes which network device in the network is the root or root bridge. If only one network device is in the network, no exchange occurs, the forward delay timer expires, and the ports move to the listening state. A port always enters the blocking state following initialization.

Figure 15-3 Interface 2 in Blocking State



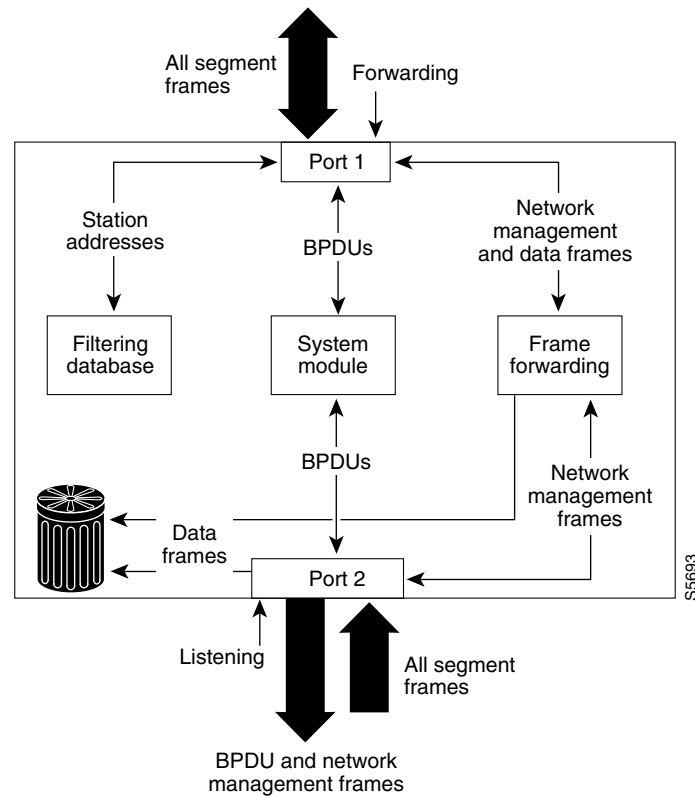
A Layer 2 LAN port in the blocking state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate end station location into its address database. (There is no learning on a blocking Layer 2 LAN port, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Does not transmit BPDUs received from the system module.
- Receives and responds to network management messages.

Listening State

The listening state is the first transitional state a Layer 2 LAN port enters after the blocking state. The Layer 2 LAN port enters this state when STP determines that the Layer 2 LAN port should participate in frame forwarding. [Figure 15-4](#) shows a Layer 2 LAN port in the listening state.

Figure 15-4 Interface 2 in Listening State



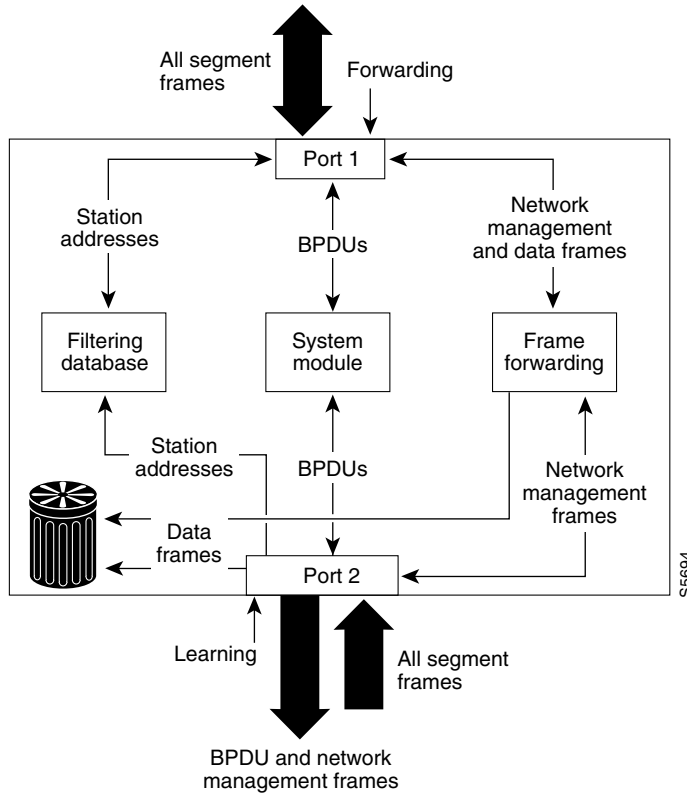
A Layer 2 LAN port in the listening state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another LAN port for forwarding.
- Does not incorporate end station location into its address database. (There is no learning at this point, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Learning State

A Layer 2 LAN port in the learning state prepares to participate in frame forwarding. The Layer 2 LAN port enters the learning state from the listening state. Figure 15-5 shows a Layer 2 LAN port in the learning state.

Figure 15-5 Interface 2 in Learning State



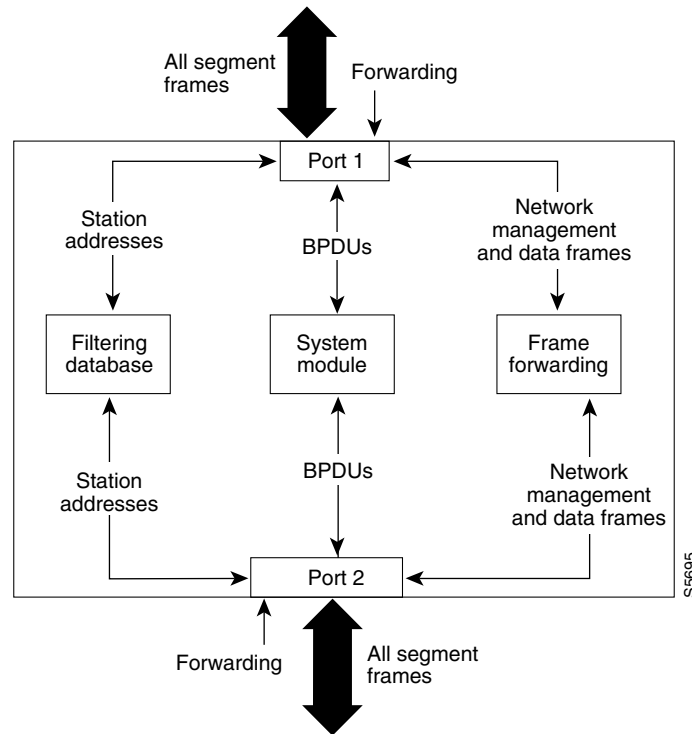
A Layer 2 LAN port in the learning state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Incorporates end station location into its address database.
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Forwarding State

A Layer 2 LAN port in the forwarding state forwards frames, as shown in Figure 15-6. The Layer 2 LAN port enters the forwarding state from the learning state.

Figure 15-6 Interface 2 in Forwarding State



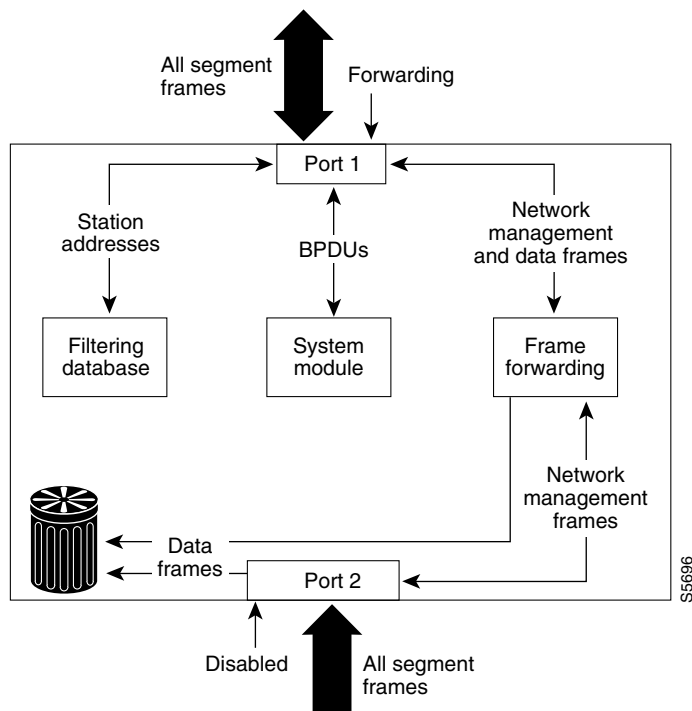
A Layer 2 LAN port in the forwarding state performs as follows:

- Forwards frames received from the attached segment.
- Forwards frames switched from another port for forwarding.
- Incorporates end station location information into its address database.
- Receives BPDUs and directs them to the system module.
- Processes BPDUs received from the system module.
- Receives and responds to network management messages.

Disabled State

A Layer 2 LAN port in the disabled state does not participate in frame forwarding or STP, as shown in Figure 15-7. A Layer 2 LAN port in the disabled state is virtually nonoperational.

Figure 15-7 Interface 2 in Disabled State



A disabled Layer 2 LAN port performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate end station location into its address database. (There is no learning, so there is no address database update.)
- Does not receive BPDUs.
- Does not receive BPDUs for transmission from the system module.

STP and IEEE 802.1Q Trunks

802.1Q trunks impose some limitations on the STP strategy for a network. In a network of Cisco network devices connected through 802.1Q trunks, the network devices maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q network devices maintain only one instance of STP for all VLANs allowed on the trunks.

When you connect a Cisco network device to a non-Cisco device through an 802.1Q trunk, the Cisco network device combines the STP instance of the 802.1Q VLAN of the trunk with the STP instance of the non-Cisco 802.1Q network device. However, all per-VLAN STP information is maintained by Cisco network devices separated by a cloud of non-Cisco 802.1Q network devices. The non-Cisco 802.1Q cloud separating the Cisco network devices is treated as a single trunk link between the network devices.

For more information on 802.1Q trunks, see [Chapter 7, “Configuring LAN Ports for Layer 2 Switching.”](#)

Understanding How IEEE 802.1w RSTP Works

**Note**

In Cisco IOS release 12.1(11)EX and later releases, RSTP is implemented as part of Multiple Spanning Tree Protocol (MSTP). In Cisco IOS release 12.1(13)E and later releases, RSTP is also available as a standalone protocol in Rapid-Per-VLAN-Spanning Tree (Rapid-PVST) mode. In this mode, the switch runs an RSTP instance on each VLAN, which follows the usual PVST+ approach.

These sections describe Rapid Spanning Tree Protocol (RSTP):

- [IEEE 802.1w RSTP Overview, page 15-13](#)
- [RSTP Port Roles, page 15-13](#)
- [RSTP Port States, page 15-14](#)
- [Rapid-PVST, page 15-14](#)

IEEE 802.1w RSTP Overview

RSTP significantly reduces the time to reconfigure the active topology of the network when changes occur to the physical topology or its configuration parameters. RSTP selects one switch as the root of a spanning tree-connected active topology and assigns port roles to individual ports of the switch, depending on whether that port is part of the active topology.

RSTP provides rapid connectivity following the failure of a switch, switch port, or a LAN. A new root port and the designated port on the other side of the bridge transition to forwarding using an explicit handshake between them. RSTP allows switch port configuration so that the ports can transition to forwarding directly when the switch reinitializes.

RSTP as specified in 802.1w supersedes STP specified in 802.1D, but remains compatible with STP.

RSTP provides backward compatibility with 802.1D bridges as follows:

- RSTP selectively sends 802.1D-configured BPDUs and topology change notification (TCN) BPDUs on a per-port basis.
- When a port initializes, the migration-delay timer starts and RSTP BPDUs are transmitted. While the migration-delay timer is active, the bridge processes all BPDUs received on that port.
- If the bridge receives an 802.1D BPDU after a port's migration-delay timer expires, the bridge assumes it is connected to an 802.1D bridge and starts using only 802.1D BPDUs.
- When RSTP uses 802.1D BPDUs on a port and receives an RSTP BPDU after the migration-delay expires, RSTP restarts the migration-delay timer and begins using RSTP BPDUs on that port.

RSTP Port Roles

RSTP uses the following definitions for port roles:

- **Root**—A forwarding port elected for the spanning tree topology.
- **Designated**—A forwarding port elected for every switched LAN segment.
- **Alternate**—An alternate path to the root bridge to that provided by the current root port.

- Backup—A backup for the path provided by a designated port toward the leaves of the spanning tree. Backup ports can exist only where two ports are connected together in a loopback by a point-to-point link or bridge with two or more connections to a shared LAN segment.
- Disabled—A port that has no role within the operation of spanning tree.

Port roles are assigned as follows:

- A root port or designated port role includes the port in the active topology.
- An alternate port or backup port role excludes the port from the active topology.

RSTP Port States

The port state controls the forwarding and learning processes and provides the values of discarding, learning, and forwarding. Table 15-4 provides a comparison between STP port states and RSTP port states.

Table 15-4 Comparison Between STP and RSTP Port States

Operational Status	STP Port State	RSTP Port State	Port Included in Active Topology
Enabled	Blocking ¹	Discarding ²	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Discarding	No

1. IEEE 802.1D port state designation.

2. IEEE 802.1w port state designation. Discarding is the same as blocking in RSTP and MST.

In a stable topology, RSTP ensures that every root port and designated port transition to forwarding, and ensures that all alternate ports and backup ports are always in the discarding state.

Rapid-PVST

Rapid-PVST uses the existing configuration for PVST+; however, Rapid-PVST uses RSTP to provide faster convergence. Independent VLANs run their own RSTP instance.

Dynamic entries are flushed immediately on a per-port basis upon receiving a topology change.

UplinkFast and BackboneFast configurations are ignored in Rapid-PVST mode; both features are included in RSTP.

Understanding How IEEE 802.1s MST Works



Note

In Cisco IOS release 12.1(11)EX and later releases, RSTP is implemented as part of Multiple Spanning Tree Protocol (MSTP). In Cisco IOS release 12.1(13)E and later releases, RSTP is also available as a standalone protocol in Rapid-Per-VLAN-Spanning Tree (Rapid-PVST) mode. In this mode, the switch runs an RSTP instance on each VLAN, which follows the usual PVST+ approach.

These sections describe Multiple Spanning Tree (MST):

- [IEEE 802.1s MST Overview, page 15-15](#)
- [MST-to-PVST Interoperability, page 15-16](#)
- [Common Spanning Tree, page 15-18](#)
- [MST Instances, page 15-18](#)
- [MST Configuration Parameters, page 15-18](#)
- [MST Regions, page 15-19](#)
- [Message Age and Hop Count, page 15-20](#)
- [Default STP Configuration, page 15-21](#)

IEEE 802.1s MST Overview

Releases 12.1(11b)EX and later releases support MST. MST in this release is based on the draft version of the IEEE standard. 802.1s for MST is an amendment to 802.1Q. MST extends the IEEE 802.1w rapid spanning tree (RST) algorithm to multiple spanning trees. This extension provides both rapid convergence and load balancing in a VLAN environment. MST converges faster than PVST+. MST is backward compatible with 802.1D STP, 802.1w (rapid spanning tree protocol [RSTP]), and the Cisco PVST+ architecture.

MST allows you to build multiple spanning trees over trunks. You can group and associate VLANs to spanning tree instances. Each instance can have a topology independent of other spanning tree instances. This new architecture provides multiple forwarding paths for data traffic and enables load balancing. Network fault tolerance is improved because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

In large networks, you can more easily administer the network and use redundant paths by locating different VLAN and spanning tree instance assignments in different parts of the network. A spanning tree instance can exist only on bridges that have compatible VLAN instance assignments. You must configure a set of bridges with the same MST configuration information, which allows them to participate in a specific set of spanning tree instances. Interconnected bridges that have the same MST configuration are referred to as an *MST region*.

MST uses the modified RSTP version called the Multiple Spanning Tree Protocol (MSTP). The MST feature has these characteristics:

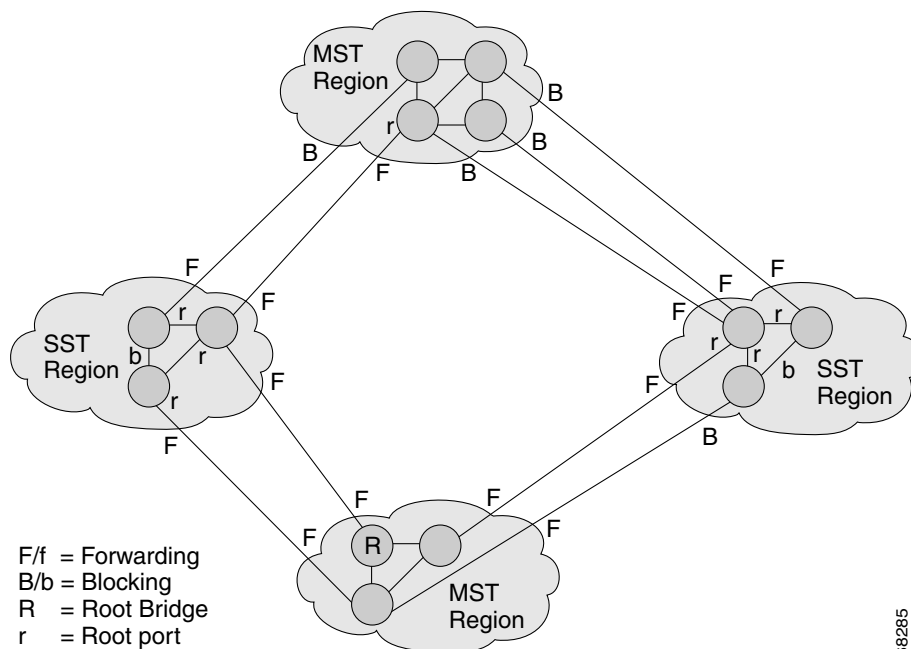
- MST runs a variant of spanning tree called internal spanning tree (IST). IST augments the common spanning tree (CST) information with internal information about the MST region. The MST region appears as a single bridge to adjacent single spanning tree (SST) and MST regions.
- A bridge running MST provides interoperability with single spanning tree bridges as follows:
 - MST bridges run IST, which augments the common spanning tree (CST) information with internal information about the MST region.
 - IST connects all the MST bridges in the region and appears as a subtree in the CST that includes the whole bridged domain. The MST region appears as a virtual bridge to adjacent SST bridges and MST regions.
 - The common and internal spanning tree (CIST) is the collection of ISTs in each MST region, the CST that interconnects the MST regions, and the SST bridges. CIST is the same as an IST inside an MST region and the same as CST outside an MST region. The STP, RSTP, and MSTP together elect a single bridge as the root of the CIST.

- MST establishes and maintains additional spanning trees within each MST region. These spanning trees are referred to as MST instances (MSTIs). The IST is numbered 0, and the MSTIs are numbered 1,2,3, and so on. Any MSTI is local to the MST region that is independent of MSTIs in another region, even if the MST regions are interconnected. MST instances combine with the IST at the boundary of MST regions to become the CST as follows:
 - Spanning tree information for an MSTI is contained in an MSTP record (M-record).
M-records are always encapsulated within MST BPDUs (MST BPDUs). The original spanning trees computed by MSTP are called M-trees. M-trees are active only within the MST region. M-trees merge with the IST at the boundary of the MST region and form the CST.
- MST provides interoperability with PVST+ by generating PVST+ BPDUs for the non-CST VLANs.
- MST supports some of the PVST+ extensions in MSTP as follows:
 - UplinkFast and BackboneFast are not available in MST mode; they are included in RSTP.
 - PortFast is supported.
 - BPDU filter and BPDU guard are supported in MST mode.
 - Loop guard and root guard are supported in MST. MST preserves the VLAN 1 disabled functionality except that BPDUs are still transmitted in VLAN 1.
 - MST switches operate as if MAC reduction is enabled.
 - For private VLANs (PVLANS), secondary VLANs must be mapped to the same instance as the primary.

MST-to-PVST Interoperability

A virtual bridged LAN may contain interconnected regions of single spanning tree (SST) and MST bridges. [Figure 15-8](#) shows this relationship.

Figure 15-8 Network with Interconnected SST and MST Regions



An MST region appears as an SST or pseudobridge to STP running in the SST region. Pseudobridges operate as follows:

- The same values for root identifiers and root path costs are sent in all BPDUs of all the pseudobridge ports. Pseudobridges differ from a single SST bridge as follows:
 - The pseudobridge BPDUs have different bridge identifiers. This difference does not affect STP operation in the neighboring SST regions because the root identifier and root cost are the same.
 - BPDUs sent from the pseudobridge ports may have significantly different message ages. Because the message age increases by 1 second for each hop, the difference in the message age is in the order of seconds.
- Data traffic from one port of a pseudobridge (a port at the edge of a region) to another port follows a path entirely contained within the pseudobridge or MST region.
- Data traffic belonging to different VLANs may follow different paths within the MST regions established by MST.
- Loop prevention is achieved by either of the following:
 - Blocking the appropriate pseudobridge ports by allowing one forwarding port on the boundary and blocking all other ports.
 - Setting the CST partitions to block the ports of the SST regions.
- A pseudobridge differs from a single SST bridge because the BPDUs sent from the pseudobridge's ports have different bridge identifiers. The root identifier and root cost are the same for both bridges.

These guidelines apply in a topology where you configure MST switches (all in the same region) to interact with PVST+ switches:

- Configure the root for all VLANs inside the MST region as shown in this example:

```
Router# show spanning-tree mst interface gigabitethernet 1/1

GigabitEthernet1/1 of MST00 is root forwarding
Edge port: no (trunk) port guard : none (default)
Link type: point-to-point (auto) bpdu filter: disable (default)
Boundary : boundary (PVST) bpdu guard : disable (default)
Bpdus sent 10, received 310

Instance Role Sts Cost Prio.Nbr Vlans mapped
-----
0 Root FWD 20000 128.1 1-2,4-2999,4000-4094
3 Boun FWD 20000 128.1 3,3000-3999
```

The ports that belong to the MST switch at the boundary simulate PVST+ and send PVST+ BPDUs for all the VLANs.

If you enable loop guard on the PVST+ switches, the ports might change to a loop-inconsistent state when the MST switches change their configuration. To correct the loop-inconsistent state, you must disable and reenable loop guard on that PVST+ switch.

- Do not locate the root for some or all of the VLANs inside the PVST+ side of the MST switch because when the MST switch at the boundary receives PVST+ BPDUs for all or some of the VLANs on its designated ports, root guard sets the port to the blocking state. Do not designate switches with a slower CPU running PVST+ as a switch running MST.

When you connect a PVST+ switch to two different MST regions, the topology change from the PVST+ switch does not pass beyond the first MST region. In this case, the topology changes are only propagated in the instance to which the VLAN is mapped. The topology change stays local to the first MST region and the CAM entries in the other region are not flushed. To make the topology change visible throughout other MST regions, you can map that VLAN to IST or connect the PVST+ switch to the two regions through access links.

Common Spanning Tree

CST (802.1Q) is a single spanning tree for all the VLANs. In a Catalyst 6000 family switch running PVST+, the VLAN 1 spanning tree corresponds to CST. In a Catalyst 6500 series switch running MST, IST (instance 0) corresponds to CST.

MST Instances

This release supports up to 16 instances; each spanning tree instance is identified by an instance ID that ranges from 0 to 15. Instance 0 is mandatory and is always present. Instances 1 through 15 are optional.

MST Configuration Parameters

MST configuration includes these three parts:

- Name—A 32-character string (null padded) identifying the MST region.
- Revision number—An unsigned 16-bit number that identifies the revision of the current MST configuration.



Note You must set the revision number when required as part of the MST configuration. The revision number is not incremented automatically each time you commit the MST configuration.

- MST configuration table—An array of 4096 bytes. Each byte, interpreted as an unsigned integer, corresponds to a VLAN. The value is the instance number to which the VLAN is mapped. The first byte that corresponds to VLAN 0 and the 4096th byte that corresponds to VLAN 4095 are unused and always set to zero.

You must configure each byte manually. You can use SNMP or the CLI to perform the configuration.

MST BPDUs contain the MST configuration ID and the checksum. An MST bridge accepts an MST BPDU only if the MST BPDU configuration ID and the checksum match its own MST region configuration ID and checksum. If one value is different, the MST BPDU is considered to be an SST BPDU.

MST Regions

These sections describe MST regions:

- [MST Region Overview, page 15-19](#)
- [Boundary Ports, page 15-19](#)
- [IST Master, page 15-19](#)
- [Edge Ports, page 15-20](#)
- [Link Type, page 15-20](#)

MST Region Overview

Interconnected bridges that have the same MST configuration are referred to as an MST region. There is no limit on the number of MST regions in the network.

To form an MST region, bridges can be either of the following:

- An MST bridge that is the only member of the MST region.
- An MST bridge interconnected by a LAN. A LAN's designated bridge has the same MST configuration as an MST bridge. All the bridges on the LAN can process MST BPDUs.

If you connect two MST regions with different MST configurations, the MST regions do the following:

- Load balance across redundant paths in the network. If two MST regions are redundantly connected, all traffic flows on a single connection with the MST regions in a network.
- Provide an RSTP handshake to enable rapid connectivity between regions. However, the handshaking is not as fast as between two bridges. To prevent loops, all the bridges inside the region must agree upon the connections to other regions. This situation introduces a delay. We do not recommend partitioning the network into a large number of regions.

Boundary Ports

A boundary port is a port that connects to a LAN, the designated bridge, of which is either an SST bridge, or a bridge with a different MST configuration. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement message from an RST or MST bridge with a different configuration.

At the boundary, the role of MST ports do not matter; their state is forced to be the same as the IST port state. If the boundary flag is set for the port, the MSTP port-role selection process assigns a port role to the boundary and assigns the same state as the state of the IST port. The IST port at the boundary can take up any port role except a backup port role.

IST Master

The IST master of an MST region is the bridge with the lowest bridge identifier and the least path cost to the CST root. If an MST bridge is the root bridge for CST, then it is the IST master of that MST region. If the CST root is outside the MST region, then one of the MST bridges at the boundary is selected as the IST master. Other bridges on the boundary that belong to the same region eventually block the boundary ports that lead to the root.

If two or more bridges at the boundary of a region have an identical path to the root, you can set a slightly lower bridge priority to make a specific bridge the IST master.

The root path cost and message age inside a region stay constant, but the IST path cost is incremented and the IST remaining hops are decremented at each hop. To display the information about the IST master, path cost, and remaining hops for the bridge, enter the **show spanning-tree mst** command.

Edge Ports

An edge port is a port that is connected to a nonbridging device (for example, a host or a router). A port that connects to a hub is also an edge port if the hub or any LAN that is connected by it does not have a bridge. An edge port can start forwarding as soon as the link is up.

MST requires that you configure all ports for each host or router. To establish rapid connectivity after a failure, you need to block the nonedge designated ports of an intermediate bridge. If the port connects to another bridge that can send back an agreement, then the port starts forwarding immediately. Otherwise, the port needs twice the forward delay time to start forwarding again. You must explicitly configure the ports that are connected to the hosts and routers as edge ports while using MST.

To prevent a misconfiguration, the PortFast operation is turned off if the port receives a BPDU. To display the configured and operational status of PortFast, enter the **show spanning-tree mst interface** command.

Link Type

Rapid connectivity is established only on point-to-point links. You must configure ports explicitly to a host or router. However, cabling in most networks meets this requirement, and you can avoid explicit configuration by treating all full-duplex links as point-to-point links by entering the **spanning-tree linktype** command.

Message Age and Hop Count

IST and MST instances do not use the message age and maximum age timer settings in the BPDU. IST and MST use a separate hop-count process that is very similar to the IP TTL process. You can configure each MST bridge with a maximum hop count. The root bridge of the instance sends a BPDU (or M-record) with the remaining hop count that is equal to the maximum hop count. When a bridge receives a BPDU (or M-record), it decrements the received remaining hop count by one. The bridge discards the BPDU (M-record) and ages out the information held for the port if the count reaches zero after decrementing. The nonroot bridges propagate the decremented count as the remaining hop count in the BPDUs (M-records) they generate.

The message age and maximum age timer settings in the RST portion of the BPDU remain the same throughout the region, and the same values are propagated by the region's designated ports at the boundary.

Default STP Configuration

Table 15-5 shows the default STP configuration.

Table 15-5 STP Default Configuration

Feature	Default Value
Enable state	STP enabled for all VLANs
Bridge priority	32768
STP port priority (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	128
STP port cost (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	<ul style="list-style-type: none"> • 10-Gigabit Ethernet: 2 • Gigabit Ethernet: 4 • Fast Ethernet: 19 • Ethernet: 100
STP VLAN port priority (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	128
STP VLAN port cost (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	<ul style="list-style-type: none"> • 10-Gigabit Ethernet: 2 • Gigabit Ethernet: 4 • Fast Ethernet: 19 • Ethernet: 100
Hello time	2 seconds
Forward delay time	15 seconds
Maximum aging time	20 seconds
Mode	PVST

STP and MST Configuration Guidelines

Follow these guidelines when configuring MST:

- Do not disable spanning tree on any VLAN in any of the PVST bridges.
- Do not use PVST bridges as the root of CST.
- Ensure that all PVST spanning tree root bridges have lower (numerically higher) priority than the CST root bridge.

- Ensure that trunks carry all of the VLANs mapped to an instance or do not carry any VLANs at all for this instance.
- Do not connect switches with access links because access links may partition a VLAN.
- Complete any MST configuration involving a large number of either existing or new logical VLAN ports during a maintenance window because the complete MST database gets reinitialized for any incremental change (such as adding new VLANs to instances or moving VLANs across instances).

Configuring STP

These sections describe how to configure STP on VLANs:

- [Enabling STP, page 15-23](#)
- [Enabling the Extended System ID, page 15-24](#)
- [Configuring the Root Bridge, page 15-25](#)
- [Configuring a Secondary Root Bridge, page 15-26](#)
- [Configuring STP Port Priority, page 15-27](#)
- [Configuring STP Port Cost, page 15-29](#)
- [Configuring the Bridge Priority of a VLAN, page 15-30](#)
- [Configuring the Hello Time, page 15-32](#)
- [Configuring the Forward-Delay Time for a VLAN, page 15-32](#)
- [Configuring the Maximum Aging Time for a VLAN, page 15-33](#)
- [Enabling Rapid-PVST, page 15-33](#)



Note

- The STP commands described in this chapter can be configured on any LAN port, but they are in effect only on LAN ports configured with the **switchport** keyword.
- With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.



Caution

We do not recommend disabling spanning tree, even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN.

Enabling STP



Note

STP is enabled by default on VLAN 1 and on all newly created VLANs.

You can enable STP on a per-VLAN basis. The Catalyst 6500 series switch maintains a separate instance of STP for each VLAN (except on VLANs on which you disable STP).

To enable STP on a per-VLAN basis, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i>	Enables STP on a per-VLAN basis. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-5 on page 15-21).
	Router(config)# default spanning-tree vlan <i>vlan_ID</i>	Reverts all STP parameters to default values for the specified VLAN.
	Router(config)# no spanning-tree vlan <i>vlan_ID</i>	Disables STP on the specified VLAN; see the following Cautions for information regarding this command.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i>	Verifies that STP is enabled.



Caution

Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network.



Caution

We do not recommend disabling spanning tree, even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN.

This example shows how to enable STP on VLAN 200:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200
Router(config)# end
Router#
```



Note

Because STP is enabled by default, entering a **show running** command to view the resulting configuration does not display the command you entered to enable STP.

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200

VLAN0200
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
            Address     00d0.00b8.14c8
            This bridge is the root
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
            Address     00d0.00b8.14c8
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time  300

Interface          Role Sts Cost          Prio.Nbr Status
-----
Fa4/4              Desg FWD 200000        128.196 P2p
Fa4/5              Back BLK 200000        128.197 P2p

Router#
```

**Note**

You must have at least one interface that is active in VLAN 200 to create a VLAN 200 spanning tree. In this example, two interfaces are active in VLAN 200.

Enabling the Extended System ID

**Note**

The extended system ID is enabled permanently on chassis that support 64 MAC addresses.

You can enable the extended system ID on chassis that support 1024 MAC addresses (see the [“Understanding the Bridge ID” section on page 15-3](#)).

To enable the extended system ID, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree extend system-id	Enables the extended system ID.
	Router(config)# no spanning-tree extend system-id	Disables the extended system ID.
		Note You cannot disable the extended system ID on chassis that support 64 MAC addresses or when you have configured extended range VLANs (see “STP Default Configuration” section on page 15-21).
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i>	Verifies the configuration.

**Note**

When you enable or disable the extended system ID, the bridge IDs of all active STP instances are updated, which might change the spanning tree topology.

This example shows how to enable the extended system ID:

```
Router# configure terminal
Router(config)# spanning-tree extend system-id
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary | include Extended
Extended system ID is enabled.
```

Configuring the Root Bridge

Catalyst 6500 series switches maintain a separate instance of STP for each active VLAN. A bridge ID, consisting of the bridge priority and the bridge MAC address, is associated with each instance. For each VLAN, the network device with the lowest bridge ID becomes the root bridge for that VLAN.

To configure a VLAN instance to become the root bridge, enter the **spanning-tree vlan *vlan_ID* root** command to modify the bridge priority from the default value (32768) to a significantly lower value.

When you enter the **spanning-tree vlan *vlan_ID* root** command, the switch checks the bridge priority of the current root bridges for each VLAN. When the extended system ID is disabled, the switch sets the bridge priority for the specified VLANs to 8192 if this value will cause the switch to become the root for the specified VLANs. When the extended system ID is enabled, the switch sets the bridge priority for the specified VLANs to 24576 if this value will cause the switch to become the root for the specified VLANs.

If the extended system ID is disabled and if any root bridge for the specified VLANs has a bridge priority lower than 8192, the switch sets the bridge priority for the specified VLANs to 1 less than the lowest bridge priority.

If the extended system ID is enabled and if any root bridge for the specified VLANs has a bridge priority lower than 24576, the switch sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority. (4096 is the value of the least significant bit of a 4-bit bridge priority value; see [Table 15-2 on page 15-3](#).)



Note

The **spanning-tree vlan *vlan_ID* root** command fails if the value required to be the root bridge is less than 1.

The **spanning-tree vlan *vlan_ID* root** command can cause the following effects:

- If the extended system ID is disabled, and if all network devices in VLAN 100 have the default priority of 32768, entering the **spanning-tree vlan 100 root primary** command on the switch sets the bridge priority for VLAN 100 to 8192, which causes the switch to become the root bridge for VLAN 100.
- If the extended system ID is enabled, and if all network devices in VLAN 20 have the default priority of 32768, entering the **spanning-tree vlan 20 root primary** command on the switch sets the bridge priority to 24576, which causes the switch to become the root bridge for VLAN 20.



Caution

The root bridge for each instance of STP should be a backbone or distribution switch. Do not configure an access switch as the STP primary root.

Use the **diameter** keyword to specify the Layer 2 network diameter (that is, the maximum number of bridge hops between any two end stations in the Layer 2 network). When you specify the network diameter, the Catalyst 6500 series switch automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can use the **hello** keyword to override the automatically calculated hello time.

**Note**

To preserve a stable STP topology, we recommend that you avoid configuring the hello time, forward delay time, and maximum age time manually after configuring the Catalyst 6500 series switch as the root bridge.

To configure a Catalyst 6500 series switch as the root bridge, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i> root primary [diameter <i>hops</i> [hello-time <i>seconds</i>]]	Configures a Catalyst 6500 series switch as the root bridge. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-5 on page 15-21).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> root	Clears the root bridge configuration.
Step 2	Router(config)# end	Exits configuration mode.

This example shows how to configure the Catalyst 6500 series switch as the root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# end
Router#
```

Configuring a Secondary Root Bridge

When you configure a Catalyst 6500 series switch as the secondary root, the STP bridge priority is modified from the default value (32768) so that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other network devices in the network use the default bridge priority of 32768).

If the extended system ID is enabled, STP sets the bridge priority to 28672. If the extended system ID is disabled, STP sets the bridge priority to 16384.

You can run this command on more than one Catalyst 6500 series switch to configure multiple backup root bridges. Use the same network diameter and hello time values as you used when configuring the primary root bridge.

To configure a Catalyst 6500 series switch as the secondary root bridge, perform this task:

	Command	Purpose
Step 1	Router(config)# [no] spanning-tree vlan <i>vlan_ID</i> root secondary [diameter <i>hops</i> [hello-time <i>seconds</i>]]	Configures a Catalyst 6500 series switch as the secondary root bridge. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> root	Clears the root bridge configuring.
Step 2	Router(config)# end	Exits configuration mode.

This example shows how to configure the Catalyst 6500 series switch as the secondary root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root secondary diameter 4
Router(config)# end
Router#
```

Configuring STP Port Priority

If a loop occurs, STP considers port priority when selecting a LAN port to put into the forwarding state. You can assign higher priority values to LAN ports that you want STP to select first and lower priority values to LAN ports that you want STP to select last. If all LAN ports have the same priority value, STP puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports. The possible priority range is 0 through 240 (default 128), configurable in increments of 16.

Cisco IOS uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

To configure the STP port priority of a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{ <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>port_channel_number</i> }}	Selects an interface to configure.
Step 2	Router(config-if)# spanning-tree port-priority <i>port_priority</i>	Configures the port priority for the LAN interface. The <i>port_priority</i> value can be from 1 to 252 in increments of 4.
	Router(config-if)# no spanning-tree port-priority	Reverts to the default port priority value.
Step 3	Router(config-if)# spanning-tree vlan <i>vlan_ID</i> port-priority <i>port_priority</i>	Configures the VLAN port priority for the LAN interface. The <i>port_priority</i> value can be from 1 to 252 in increments of 4. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
	Router(config-if)# [no] spanning-tree vlan <i>vlan_ID</i> port-priority	Reverts to the default VLAN port priority value.
Step 4	Router(config-if)# end	Exits configuration mode.

	Command	Purpose
Step 5	<pre>Router# show spanning-tree interface {type¹ slot/port} {port-channel port_channel_number} Router# show spanning-tree vlan vlan_ID</pre>	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabithernet, or tengigabithernet

This example shows how to configure the STP port priority of Fast Ethernet port 4/4:

```
Router# configure terminal
Router(config)# interface fastethernet 4/4
Router(config-if)# spanning-tree port-priority 160
Router(config-if)# end
Router#
```

This example shows how to verify the configuration of Fast Ethernet port 4/4:

```
Router# show spanning-tree interface fastethernet 4/4
Vlan          Role Sts Cost      Prio.Nbr Status
-----
VLAN0001      Back BLK 200000   160.196 P2p
VLAN0006      Back BLK 200000   160.196 P2p
...
VLAN0198      Back BLK 200000   160.196 P2p
VLAN0199      Back BLK 200000   160.196 P2p
VLAN0200      Back BLK 200000   160.196 P2p
Router#
```

Fastethernet 4/4 is a trunk. Several VLANs are configured and active as shown in the example. The port priority configuration applies to all VLANs on this interface.



Note

The **show spanning-tree interface** command only displays information if the port is connected and operating. If this condition is not met, enter a **show running-config interface** command to verify the configuration.

This example shows how to configure the VLAN port priority of Fast Ethernet port 4/4:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree vlan 200 port-priority 64
Router(config-if)# ^Z
Router#
```

The configuration entered in the example only applies to VLAN 200. All VLANs other than 200 still have a port priority of 160.

This example shows how to verify the configuration:

```
Router# show spanning-tree interface fastethernet 4/4
Vlan          Role Sts Cost      Prio.Nbr Status
-----
VLAN0001      Back BLK 200000   160.196 P2p
VLAN0006      Back BLK 200000   160.196 P2p
...
VLAN0199      Back BLK 200000   160.196 P2p
VLAN0200      Desg FWD 200000    64.196  P2p
Router#
```

You also can display spanning tree information for VLAN 200 using the following command:

```
Router# show spanning-tree vlan 200 interface fastEthernet 4/4
Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Desg LRN 200000    64.196  P2p
```

Configuring STP Port Cost

The STP port path cost default value is determined from the media speed of a LAN interface. If a loop occurs, STP considers port cost when selecting a LAN interface to put into the forwarding state. You can assign lower cost values to LAN interfaces that you want STP to select first and higher cost values to LAN interfaces that you want STP to select last. If all LAN interfaces have the same cost value, STP puts the LAN interface with the lowest LAN interface number in the forwarding state and blocks other LAN interfaces. The possible cost range is 0 through 200000000 (the default is media specific).

STP uses the port cost value when the LAN interface is configured as an access port and uses VLAN port cost values when the LAN interface is configured as a trunk port.

To configure the STP port cost of a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel port_channel_number}}	Selects an interface to configure.
Step 2	Router(config-if)# spanning-tree cost port_cost	Configures the port cost for the LAN interface. The <i>port_cost</i> value can be from 1 to 200000000 (1 to 65535 in Release 12.1(2)E and earlier releases).
	Router(config-if)# no spanning-tree cost	Reverts to the default port cost.
Step 3	Router(config-if)# [no] spanning-tree vlan vlan_ID cost port_cost	Configures the VLAN port cost for the LAN interface. The <i>port_cost</i> value can be from 1 to 200000000. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
Step 4	Router(config-if)# no spanning-tree vlan vlan_ID cost	Reverts to the default VLAN port cost.
Step 5	Router(config-if)# end	Exits configuration mode.
Step 6	Router# show spanning-tree interface {type ¹ slot/port} {port-channel port_channel_number} show spanning-tree vlan vlan_ID	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to change the STP port cost of Fast Ethernet port 4/4:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree cost 1000
Router(config-if)# ^Z
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree interface fastEthernet 4/4
Vlan      Role Sts Cost      Prio.Nbr Status
-----
VLAN0001  Back BLK 1000    160.196  P2p
```

```

VLAN0006      Back BLK 1000      160.196 P2p
VLAN0007      Back BLK 1000      160.196 P2p
VLAN0008      Back BLK 1000      160.196 P2p
VLAN0009      Back BLK 1000      160.196 P2p
VLAN0010      Back BLK 1000      160.196 P2p
Router#

```

This example shows how to configure the port priority at an individual port VLAN cost for VLAN 200:

```

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree vlan 200 cost 2000
Router(config-if)# ^Z
Router#

```

This example shows how to verify the configuration:

```

Router# show spanning-tree vlan 200 interface fastEthernet 4/4
Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Desg FWD 2000      64.196 P2p

```

**Note**

In the following output other VLANs (VLAN 1 for example) have not been affected by this configuration.

```

Router# show spanning-tree vlan 1 interface fastEthernet 4/4
Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Back BLK 1000      160.196 P2p
Router#

```

**Note**

The **show spanning-tree** command only displays information for ports that are in link-up operative state and are appropriately configured for DTP. If these conditions are not met, you can enter a **show running-config** command to confirm the configuration.

Configuring the Bridge Priority of a VLAN

**Note**

Be careful when using this command. For most situations, we recommend that you enter the **spanning-tree vlan *vlan_ID* root primary** and the **spanning-tree vlan *vlan_ID* root secondary** commands to modify the bridge priority.

To configure the STP bridge priority of a VLAN when the extended system ID is disabled, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i> priority <i>bridge_priority</i>	Configures the bridge priority of a VLAN when the extended system ID is disabled. The <i>bridge_priority</i> value can be from 1 to 65535. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> priority	Reverts to the default bridge priority value.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> bridge [detail]	Verifies the configuration.

To configure the STP bridge priority of a VLAN when the extended system ID is enabled, perform this task:

	Command	Purpose
Step 1	Router(config)# [no] spanning-tree vlan <i>vlan_ID</i> priority {0 4096 8192 12288 16384 20480 24576 28672 32768 36864 40960 45056 49152 53248 57344 61440}	Configures the bridge priority of a VLAN when the extended system ID is enabled. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> priority	Reverts to the default bridge priority value.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> bridge [detail]	Verifies the configuration.

This example shows how to configure the bridge priority of VLAN 200 to 33792 when the extended system ID is disabled:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 priority 33792
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
Vlan                Bridge ID           Hello Time  Max Age  Fwd Delay  Protocol
-----
VLAN200             33792 0050.3e8d.64c8   2         20       15       ieee
Router#
```

Configuring the Hello Time



Note

Be careful when using this command. For most situations, we recommend that you use the **spanning-tree vlan *vlan_ID* root primary** and **spanning-tree vlan *vlan_ID* root secondary** commands to modify the hello time.

To configure the STP hello time of a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i> hello-time <i>hello_time</i>	Configures the hello time of a VLAN. The <i>hello_time</i> value can be from 1 to 10 seconds. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> hello-time	Reverts to the default hello time.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> bridge [detail]	Verifies the configuration.

This example shows how to configure the hello time for VLAN 200 to 7 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 hello-time 7
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge

Vlan                Bridge ID           Hello Max  Fwd
                   Time  Age Delay  Protocol
-----
VLAN200             49152 0050.3e8d.64c8   7   20   15  ieee
Router#
```

Configuring the Forward-Delay Time for a VLAN

To configure the STP forward delay time for a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i> forward-time <i>forward_time</i>	Configures the forward time of a VLAN. The <i>forward_time</i> value can be from 4 to 30 seconds. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> forward-time	Reverts to the default forward time.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> bridge [detail]	Verifies the configuration.

This example shows how to configure the forward delay time for VLAN 200 to 21 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 forward-time 21
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge

Vlan                Bridge ID           Hello Time  Max Age  Fwd Delay  Protocol
-----
VLAN200             49152 0050.3e8d.64c8  2         20       21       ieee
Router#
```

Configuring the Maximum Aging Time for a VLAN

To configure the STP maximum aging time for a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i> max-age <i>max_age</i>	Configures the maximum aging time of a VLAN. The <i>max_age</i> value can be from 6 to 40 seconds. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 9-1 on page 9-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> max-age	Reverts to the default maximum aging time.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> bridge [detail]	Verifies the configuration.

This example shows how to configure the maximum aging time for VLAN 200 to 36 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 max-age 36
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge

Vlan                Bridge ID           Hello Time  Max Age  Fwd Delay  Protocol
-----
VLAN200             49152 0050.3e8d.64c8  2         36       15       ieee
Router#
```

Enabling Rapid-PVST

Rapid-PVST uses the existing PVST+ framework for configuration and interaction with other features. It also supports some of the PVST+ extensions.

To enable Rapid-PVST mode on the switch, enter the **spanning-tree mode rapid-pvst** command in privileged mode. To configure the switch in Rapid-PVST mode, see the “[Configuring STP](#)” section on [page 15-22](#).

Specifying the Link Type

Rapid connectivity is established only on point-to-point links. Spanning tree views a point-to-point link as a segment connecting only two switches running the spanning tree algorithm. Because the switch assumes that all full-duplex links are point-to-point links and that half-duplex links are shared links, you can avoid explicitly configuring the link type. To configure a specific link type, enter the **spanning-tree linktype** command.

Restarting Protocol Migration

A switch running both MSTP and RSTP supports a built-in protocol migration process that enables the switch to interoperate with legacy 802.1D switches. If this switch receives a legacy 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MSTP switch can also detect that a port is at the boundary of a region when it receives a legacy BPDU, or an MST BPDU (version 3) associated with a different region, or an RST BPDU (version 2).

However, the switch does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the entire switch, you can use the **clear spanning-tree detected-protocols** privileged EXEC command. To restart the protocol migration process on a specific interface, enter the **clear spanning-tree detected-protocols interface *interface-id*** privileged EXEC command.

Configuring IEEE 802.1s MST

Release 12.1(13)E and later releases support MST. These sections describe how to configure MST:

- [Enabling MST, page 15-34](#)
- [Displaying MST Configurations, page 15-36](#)
- [Configuring MST Instance Parameters, page 15-39](#)
- [Configuring MST Instance Port Parameters, page 15-40](#)
- [Restarting Protocol Migration, page 15-40](#)

Enabling MST

To enable and configure MST on the switch, perform these tasks in privileged mode:

	Command	Purpose
Step 1	Router# show spanning-tree mst configuration	Displays the current MST configuration.
Step 2	Router(config)# spanning-tree mode mst	Configures MST mode.
Step 3	Router(config)# spanning-tree mst configuration	Configures the MST region by entering the MST configuration submenu.
	Router(config)# no spanning-tree mst configuration	Clears the MST configuration.

	Command	Purpose
Step 4	Router(config-mst)# show current	Displays the current MST configuration from within the MST configuration submenu
Step 5	Router(config-mst)# name name revision revision_number instance instance_number vlan vlan_range	Enters the MST configuration.
Step 6	Router(config-mst)# no instance instance_number	(Optional) Unmaps all VLANs that were mapped to an instance.
Step 7	Router(config-mst)# no instance instance_number vlan vlan_number	(Optional) Unmaps a VLAN from an instance.
Step 8	Router(config-mst)# end	Applies the configuration and exit configuration mode.
Step 9	Router# show spanning-tree mst config	Shows the MST configuration from the global configuration mode.

These examples show how to enable MST:

```

Router# show spanning-tree mst configuration
% Switch is not in mst mode
Name      []
Revision  0
Instance  Vlans mapped
-----
0         1-4094
-----

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# spanning-tree mode mst

Router(config)# spanning-tree mst configuration

Router(config-mst)# show current
Current MST configuration
Name      []
Revision  0
Instance  Vlans mapped
-----
0         1-4094
-----

Router(config-mst)# name cisco
Router(config-mst)# revision 2
Router(config-mst)# instance 1 vlan 1
Router(config-mst)# instance 2 vlan 1-1000
Router(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0         1001-4094
2         1-1000
-----

Router(config-mst)# no instance 2
Router(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0         1-4094
-----

```

```

Router(config-mst)# instance 1 vlan 2000-3000
Router(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0          1-1999,2500,3001-4094
1          2000-2499,2501-3000
-----

Router(config)# exit
Router(config)# no spanning-tree mst configuration
Router(config)# do show spanning-tree mst configuration
Name      []
Revision  0
Instance  Vlans mapped
-----
0          1-4094
-----

```

Displaying MST Configurations

To display MST configurations, perform these tasks in MST mode:

	Command	Purpose
Step 1	Router# show spanning-tree mst configuration	Displays the active configuration.
Step 2	Router# show spanning-tree mst [<i>detail</i>]	Displays information about the MST instances currently running.
Step 3	Router# show spanning-tree mst <i>instance-id</i> [<i>detail</i>]	Displays information about a specific MST instance.
Step 4	Router# show spanning-tree mst interface <i>interface name</i> [<i>detail</i>]	Displays information for a given port.
Step 5	Router# show spanning-tree mst <i>number interface interface name</i> [<i>detail</i>]	Displays MST information for a given port and a given instance.
Step 6	Router# show spanning-tree mst [<i>x</i>] [<i>interface Y</i>] <i>detail</i>	Displays detailed MST information.
Step 7	Router# show spanning-tree vlan <i>vlan_ID</i>	Displays VLAN information in MST mode.

These examples show how to display spanning tree VLAN configurations in MST mode:

```

Router(config)# spanning-tree mst configuration
Router(config-mst)# instance 1 vlan 1-10
Router(config-mst)# name cisco
Router(config-mst)# revision 1
Router(config-mst)# ^Z

Router# show spanning-tree mst configuration
Name      [cisco]
Revision  1
Instance  Vlans mapped
-----
0          11-4094
1          1-10
-----

```

```

Router# show spanning-tree mst

##### MST00          vlans mapped: 11-4094
Bridge      address 00d0.00b8.1400  priority 32768 (32768 sysid 0)
Root       address 00d0.004a.3c1c  priority 32768 (32768 sysid 0)
           port    Fa4/48          path cost 203100
IST master  this switch
Operational hello time 2, forward delay 15, max age 20, max hops 20
Configured  hello time 2, forward delay 15, max age 20, max hops 20

Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Back BLK 1000      160.196 P2p
Fa4/5          Desg FWD 200000     128.197 P2p
Fa4/48        Root FWD 200000     128.240 P2p Bound(STP)

##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400  priority 32769 (32768 sysid 1)
Root       this switch for MST01

Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Back BLK 1000      160.196 P2p
Fa4/5          Desg FWD 200000     128.197 P2p
Fa4/48        Boun FWD 200000     128.240 P2p Bound(STP)

Router# show spanning-tree mst 1

##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400  priority 32769 (32768 sysid 1)
Root       this switch for MST01

Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Back BLK 1000      160.196 P2p
Fa4/5          Desg FWD 200000     128.197 P2p
Fa4/48        Boun FWD 200000     128.240 P2p Bound(STP)

Router# show spanning-tree mst interface fastEthernet 4/4

FastEthernet4/4 of MST00 is backup blocking
Edge port:no          (default)          port guard :none          (default)
Link type:point-to-point (auto)          bpdu filter:disable      (default)
Boundary :internal          bpdu guard :disable      (default)
Bpdus sent 2, received 368

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-----
0        Back BLK 1000      160.196 11-4094
1        Back BLK 1000      160.196 1-10

Router# show spanning-tree mst 1 interface fastEthernet 4/4

FastEthernet4/4 of MST01 is backup blocking
Edge port:no          (default)          port guard :none          (default)
Link type:point-to-point (auto)          bpdu filter:disable      (default)
Boundary :internal          bpdu guard :disable      (default)
Bpdus (MRecords) sent 2, received 364

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-----
1        Back BLK 1000      160.196 1-10

```

```

Router# show spanning-tree mst 1 detail

##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 32769 (32768 sysid 1)
Root       this switch for MST01

FastEthernet4/4 of MST01 is backup blocking
Port info      port id      160.196 priority 160 cost      1000
Designated root address 00d0.00b8.1400 priority 32769 cost      0
Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.197
Timers:message expires in 5 sec, forward delay 0, forward transitions 0
Bpdus (MRecords) sent 123, received 1188

FastEthernet4/5 of MST01 is designated forwarding
Port info      port id      128.197 priority 128 cost      200000
Designated root address 00d0.00b8.1400 priority 32769 cost      0
Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.197
Timers:message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 1188, received 123

FastEthernet4/48 of MST01 is boundary forwarding
Port info      port id      128.240 priority 128 cost      200000
Designated root address 00d0.00b8.1400 priority 32769 cost      0
Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.240
Timers:message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 78, received 0

Router# show spanning-tree vlan 10

MST01
Spanning tree enabled protocol mstp
Root ID      Priority    32769
            Address    00d0.00b8.1400
            This bridge is the root
            Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID    Priority    32769 (priority 32768 sys-id-ext 1)
            Address    00d0.00b8.1400
            Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Back BLK 1000    160.196 P2p
Fa4/5          Desg FWD 200000    128.197 P2p

Router# show spanning-tree summary
Root bridge for:MST01
EtherChannel misconfiguration guard is enabled
Extended system ID is enabled
Portfast is disabled by default
PortFast BPDU Guard is disabled by default
Portfast BPDU Filter is disabled by default
Loopguard is disabled by default
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is long

Name          Blocking Listening Learning Forwarding STP Active
-----
MST00          1          0          0          2          3
MST01          1          0          0          2          3
-----
2 msts        2          0          0          4          6
Router#

```

Configuring MST Instance Parameters

To configure MST instance parameters, perform these tasks:

	Command	Purpose
Step 1	Router(config)# spanning-tree mst X priority Y	Configures the priority for an MST instance
Step 2	Router(config)# spanning-tree mst X root [primary secondary]	Configures the bridge as root for an MST instance.
Step 3	Router# show spanning-tree mst	Verifies the configuration.

This example shows how to configure MST instance parameters:

```

Router(config)# spanning-tree mst 1 priority ?
    <0-61440> bridge priority in increments of 4096

Router(config)# spanning-tree mst 1 priority 1
% Bridge Priority must be in increments of 4096.
% Allowed values are:
    0      4096  8192  12288  16384  20480  24576  28672
    32768  36864  40960  45056  49152  53248  57344  61440

Router(config)# spanning-tree mst 1 priority 49152
Router(config)#

Router(config)# spanning-tree mst 0 root primary
mst 0 bridge priority set to 24576
mst bridge max aging time unchanged at 20
mst bridge hello time unchanged at 2
mst bridge forward delay unchanged at 15
Router(config)# ^Z
Router#

Router# show spanning-tree mst

##### MST00          vlans mapped: 11-4094
Bridge      address 00d0.00b8.1400 priority 24576 (24576 sysid 0)
Root        this switch for CST and IST
Configured  hello time 2, forward delay 15, max age 20, max hops 20

Interface   Role Sts Cost      Prio.Nbr Status
-----
Fa4/4       Back BLK 1000    160.196 P2p
Fa4/5       Desg FWD 200000    128.197 P2p
Fa4/48      Desg FWD 200000    128.240 P2p Bound(STP)

##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 49153 (49152 sysid 1)
Root        this switch for MST01

Interface   Role Sts Cost      Prio.Nbr Status
-----
Fa4/4       Back BLK 1000    160.196 P2p
Fa4/5       Desg FWD 200000    128.197 P2p
Fa4/48      Boun FWD 200000    128.240 P2p Bound(STP)

Router#

```

Configuring MST Instance Port Parameters

To configure MST instance port parameters, perform these tasks:

	Command	Purpose
Step 1	Router(config-if)# spanning-tree mst x cost y	Configures the MST instance port cost.
Step 2	Router(config-if)# spanning-tree mst x port-priority y	Configures the MST instance port priority.
Step 3	Router# show spanning-tree mst x interface y	Verifies the configuration.

This example shows how to configure MST instance port parameters:

```
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree mst 1 ?
    cost          Change the interface spanning tree path cost for an instance
    port-priority Change the spanning tree port priority for an instance

Router(config-if)# spanning-tree mst 1 cost 1234567
Router(config-if)# spanning-tree mst 1 port-priority 240
Router(config-if)# ^Z

Router# show spanning-tree mst 1 interface fastEthernet 4/4

FastEthernet4/4 of MST01 is backup blocking
Edge port:no                (default)          port guard :none          (default)
Link type:point-to-point (auto)          bpdu filter:disable      (default)
Boundary :internal          bpdu guard :disable      (default)
Bpdus (MRecords) sent 125, received 1782

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-----
1          Back BLK 1234567  240.196  1-10

Router#
```

Restarting Protocol Migration

A switch running both MSTP and RSTP supports a built-in protocol migration mechanism that enables the switch to interoperate with legacy 802.1D switches. If this switch receives a legacy 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MSTP switch can also detect that a port is at the boundary of a region when it receives a legacy BPDU, an MST BPDU (version 3) associated with a different region, or an RST BPDU (version 2).

However, the switch does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the entire switch, you can use the **clear spanning-tree detected-protocols** privileged EXEC command. Use the **clear spanning-tree detected-protocols interface *interface-id*** privileged EXEC command to restart the protocol migration process on a specific interface.

This example shows how to restart protocol migration:

```
Router# clear spanning-tree detected-protocols interface fastEthernet 4/4
Router#
```




Configuring Optional STP Features

This chapter describes how to configure optional STP features.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How PortFast Works, page 16-2](#)
- [Understanding How BPDU Guard Works, page 16-2](#)
- [Understanding How PortFast BPDU Filtering Works, page 16-2](#)
- [Understanding How UplinkFast Works, page 16-3](#)
- [Understanding How BackboneFast Works, page 16-4](#)
- [Understanding How EtherChannel Guard Works, page 16-6](#)
- [Understanding How Root Guard Works, page 16-6](#)
- [Understanding How Loop Guard Works, page 16-6](#)
- [Enabling PortFast, page 16-8](#)
- [Enabling PortFast BPDU Filtering, page 16-10](#)
- [Enabling BPDU Guard, page 16-11](#)
- [Enabling UplinkFast, page 16-12](#)
- [Enabling BackboneFast, page 16-13](#)
- [Enabling EtherChannel Guard, page 16-14](#)
- [Enabling Root Guard, page 16-14](#)
- [Enabling Loop Guard, page 16-15](#)



Note

-
- For information on configuring the spanning tree protocol (STP), see [Chapter 15, “Configuring STP and IEEE 802.1s MST.”](#)
 - With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.
-

Understanding How PortFast Works

STP PortFast causes a Layer 2 LAN interface configured as an access port to enter the forwarding state immediately, bypassing the listening and learning states. You can use PortFast on Layer 2 access ports connected to a single workstation or server to allow those devices to connect to the network immediately, instead of waiting for STP to converge. Interfaces connected to a single workstation or server should not receive bridge protocol data units (BPDUs). When configured for PortFast, a port is still running the spanning tree protocol. A PortFast enabled port can immediately transition to the blocking state if necessary (this could happen on receipt of a superior BPDU).

With Release 12.1(11b)E:

- PortFast can be enabled on trunk ports
- PortFast can have an operational value that is different from the configured value.

**Caution**

Because the purpose of PortFast is to minimize the time that access ports must wait for STP to converge, it should only be used on access ports. If you enable PortFast on a port connected to a switch, you might create a temporary bridging loop.

Understanding How BPDU Guard Works

When enabled on a port, BPDU Guard shuts down a port that receives a BPDU. When configured globally, BPDU Guard is only effective on ports in the operational PortFast state. In a valid configuration, PortFast Layer 2 LAN interfaces do not receive BPDUs. Reception of a BPDU by a PortFast Layer 2 LAN interface signals an invalid configuration, such as connection of an unauthorized device. BPDU Guard provides a secure response to invalid configurations, because the administrator must manually put the Layer 2 LAN interface back in service. With release 12.1(11b)E, BPDU Guard can also be configured at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the PortFast configuration.

**Note**

When enabled globally, BPDU Guard applies to all interfaces that are in an operational PortFast state.

Understanding How PortFast BPDU Filtering Works

Release 12.1(13)E and later releases support PortFast BPDU filtering, which allows the administrator to prevent the system from sending or even receiving BPDUs on specified ports.

When configured globally, PortFast BPDU filtering applies to all operational PortFast ports. Ports in an operational PortFast state are supposed to be connected to hosts, that typically drop BPDUs. If an operational PortFast port receives a BPDU, it immediately loses its operational PortFast status. In that case, PortFast BPDU filtering is disabled on this port and STP resumes sending BPDUs on this port.

PortFast BPDU filtering can also be configured on a per-port basis. When PortFast BPDU filtering is explicitly configured on a port, it does not send any BPDUs and drops all BPDUs it receives.

**Caution**

Explicate configuring PortFast BPDU filtering on a port that is not connected to a host can result in bridging loops as the port will ignore any BPDU it receives and go to forwarding.

When you enable PortFast BPDUs filtering globally and set the port configuration as the default for PortFast BPDUs filtering (see the “[Enabling PortFast BPDUs Filtering](#)” section on page 16-10), then PortFast enables or disables PortFast BPDUs filtering.

If the port configuration is not set to default, then the PortFast configuration will not affect PortFast BPDUs filtering. [Table 16-1](#) lists all the possible PortFast BPDUs filtering combinations. PortFast BPDUs filtering allows access ports to move directly to the forwarding state as soon as the end hosts are connected.

Table 16-1 PortFast BPDUs Filtering Port Configurations

Per-Port Configuration	Global Configuration	PortFast State	PortFast BPDUs Filtering State
Default	Enable	Enable	Enable ¹
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

1. The port transmits at least 10 BPDUs. If this port receives any BPDUs, then PortFast and PortFast BPDUs filtering are disabled.

Understanding How UplinkFast Works

UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 LAN interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

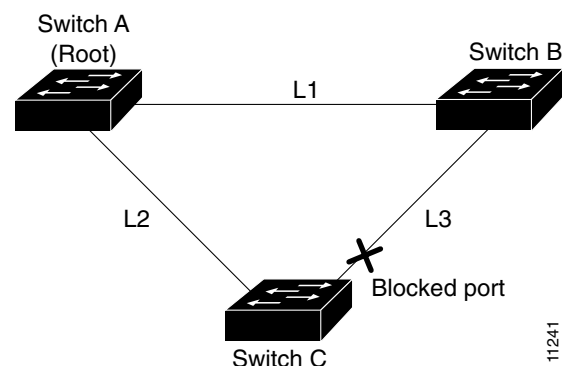


Note

UplinkFast is most useful in wiring-closet switches. This feature may not be useful for other types of applications.

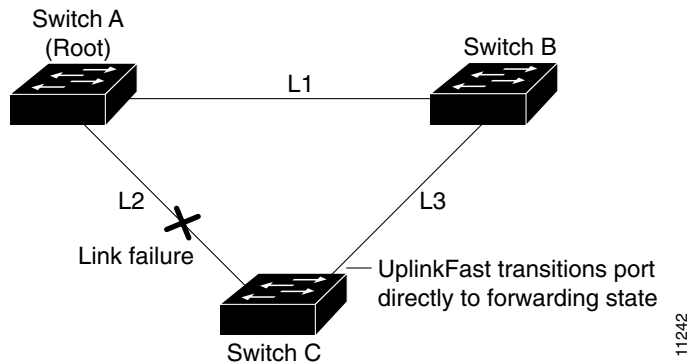
[Figure 16-1](#) shows an example topology with no link failures. Switch A, the root bridge, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 16-1 UplinkFast Example Before Direct Link Failure



If Switch C detects a link failure on the currently active link L2 on the root port (a *direct* link failure), UplinkFast unblocks the blocked port on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 16-2. This switchover takes approximately one to five seconds.

Figure 16-2 UplinkFast Example After Direct Link Failure



Understanding How BackboneFast Works

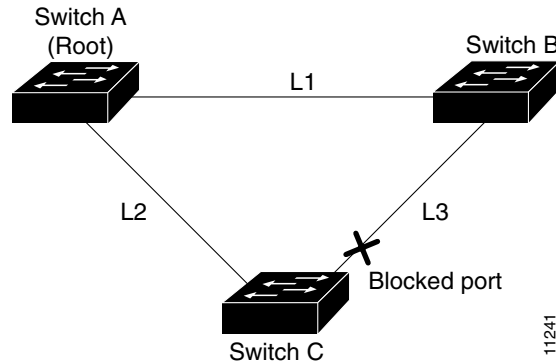
BackboneFast is initiated when a root port or blocked port on a network device receives inferior BPDUs from its designated bridge. An inferior BPDU identifies one network device as both the root bridge and the designated bridge. When a network device receives an inferior BPDU, it indicates that a link to which the network device is not directly connected (an *indirect* link) has failed (that is, the designated bridge has lost its connection to the root bridge). Under normal STP rules, the network device ignores inferior BPDUs for the configured maximum aging time, as specified by the STP **max-age** command.

The network device tries to determine if it has an alternate path to the root bridge. If the inferior BPDU arrives on a blocked port, the root port and other blocked ports on the network device become alternate paths to the root bridge. (Self-looped ports are not considered alternate paths to the root bridge.) If the inferior BPDU arrives on the root port, all blocked ports become alternate paths to the root bridge. If the inferior BPDU arrives on the root port and there are no blocked ports, the network device assumes that it has lost connectivity to the root bridge, causes the maximum aging time on the root to expire, and becomes the root bridge according to normal STP rules.

If the network device has alternate paths to the root bridge, it uses these alternate paths to transmit a new kind of Protocol Data Unit (PDU) called the Root Link Query PDU. The network device sends the Root Link Query PDU out all alternate paths to the root bridge. If the network device determines that it still has an alternate path to the root, it causes the maximum aging time to expire on the ports on which it received the inferior BPDU. If all the alternate paths to the root bridge indicate that the network device has lost connectivity to the root bridge, the network device causes the maximum aging times on the ports on which it received an inferior BPDU to expire. If one or more alternate paths can still connect to the root bridge, the network device makes all ports on which it received an inferior BPDU its designated ports and moves them out of the blocking state (if they were in the blocking state), through the listening and learning states, and into the forwarding state.

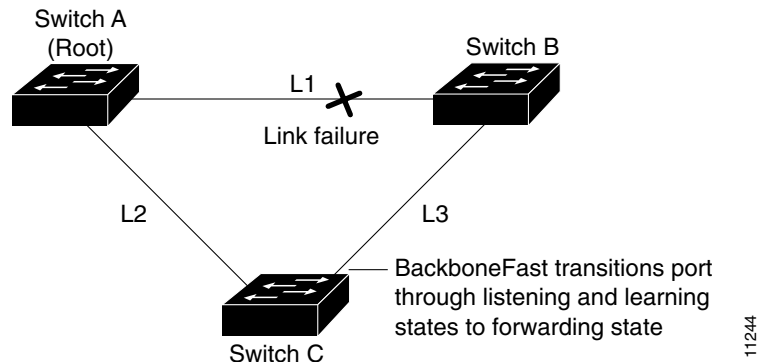
Figure 16-3 shows an example topology with no link failures. Switch A, the root bridge, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that connects directly to Switch B is in the blocking state.

Figure 16-3 BackboneFast Example Before Indirect Link Failure



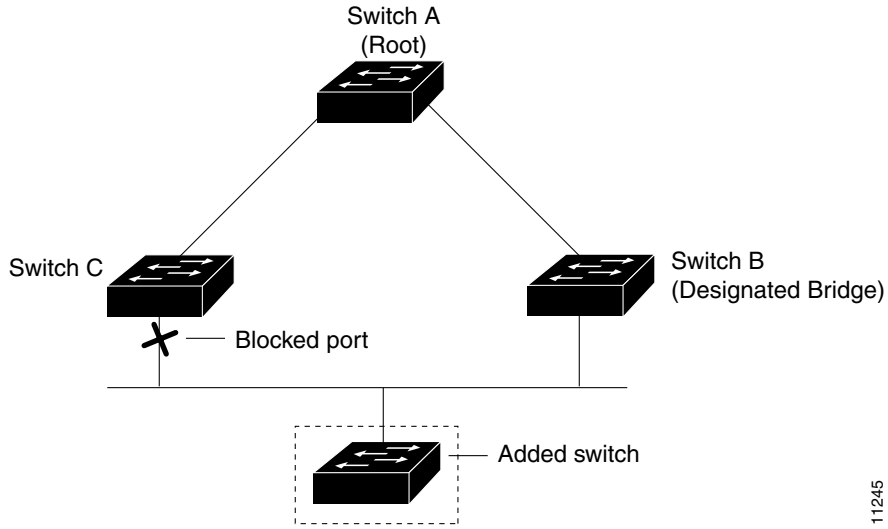
If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root bridge over L1, it detects the failure and elects itself the root and begins sending BPDUs to Switch C indicating itself as the root. When Switch C receives the inferior BPDUs from Switch B, Switch C infers that an indirect failure has occurred. At that point, BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the maximum aging time for the port to expire. BackboneFast then transitions the Layer 2 LAN interface on Switch C to the forwarding state, providing a path from Switch B to Switch A. This switchover takes approximately 30 seconds, twice the Forward Delay time if the default Forward Delay time of 15 seconds is set. Figure 16-4 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 16-4 BackboneFast Example After Indirect Link Failure



If a new network device is introduced into a shared-medium topology as shown in Figure 16-5, BackboneFast is not activated because the inferior BPDUs did not come from the recognized designated bridge (Switch B). The new network device begins sending inferior BPDUs that indicate that it is the root bridge. However, the other network devices ignore these inferior BPDUs and the new network device learns that Switch B is the designated bridge to Switch A, the root bridge.

Figure 16-5 Adding a Network Device in a Shared-Medium Topology



11245

Understanding How EtherChannel Guard Works

EtherChannel guard detects a misconfigured EtherChannel where interfaces on the Catalyst 6500 series switch are configured as an EtherChannel while interfaces on the other device are not or not all the interfaces on the other device are in the same EtherChannel.

In response to misconfiguration detected on the other device, EtherChannel guard puts interfaces on the Catalyst 6500 series switch into the errdisabled state.

Understanding How Root Guard Works

The STP root guard feature prevents a port from becoming root port or blocked port. If a port configured for root guard receives a superior BPDU, the port immediately goes to the root-inconsistent (blocked) state.

Understanding How Loop Guard Works

Loop guard helps prevent bridging loops that could occur because of a uni-directional link failure on a point-to-point link. When enabled globally, the loop guard applies to all point-to-point ports on the system. Loop guard detects root ports and blocked ports and ensures that they keep receiving BPDUs from their designated port on the segment. If a loop guard enabled root or blocked port stop a receiving BPDUs from its designated port, it transitions to the loop-inconsistent blocking state, assuming there is a physical link error on this port. The port recovers from this loop-inconsistent state as soon as it receives a BPDU.

You can enable loop guard on a per-port basis. When you enable loop guard, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable loop guard, it is disabled for the specified ports. Disabling loop guard moves all loop-inconsistent ports to the listening state.

If you enable loop guard on a channel and the first link becomes unidirectional, loop guard blocks the entire channel until the affected port is removed from the channel. Figure 16-6 shows loop guard in a triangle switch configuration.

Figure 16-6 Triangle Switch Configuration with Loop Guard

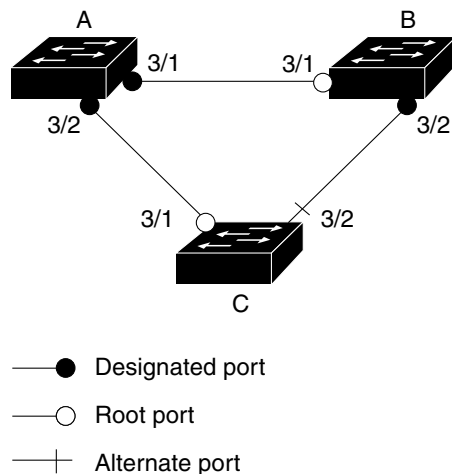


Figure 16-6 illustrates the following configuration:

- Switches A and B are distribution switches.
- Switch C is an access switch.
- Loop guard is enabled on ports 3/1 and 3/2 on Switches A, B, and C.

Enabling loop guard on a root switch has no effect but provides protection when a root switch becomes a nonroot switch.

Follow these guidelines when using loop guard:

- You cannot enable loop guard on PortFast-enabled or dynamic VLAN ports.
- You cannot enable loop guard if root guard is enabled.

Loop guard interacts with other features as follows:

- Loop guard does not affect the functionality of UplinkFast or BackboneFast.
- Enabling loop guard on ports that are not connected to a point-to-point link will not work.
- Root guard forces a port to be always designated as the root port. Loop guard is effective only if the port is a root port or an alternate port. You cannot enable loop guard and root guard on a port at the same time.
- Loop guard uses the ports known to spanning tree. Loop guard can take advantage of logical ports provided by the Port Aggregation Protocol (PAgP). However, to form a channel, all the physical ports grouped in the channel must have compatible configurations. PAgP enforces uniform configurations of root guard or loop guard on all the physical ports to form a channel.

These caveats apply to loop guard:

- Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, loop guard blocks the channel, even if other links in the channel are functioning properly.
- If a set of ports that are already blocked by loop guard are grouped together to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.

- If a channel is blocked by loop guard and the channel breaks, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.



Note You can enable UniDirectional Link Detection (UDLD) to help isolate the link failure. A loop may occur until UDLD detects the failure, but loop guard will not be able to detect it.

- Loop guard has no effect on a disabled spanning tree instance or a VLAN.

Enabling PortFast



Caution

Use PortFast *only* when connecting a single end station to a Layer 2 access port. Otherwise, you might create a network loop.

To enable PortFast on a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {type ¹ slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 2	Router(config-if)# spanning-tree portfast	Enables PortFast on a Layer 2 access port connected to a single workstation or server.
Step 3	Router(config-if)# spanning-tree portfast default	Disables PortFast.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show running interface {type ¹ slot/port} {port-channel port_channel_number}	Verifies the configuration.

1. type = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable PortFast on Fast Ethernet interface 5/8:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree portfast
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show running-config interface fastethernet 5/8
Building configuration...

Current configuration:
!
interface FastEthernet5/8
 no ip address
 switchport
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast
end

Router#
```


To enable the default PortFast configuration, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree portfast default	Configures the PortFast default.
Step 2	Router(config)# show spanning-tree summary totals	Verifies the global configuration.
Step 3	Router(config)# show spanning-tree interface x detail	Verifies the effect on a specific port.
Step 4	Router(config-if)# spanning-tree portfast trunk	Enables the PortFast trunk on a port
Step 5	Router# show spanning-tree interface fastEthernet x detail	Verifies the configuration.

This example shows how to enable the default PortFast configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# spanning-tree portfast default
Router(config)# ^Z
```

```
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID is disabled
Portfast is enabled by default
PortFast BPDU Guard is disabled by default
Portfast BPDU Filter is disabled by default
Loopguard is disabled by default
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is long
```

```
Name                Blocking Listening Learning Forwarding STP Active
-----
VLAN0001             0          0          0          1          1
VLAN0010             0          0          0          2          2
-----
2 vlans              0          0          0          3          3
Router#
```

```
Router# show spanning-tree interface fastEthernet 4/4 detail
```

```
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by default
  Link type is point-to-point by default
  BPDU:sent 10, received 0
```

```
Router(config-if)# spanning-tree portfast trunk
%Warning:portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION
```

```
Router(config-if)# ^Z
```

```

Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
Port path cost 1000, Port priority 160, Port Identifier 160.196.
Designated root has priority 32768, address 00d0.00b8.140a
Designated bridge has priority 32768, address 00d0.00b8.140a
Designated port id is 160.196, designated path cost 0
Timers:message age 0, forward delay 0, hold 0
Number of transitions to forwarding state:1
The port is in the portfast mode by portfast trunk configuration
Link type is point-to-point by default
BPDU:sent 30, received 0
Router#

```

Enabling PortFast BPDU Filtering

These sections describe how to configure PortFast BPDU filtering on the switch:

To enable PortFast BPDU filtering globally, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree portfast bpdudfilter default	Enables BPDU filtering globally on the switch.
Step 2	Router# show spanning-tree summary totals	Verifies the configuration.

BPDU filtering is set to default on each port. This example shows how to enable PortFast BPDU filtering on the port and verify the configuration in PVST+ mode:



Note

For PVST+ information, see [Chapter 15, “Configuring STP and IEEE 802.1s MST.”](#)

```

Router(config)# spanning-tree portfast bpdudfilter default
Router(config)# ^Z

```

```

Router# show spanning-tree summary totals
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast              is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard             is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long

```

```

Name                    Blocking Listening Learning Forwarding STP Active
-----
2 vlans                  0          0          0          3          3
Router#

```

To enable PortFast BPDU filtering on a nontrunking port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface fastEthernet 4/4	Selects the interface to configure.
Step 2	Router(config-if)# spanning-tree bpduguard enable	Enables BPDU filtering.
Step 3	Router# show spanning-tree interface fastEthernet 4/4	Verifies the configuration.

This example shows how to enable PortFast BPDU filtering on a nontrunking port:

```
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree bpduguard enable
Router(config-if)# ^Z

Router# show spanning-tree interface fastEthernet 4/4

Vlan          Role Sts Cost      Prio.Nbr Status
-----
VLAN0010      Desg FWD 1000      160.196 Edge P2p
Router# show spanning-tree interface fastEthernet 4/4 detail
Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
Port path cost 1000, Port priority 160, Port Identifier 160.196.
Designated root has priority 32768, address 00d0.00b8.140a
Designated bridge has priority 32768, address 00d0.00b8.140a
Designated port id is 160.196, designated path cost 0
Timers:message age 0, forward delay 0, hold 0
Number of transitions to forwarding state:1
The port is in the portfast mode by portfast trunk configuration
Link type is point-to-point by default
Bpdu filter is enabled
BPDU:sent 0, received 0
Router#
```

Enabling BPDU Guard

To enable BPDU Guard globally, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree portfast bpduguard default	Enables BPDU Guard globally.
	Router(config)# no spanning-tree portfast bpduguard default	Disables BPDU Guard globally.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree summary totals	Verifies the configuration.

This example shows how to enable BPDU Guard:

```
Router# configure terminal
Router(config)# spanning-tree portfast bpduguard
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary totals default
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID is disabled
Portfast is enabled by default
PortFast BPDU Guard is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard is disabled by default
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is long

Name Blocking Listening Learning Forwarding STP Active
-----
2 vlans 0 0 0 3 3
Router#
```

Enabling UplinkFast

UplinkFast increases the bridge priority to 49152 and adds 3000 to the STP port cost of all Layer 2 LAN interfaces on the Catalyst 6500 series switch, decreasing the probability that the switch will become the root bridge. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second). UplinkFast cannot be enabled on VLANs that have been configured for bridge priority. To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan *vlan_ID* priority** command in global configuration mode.



Note

When you enable UplinkFast, it affects all VLANs on the Catalyst 6500 series switch. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree uplinkfast [max-update-rate <i>max_update_rate</i>]	Enables UplinkFast.
	Router(config)# no spanning-tree uplinkfast max-update-rate	Reverts to the default rate.
	Router(config)# no spanning-tree uplinkfast	Disables UplinkFast.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i>	Verifies that UplinkFast is enabled.

This example shows how to enable UplinkFast with an update rate of 400 packets per second:

```
Router# configure terminal
Router(config)# spanning-tree uplinkfast max-update-rate 400
Router(config)# exit
Router#
```

This example shows how to verify that UplinkFast is enabled:

```
Router# show spanning-tree uplinkfast
UplinkFast is enabled
Router#
```

Enabling BackboneFast



Note

BackboneFast operates correctly only when enabled on all network devices in the network. BackboneFast is not supported on Token Ring VLANs. This feature is supported for use with third-party network devices.

To enable BackboneFast, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree backbonefast	Enables BackboneFast.
	Router(config)# no spanning-tree backbonefast	Disables BackboneFast.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i>	Verifies that BackboneFast is enabled.

This example shows how to enable BackboneFast:

```
Router# configure terminal
Router(config)# spanning-tree backbonefast
Router(config)# end
Router#
```

This example shows how to verify that BackboneFast is enabled:

```
Router# show spanning-tree backbonefast
BackboneFast is enabled

BackboneFast statistics
-----
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)    : 0
Number of RLQ request PDUs received (all VLANs)  : 0
Number of RLQ response PDUs received (all VLANs) : 0
Number of RLQ request PDUs sent (all VLANs)      : 0
Number of RLQ response PDUs sent (all VLANs)     : 0
Router#
```

Enabling EtherChannel Guard

To enable EtherChannel guard, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree etherchannel guard misconfig	Enables EtherChannel guard.
	Router(config)# no spanning-tree etherchannel guard misconfig	Disables EtherChannel guard.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree summary include EtherChannel	Verifies that EtherChannel guard is enabled.

This example shows how to enable EtherChannel guard:

```
Router# configure terminal
Router(config)# spanning-tree etherchannel guard misconfig
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary | include EtherChannel
EtherChannel misconfiguration guard is enabled
```

Enter the **show interface status err-disable** command to display interfaces in the errdisable state.

After the misconfiguration has been cleared, interfaces in the errdisable state might automatically recover. To manually return an interface to service, enter a **shutdown** and then a **no shutdown** command for the interface.

Enabling Root Guard

To enable root guard, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {type ¹ slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 2	Router(config-if)# spanning-tree guard root	Enables root guard.
	Router(config-if)# no spanning-tree guard root	Disables root guard.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show spanning-tree Router# show running interface {type ¹ slot/port} {port-channel port_channel_number}	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

Enter the **show spanning-tree inconsistentports** command to display ports that are in the root-inconsistent state.

Enabling Loop Guard

Use the **set spanning-tree guard** command to enable or disable the spanning tree loop guard feature on a per-port basis.

To enable loop guard globally on the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree loopguard default	Enables loop guard globally on the switch.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning tree interface 4/4 detail	Verifies the configuration impact on a port.

This example shows how to enable loop guard globally:

```
Router# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Router(config)# spanning-tree loopguard default
```

```
Router(config)# ^Z
```

```
Router# show spanning-tree interface fastEthernet 4/4 detail
```

```
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  Loop guard is enabled by default on the port
  BPDU:sent 0, received 0
```

To enable loop guard on an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {type ¹ slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 2	Router(config-if)# spanning-tree guard loop	Configures loop guard.
Step 3	Router(config)# end	Exits configuration mode.
Step 4	Router# show spanning tree interface 4/4 detail	Verifies the configuration impact on that port.

1. *type* = ethernet, fastethernet, gigabithernet, or tengigabithernet

This example shows how to enable loop guard:

```
Router# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Router(config)# interface fastEthernet 4/4
```

```
Router(config-if)# spanning-tree guard loop
```

```
Router(config-if)# ^Z
```

This example shows how to verify the configuration:

```
Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  Loop guard is enabled on the port
  BPDU:sent 0, received 0
Router#
```




Configuring IP Unicast Layer 3 Switching on Supervisor Engine 2

This chapter describes how to configure IP unicast Layer 3 switching for Policy Feature Card 2 (PFC2), Distributed Forwarding Cards (DFCs), and Multilayer Switch Feature Card 2 (MSFC2).



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication and the publications at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

This chapter consists of these sections:

- [Understanding How Layer 3 Switching Works, page 17-1](#)
- [Default Hardware Layer 3 Switching Configuration, page 17-4](#)
- [Layer 3 Switching Configuration Guidelines and Restrictions, page 17-4](#)
- [Configuring Hardware Layer 3 Switching, page 17-5](#)
- [Displaying Hardware Layer 3 Switching Statistics, page 17-6](#)



Note

- Supervisor Engine 2, PFC2, and MSFC2 support IPX with fast switching on the MSFC2. For more information, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_c/xcprt1/xcdips.htm

- For information about IP multicast Layer 3 switching, see [Chapter 18, “Configuring IP Multicast Layer 3 Switching.”](#)

Understanding How Layer 3 Switching Works

These sections describe Layer 3 switching with PFC2 and DFCs:

- [Understanding Hardware Layer 3 Switching on PFC2 and DFCs, page 17-2](#)
- [Understanding Layer 3-Switched Packet Rewrite, page 17-2](#)

Understanding Hardware Layer 3 Switching on PFC2 and DFCs

Hardware Layer 3 switching allows the PFC2 and DFCs, instead of the MSFC2, to forward IP unicast traffic between subnets. Hardware Layer 3 switching provides wire-speed forwarding on the PFC2 and DFCs, instead of in software on the MSFC2. Hardware Layer 3 switching requires minimal support from the MSFC2. The MSFC2 routes any traffic that cannot be hardware Layer 3 switched.

Hardware Layer 3 switching supports the routing protocols configured on the MSFC2. Hardware Layer 3 switching does not replace the routing protocols configured on the MSFC2.

Hardware Layer 3 switching, which runs equally on the PFC2 and DFCs to provide IP unicast Layer 3 switching locally on each module, consists of the following functions:

- Hardware access control list (ACL) switching—For policy-based routing (PBR)
- Hardware NetFlow switching—For TCP intercept, reflexive ACL forwarding decisions, Web Cache Communication Protocol (WCCP), and server load balancing (SLB)
- Hardware Cisco Express Forwarding (CEF) switching—For all other IP unicast traffic

Hardware Layer 3 switching on the PFC2 supports modules that do not have a DFC. The MSFC2 forwards traffic that cannot be Layer 3 switched.

Traffic is hardware Layer 3 switched after being processed by access lists and quality of service (QoS).

Hardware Layer 3 switching makes a forwarding decision locally on the ingress-port module for each packet and sends the rewrite information for each packet to the egress port, where the rewrite occurs when the packet is transmitted from the Catalyst 6500 series switch.

Hardware Layer 3 switching generates flow statistics for Layer 3-switched traffic. Hardware Layer 3 flow statistics can be used for NetFlow Data Export (NDE). (See [Chapter 33, “Configuring NDE”](#).)

Understanding Layer 3-Switched Packet Rewrite

When a packet is Layer 3 switched from a source in one subnet to a destination in another subnet, the Catalyst 6500 series switch performs a packet rewrite at the egress port based on information learned from the MSFC2 so that the packets appear to have been routed by the MSFC2.

Packet rewrite alters five fields:

- Layer 2 (MAC) destination address
- Layer 2 (MAC) source address
- Layer 3 IP Time to Live (TTL)
- Layer 3 checksum
- Layer 2 (MAC) checksum (also called the frame checksum or FCS)



Note

Packets are rewritten with the encapsulation appropriate for the next-hop subnet.

If Source A and Destination B are in different subnets and Source A sends a packet to the MSFC2 to be routed to Destination B, the switch recognizes that the packet was sent to the Layer 2 (MAC) address of the MSFC2.

To perform Layer 3 switching, the switch rewrites the Layer 2 frame header, changing the Layer 2 destination address to the Layer 2 address of Destination B and the Layer 2 source address to the Layer 2 address of the MSFC2. The Layer 3 addresses remain the same.

In IP unicast and IP multicast traffic, the switch decrements the Layer 3 TTL value by 1 and recomputes the Layer 3 packet checksum. The switch recomputes the Layer 2 frame checksum and forwards (or, for multicast packets, replicates as necessary) the rewritten packet to Destination B's subnet.

A received IP unicast packet is formatted (conceptually) as follows:

Layer 2 Frame Header		Layer 3 IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
MSFC2 MAC	Source A MAC	Destination B IP	Source A IP	n	calculation1		

After the switch rewrites an IP unicast packet, it is formatted (conceptually) as follows:

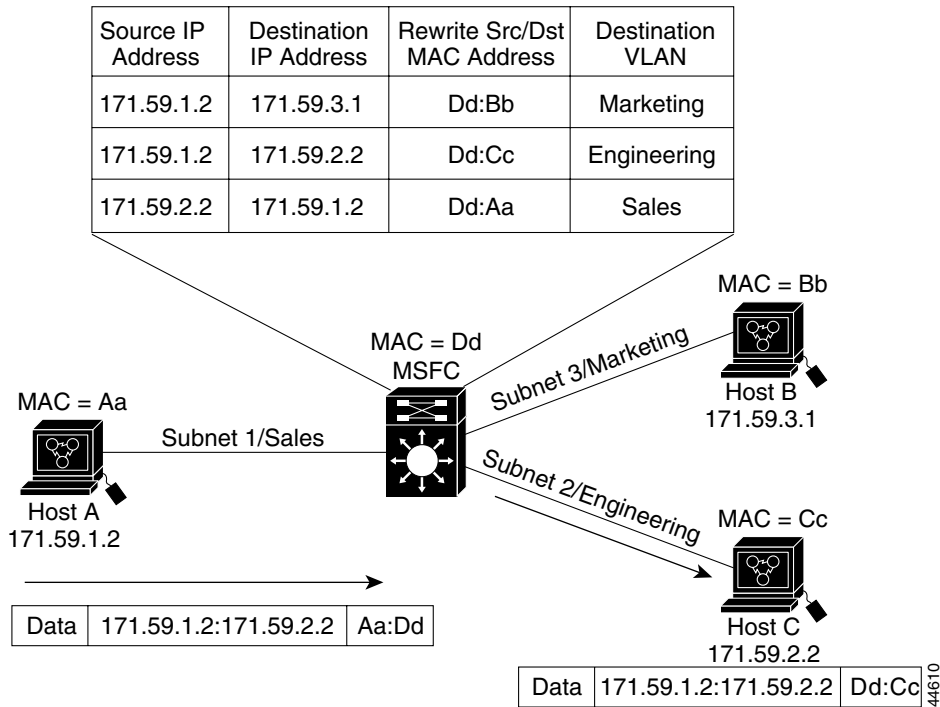
Layer 2 Frame Header		Layer 3 IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
Destination B MAC	MSFC2 MAC	Destination B IP	Source A IP	n-1	calculation2		

Hardware Layer 3 Switching Examples

Figure 17-1 on page 17-3 shows a simple network topology. In this example, Host A is on the Sales VLAN (IP subnet 171.59.1.0), Host B is on the Marketing VLAN (IP subnet 171.59.3.0), and Host C is on the Engineering VLAN (IP subnet 171.59.2.0).

When Host A initiates an HTTP file transfer to Host C, Hardware Layer 3 switching uses the information in the local forwarding information base (FIB) and adjacency table to forward packets from Host A to Host C.

Figure 17-1 Hardware Layer 3 Switching Example Topology



Default Hardware Layer 3 Switching Configuration

Table 17-1 shows the default hardware Layer 3 switching configuration.

Table 17-1 Default Hardware Layer 3 Switching Configuration

Feature	Default Value
Hardware Layer 3 switching enable state	Enabled (cannot be disabled)
Cisco IOS CEF enable state on MSFC2	Enabled (cannot be disabled)
Cisco IOS dCEF ¹ enable state on MSFC2	Enabled (cannot be disabled)
IGMP ² snooping	Enabled
Multicast routing on MSFC2	Disabled globally
PIM ³ routing on MSFC2	Disabled on all Layer 3 interfaces
IP multicast Layer 3 switching threshold	Unconfigured—no default value
IP multicast Layer 3 switching	Enabled when multicast routing is enabled and IP PIM is enabled on the interface

1. dCEF = Distributed Cisco Express Forwarding
2. IGMP = Internet Group Management Protocol
3. PIM = Protocol Independent Multicast

Layer 3 Switching Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring hardware Layer 3 switching:

- The PFC2 supports a maximum of 16 unique Hot Standby Routing Protocol (HSRP) group numbers. You can use the same HSRP group numbers in different VLANs. If you configure more than 16 HSRP groups, this restriction prevents use of the VLAN number as the HSRP group number.



Note Identically numbered HSRP groups use the same virtual MAC address, which might cause errors if you configure bridge groups.

- Hardware Layer 3 switching supports the following ingress and egress encapsulations:
 - Ethernet V2.0 (ARPA)
 - 802.3 with 802.2 with 1 byte control (SAP1)
 - 802.3 with 802.2 and SNAP



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Configuring Hardware Layer 3 Switching


Note

For information on configuring unicast routing on the MSFC2, see [Chapter 12, “Configuring Layer 3 Interfaces.”](#)

Hardware Layer 3 switching is permanently enabled on Supervisor Engine 2 with PFC2, MSFC2, and Distributed Feature Card (DFC). No configuration is required.

To display information about Layer 3-switched traffic, perform this task:

Command	Purpose
Router# show interface {{type ¹ slot/port} {port-channel number}} begin L3	Displays a summary of Layer 3-switched traffic.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display information about hardware Layer 3-switched traffic on Fast Ethernet port 3/3:

```
Router# show interface fastethernet 3/3 | begin L3
L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 12 pkt, 778 bytes mcast
L3 out Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
4046399 packets input, 349370039 bytes, 0 no buffer
Received 3795255 broadcasts, 2 runts, 0 giants, 0 throttles
<...output truncated...>
Router#
```


Note

The Layer 3 switching packet count is updated approximately every five seconds.

Cisco IOS CEF and dCEF are permanently enabled on the MSFC2. No configuration is required to support hardware Layer 3 switching.

The Cisco IOS CEF **ip load-sharing per-packet**, **ip cef accounting per-prefix**, and **ip cef accounting non-recursive** commands on the MSFC2 apply only to traffic that is CEF-switched in software on the MSFC2. The commands do not affect traffic that is hardware Layer 3 switched on the PFC2 or on DFC-equipped switching modules.

For information about Cisco IOS CEF and dCEF on the MSFC2, refer to these publications:

- The “Cisco Express Forwarding” section at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_c/xcprt2/index.htm
- The *Cisco IOS Switching Services Command Reference* publication at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_r/index.htm

Displaying Hardware Layer 3 Switching Statistics

Hardware Layer 3 switching statistics are obtained on a per-VLAN basis.

To display hardware Layer 3 switching statistics, perform this task:

Command	Purpose
Router# show interfaces <i>{{type¹ slot/port}}</i> <i>{port-channel number}</i>	Displays hardware Layer 3 switching statistics.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display hardware Layer 3 switching statistics:

```
Router# show interfaces gigabitethernet 9/5 | include Switched
L2 Switched: ucast: 8199 pkt, 1362060 bytes - mcast: 6980 pkt, 371952 bytes
L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes mcast
L3 out Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
```

To display adjacency table information, perform this task:

Command	Purpose
Router# show adjacency <i>[{{type¹ slot/port}}</i> <i>{port-channel number}</i>] detail internal summary]	Displays adjacency table information. The optional detail keyword displays detailed adjacency information, including Layer 2 information.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display adjacency statistics:

```
Router# show adjacency gigabitethernet 9/5 detail
Protocol Interface Address
IP GigabitEthernet9/5 172.20.53.206(11)
504 packets, 6110 bytes
00605C865B82
000164F83FA50800
ARP 03:49:31
```



Note

Adjacency statistics are updated approximately every 60 seconds.



Configuring IP Multicast Layer 3 Switching

This chapter describes how to configure IP multicast Layer 3 switching on the Catalyst 6500 series switches.



Note

For more information on the syntax and usage for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How IP Multicast Layer 3 Switching Works, page 18-1](#)
- [Default IP Multicast Layer 3 Switching Configuration, page 18-7](#)
- [IP Multicast Layer 3 Switching Configuration Guidelines and Restrictions, page 18-8](#)
- [Configuring IP Multicast Layer 3 Switching, page 18-9](#)



Note

In this chapter, the term “PFC” refers to either a PFC2 or a PFC1, except when specifically differentiated, and the term “MSFC” refers to either an MSFC2 or an MSFC1, except when specifically differentiated.

Understanding How IP Multicast Layer 3 Switching Works

These sections describe how IP multicast Layer 3 switching works:

- [IP Multicast Layer 3 Switching Overview, page 18-2](#)
- [Multicast Layer 3 Switching Cache, page 18-2](#)
- [IP Multicast Layer 3 Switching Flow Mask, page 18-3](#)
- [Layer 3-Switched Multicast Packet Rewrite, page 18-3](#)
- [Partially and Completely Switched Flows, page 18-4](#)
- [Non-RPF Traffic Processing, page 18-5](#)

IP Multicast Layer 3 Switching Overview

Policy Feature Card 2 (PFC2) provides Layer 3 switching for IP multicast flows using the hardware replication table and hardware Cisco Express Forwarding (CEF), which uses the forwarding information base (FIB) and the adjacency table on the PFC2. In systems with Distributed Forwarding Cards (DFCs), IP multicast flows are Layer 3 switched locally using Multicast Distributed Hardware Switching (MDHS). MDHS uses local hardware CEF and replication tables on each DFC to perform Layer 3 switching and rate limiting of reverse path forwarding (RPF) failures locally on each DFC-equipped switching module.

The PFC2 and the DFCs support hardware switching of (*,G) state flows. PFC1, PFC2, and the DFCs support rate limiting of non-RPF traffic.

Policy Feature Card 1 (PFC1) provides Layer 3 switching of IP multicast flows with Multilayer Switching (MLS) using the NetFlow and hardware replication tables.

Multicast Layer 3 switching forwards IP multicast data packet flows between IP subnets using advanced application-specific integrated circuit (ASIC) switching hardware, offloading processor-intensive multicast forwarding and replication from network routers.

Layer 3 flows that cannot be hardware switched are still forwarded in software by routers. Protocol Independent Multicast (PIM) is used for route determination.

PFC1, PFC2, and the DFCs all use the Layer 2 multicast forwarding table to determine on which ports Layer 2 multicast traffic should be forwarded (if any). The multicast forwarding table entries are populated in conjunction with Internet Group Management Protocol (IGMP) snooping (see [Chapter 21, “Configuring IGMP Snooping”](#)).

Multicast Layer 3 Switching Cache

PFC1, PFC2, and the DFCs maintain Layer 3 switching information in one or more hardware tables as follows:

- PFC1 populates the Layer 3 flow as {source IP, IP group, ingress-interface/VLAN} in the NetFlow cache. It also stores the Layer 3 rewrite information and a pointer to a list of outgoing interfaces (such as replication entries) for the flow. If a flow does not match these parameters, it is considered a NetFlow miss and is bridged on the incoming port based on the Layer 2 lookup.
- PFC2 and DFC populate the (S,G) or (*,G) flows in the hardware FIB table with the appropriate masks; for example, (S/32, G/32) and (*/0, G/32). The RPF interface and the adjacency pointer information is also stored in each entry. The adjacency table contains the rewrite and a pointer to the replication entries. If a flow matches a FIB entry, the RPF check compares the incoming interface/VLAN with the entry. A mismatch is an RPF failure, which can be rate limited if this feature is enabled.

In systems with PFC1, the maximum switching cache size is 128K entries and is shared by all Layer 3 switching processes on the switch (such as IP unicast MLS and Internetwork Packet Exchange [IPX] MLS). However, a cache exceeding 32K entries increases the probability that a flow will not be switched by the PFC and will get forwarded to the MSFC.

In systems with PFC1 or PFC2, the MSFC updates its multicast routing table and forwards the new information to the PFC whenever it receives traffic for a new flow. In addition, if an entry in the multicast routing table on the MSFC ages out, the MSFC deletes the entry and forwards the updated information to the PFC. In systems with DFCs, flows are populated symmetrically on all DFCs and on PFC2.

The Layer 3 switching cache contains flow information for all active Layer 3-switched flows. After the switching cache is populated, multicast packets identified as belonging to an existing flow can be Layer 3 switched based on the cache entry for that flow. For each cache entry, the PFC maintains a list of outgoing interfaces for the IP multicast group. From this list, the PFC determines onto which VLANs traffic from a given multicast flow should be replicated.

These commands affect the Layer 3 switching cache entries:

- Clearing the multicast routing table (using the **clear ip mroute** command) clears all multicast Layer 3 switching cache entries.
- Disabling IP multicast routing on the MSFC (using the **no ip multicast-routing** command) purges all multicast Layer 3 switching cache entries on the PFC.
- Disabling multicast Layer 3 switching on an individual interface basis (using the **no mls ip multicast** command) causes flows that use this interface as the RPF interface to be routed only by the MSFC in software.

IP Multicast Layer 3 Switching Flow Mask

IP multicast Layer 3 switching with PFC1 supports only the multicast source-destination-VLAN flow mask. PFC1 maintains one multicast Layer 3 switching cache entry for each {source IP, destination group IP, source VLAN}. The multicast source-destination-VLAN flow mask differs from the IP unicast MLS source-destination-ip flow mask in that, for IP multicast Layer 3 switching, the source VLAN is included as part of the entry. The source VLAN is the multicast RPF interface for the multicast flow. Flows are based on the IP address of the source device, the destination IP multicast group address, and the source VLAN. The MSFC uses the RPF interface to send a unicast packet back to the source.

Layer 3-Switched Multicast Packet Rewrite



Note

Only ARPA rewrites are supported for IP multicast packets. Subnetwork Address Protocol (SNAP) rewrites are not supported.

When a multicast packet is Layer 3 switched from a multicast source to a destination multicast group, PFC1 performs a packet rewrite based on information learned from the MSFC and stored in the Layer 3 switching cache. In the case of PFC2 and the DFCs, the packet rewrite is based on information learned from the MSFC2 and is stored in the adjacency table. The format of the packet rewrite is the same for PFC1, PFC2, and DFCs.

For example, Server A sends a multicast packet addressed to IP multicast group G1. If there are members of group G1 on VLANs other than the source VLAN, the PFC must perform a packet rewrite when it replicates the traffic to the other VLANs (the switch also bridges the packet in the source VLAN).

When the PFC receives the multicast packet, it is formatted (conceptually) as follows:

Layer 2 Frame Header		Layer 3 IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
<i>Group G1 MAC¹</i>	<i>Source A MAC</i>	<i>Group G1 IP</i>	<i>Source A IP</i>	<i>n</i>	<i>calculation1</i>		

1. In this example, Destination B is a member of Group G1.

The PFC rewrites the packet as follows:

- Changes the source MAC address in the Layer 2 frame header from the MAC address of the host to the MAC address of the MSFC (this MAC address is stored in the multicast Layer 3 switching cache entry for the flow)
- Decrements the IP header Time to Live (TTL) by one and recalculates the IP header checksum

The result is a rewritten IP multicast packet that appears to have been routed. The PFC replicates the rewritten packet onto the appropriate destination VLANs, where it is forwarded to members of IP multicast group G1.

After the PFC performs the packet rewrite, the packet is formatted (conceptually) as follows:

Frame Header		IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
<i>Group G1 MAC</i>	<i>MSFC MAC</i>	<i>Group G1 IP</i>	<i>Source A IP</i>	<i>n-1</i>	<i>calculation2</i>		

Partially and Completely Switched Flows

When at least one outgoing Layer 3 interface for a given flow is multilayer switched and at least one outgoing interface is not multilayer switched, that flow is considered partially switched. When a partially switched flow is created, all multicast traffic belonging to that flow still reaches the MSFC and is software forwarded on those outgoing interfaces that are not multilayer switched.

These sections describe partially and completely switched flow:

- [Partially Switched Flows with PFC1 or PFC2, page 18-4](#)
- [Partially Switched Flows with PFC2, page 18-5](#)
- [Completely Switched Flows, page 18-5](#)

Partially Switched Flows with PFC1 or PFC2

If your system has a PFC1 or PFC2 installed, a flow might be partially switched instead of completely switched in these situations:

- The switch is configured as a member of the IP multicast group (using the **ip igmp join-group** command) on the RPF interface of the multicast source.
- During the registering state if the switch is the first-hop router to the source in PIM sparse mode (in this case, the switch must send PIM-register messages to the rendezvous point [RP]).
- The multicast TTL threshold is configured on an outgoing interface for the flow (using the **ip multicast ttl-threshold** command).
- The multicast helper is configured on the RPF interface for the flow, and multicast to broadcast translation is required.
- The outgoing interface is a generic routing encapsulation (GRE) Distance Vector Multicast Routing Protocol (DVMRP) tunnel interface.
- The maximum transmission unit (MTU) of the RPF interface is greater than the MTU of any outgoing interface.
- If Network Address Translation (NAT) is configured on an interface, and source address translation is required for the outgoing interface.

Partially Switched Flows with PFC2

In PFC2 systems, (*,G) flows will be partially switched on the last-hop leaf router if the shared-tree to shortest-path-tree (SPT) threshold is not equal to infinity. This allows the flow to transition from SPT.

**Note**

With a PFC2, flows matching an output ACL on an outgoing interface are routed in software.

Completely Switched Flows

When all the outgoing Layer 3 interfaces for a given flow are Layer 3 switched, and none of the above situations apply to the flow, that flow is considered completely switched. When a completely switched flow is created, the PFC prevents multicast traffic bridged on the source VLAN for that flow from reaching the MSFC interface in that VLAN, freeing the MSFC of the forwarding and replication load for that flow.

One consequence of a completely switched flow is that multicast statistics on a per-packet basis for that flow cannot be recorded. Therefore, the PFC periodically sends multicast packet and byte count statistics for all completely switched flows to the MSFC. The MSFC updates the corresponding multicast routing table entry and resets the expiration timer for that multicast route.

**Note**

A (*,G) state is created on the PIM-RP or for PIM-dense mode but is not used for forwarding the flows, and Layer 3 switching entries are not created for these flows.

Non-RPF Traffic Processing

These sections describe non-RPF traffic processing:

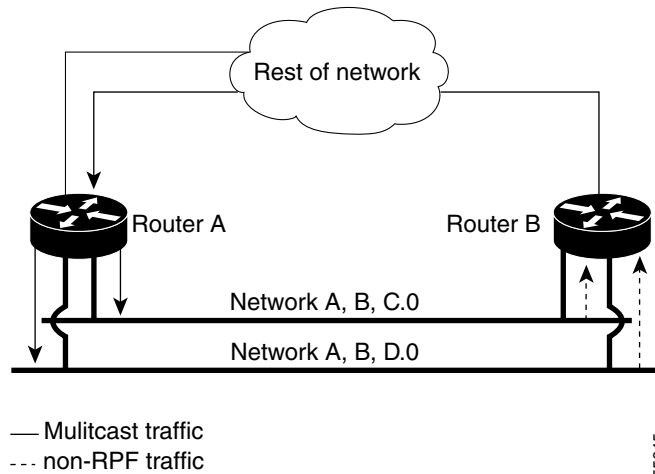
- [Non-RPF Traffic Overview, page 18-5](#)
- [Filtering of RPF Failures for Stub Networks, page 18-6](#)
- [Rate Limiting of RPF Failure Traffic, page 18-6](#)

Non-RPF Traffic Overview

In a redundant configuration where multiple routers connect to the same LAN segment, only one router forwards the multicast traffic from the source to the receivers on the outgoing interfaces (see [Figure 18-1](#)). In this kind of topology, only the PIM designated router (PIM DR) forwards the data in the common VLAN, but the non-PIM DR receives the forwarded multicast traffic. The redundant router (non-PIM DR) must drop this traffic because it has arrived on the wrong interface and fails the RPF check. Traffic that fails the RPF check is called non-RPF traffic.

The Catalyst 6500 series switch processes non-RPF traffic in hardware on the PFC by filtering (dropping) or rate limiting the non-RPF traffic.

Figure 18-1 Redundant Multicast Router Configuration in a Stub Network



Filtering of RPF Failures for Stub Networks

PFC1, PFC2, and the DFCs support ACL-based filtering of RPF failures for sparse mode stub networks. When you enable the ACL-based method of filtering RPF failures by entering the `mls ip multicast stub` command on the redundant router, the following ACLs automatically download to the PFC and are applied to the interface you specify:

```
access-list 100 permit ip A.B.C.0 0.0.0.255 any
access-list 100 permit ip A.B.D.0 0.0.0.255 any
access-list 100 permit ip any 224.0.0.0 0.0.0.255
access-list 100 permit ip any 224.0.1.0 0.0.0.255
access-list 100 deny ip any 224.0.0.0 15.255.255.255
```

The ACLs filter RPF failures and drop them in hardware so that they are not forwarded to the router.

Use the ACL-based method of filtering RPF failures only in sparse mode stub networks where there are no downstream routers. For dense mode groups, RPF failure packets have to be seen on the router for the PIM assert mechanism to function properly. Use CEF- or NetFlow-based rate limiting to rate-limit RPF failures in dense mode networks and sparse mode transit networks.

For information on configuring ACL-based filtering of RPF failures, see the “[Configuring ACL-Based Filtering of RPF Failures](#)” section on page 18-14.

Rate Limiting of RPF Failure Traffic

Rate limiting of packets that fail the RPF check (non-RPF packets) drops most non-RPF packets in hardware. According to the multicast protocol specification, the router needs to see the non-RPF packets for the PIM assert mechanism to work, so all non-RPF packets cannot be dropped in hardware. To support the PIM assert mechanism, the PFC leaks a percentage of the non-RPF flow packets to the MSFC.

These sections describe two modes of RPF failure rate limiting:

- [NetFlow-Based Rate Limiting of RPF Failures, page 18-7](#)
- [CEF-Based Rate Limiting of RPF Failures, page 18-7](#)



Note

PFC2 and the DFCs support both rate-limiting modes. CEF-based rate limiting of RPF failures is the default on systems with PFC2 and for DFCs. NetFlow-based rate limiting of RPF failures is the only rate limiting mode supported with PFC1.

NetFlow-Based Rate Limiting of RPF Failures

With NetFlow-based rate limiting of RPF failures, a NetFlow entry is created for each non-RPF flow. When a non-RPF packet arrives, the MSFC communicates information about the group, the source, and the interface on which the packet arrived to the PFC. The PFC then installs a NetFlow entry and bridges the packet to all ports in the VLAN, excluding the internal router port.

The PFC checks for non-RPF traffic every 2 seconds. An entry is kept for a maximum of 20 seconds if non-RPF traffic exists.

To configure NetFlow-based rate limiting of RPF failures, see the [“Enabling NetFlow-Based Rate Limiting of RPF Failures”](#) section on page 18-12.

CEF-Based Rate Limiting of RPF Failures

PFC2 and the DFCs support both CEF-based rate limiting of RPF failures and NetFlow-based rate limiting of RPF failures. In the CEF-based mode, the PFC2 or the DFC drops non-RPF packets instead of bridging them to the MSFC2. To support the PIM assert mechanism, CEF-based rate limiting works in 10-second intervals. For a short duration in each 10-second interval, packets are leaked to the MSFC. During the remainder of each 10-second interval, the non-RPF packets are dropped in hardware. CEF-based rate limiting of RPF failures is enabled by default on systems with PFC2 and on the DFCs and does not require any user configuration.

For information on configuring CEF-based rate limiting of RPF failures, see the [“Enabling CEF-Based Rate Limiting of RPF Failures”](#) section on page 18-13.

Default IP Multicast Layer 3 Switching Configuration

Table 18-1 shows the default IP multicast Layer 3 switching configuration.

Table 18-1 Default IP Multicast Layer 3 Switching Configuration

Feature	Default Value
ACL for stub networks	Disabled on all interfaces
Installing of directly connected subnet entries	Enabled globally
CEF-based rate limiting	Enabled globally (PFC2 only)
Netflow-based rate limiting	Disabled globally
Multicast routing	Disabled globally
PIM routing	Disabled on all interfaces
IP multicast Layer 3 switching	Enabled when multicast routing is enabled and PIM is enabled on the interface
Shortcut consistency checking	Enabled

Internet Group Management Protocol (IGMP) snooping is enabled by default on all VLAN interfaces. If you disable IGMP snooping on an interface, multicast Layer 3 flows are still hardware switched. Bridging of the flow on an interface with IGMP snooping disabled causes flooding to all forwarding interfaces of the VLAN. For details on configuring IGMP snooping, see [Chapter 21, “Configuring IGMP Snooping.”](#)

IP Multicast Layer 3 Switching Configuration Guidelines and Restrictions

These sections describe IP Multicast Layer 3 switching configuration restrictions:

- [PFC2 with MSCF2, page 18-8](#)
- [PFC1 with MSFC or MSCF2, page 18-8](#)
- [PFC1 and PFC2 General Restrictions, page 18-9](#)
- [Unsupported Features, page 18-9](#)

PFC2 with MSCF2

In systems with PCF2 and MSFC2, IP multicast Layer 3 switching is not provided for an IP multicast flow in the following situations:

- For IP multicast groups that fall into the range 224.0.0.* (where * is in the range 0 to 255), which is used by routing protocols. Layer 3 switching is supported for groups 225.0.0.* through 239.0.0.* and 224.128.0.* through 239.128.0.*.



Note Groups in the 224.0.0.* range are reserved for routing control packets and must be flooded to all forwarding ports of the VLAN. These addresses map to the multicast MAC address range 01-00-5E-00-00-xx, where xx is in the range 0–0xFF.

- For PIM auto-RP multicast groups (IP multicast group addresses 224.0.1.39 and 224.0.1.40).
- If the SPT bit for the flow is cleared when running PIM sparse mode for the interface or group.
- For packets with IP options. However, packets in the flow that do not specify IP options are hardware switched.
- For source traffic received on tunnel interfaces (such as MBONE traffic).

PFC1 with MSFC or MSCF2

In systems with PFC1 and MSFC or MSFC2, IP multicast Layer 3 switching is not provided for an IP multicast flow in the following situations:

- For IP multicast groups that fall into these ranges (where * is in the range 0 to 255):
224.0.0.* through 239.0.0.*
224.128.0.* through 239.128.0.*



Note Groups in the 224.0.0.* range are reserved for routing control packets and must be flooded to all forwarding interfaces of the VLAN. All these addresses map to the multicast MAC address range 01-00-5E-00-00-xx, where xx is in the range 0–0xFF.

- For PIM auto-RP multicast groups (IP multicast group addresses 224.0.1.39 and 224.0.1.40).
- For flows that are forwarded on the multicast-shared tree (that is, {*,G,*} forwarding) when the interface or group is running PIM sparse mode.

- If the SPT bit for the flow is cleared when running PIM sparse mode for the interface or group.
- For packets that require fragmentation and packets with IP options. However, packets in the flow that do not specify IP options are Layer 3 switched.
- For source traffic received on tunnel interfaces (such as MBONE traffic).

PFC1 and PFC2 General Restrictions

Input ACL deny is not applied by the hardware ACL engine when the Layer 2 entry corresponding to the Layer 3 flow does not exist in the Layer 2 forwarding table. The ACL will be applied by the MSFC software.

Unsupported Features

If you enable IP multicast Layer 3 switching, IP accounting for Layer 3 interfaces does not report accurate values. The **show ip accounting** command is not supported.

Configuring IP Multicast Layer 3 Switching

These sections describe how to configure IP multicast Layer 3 switching:

- [Source Specific Multicast with IGMPv3, IGMP v3lite, and URD, page 18-10](#)
- [Enabling IP Multicast Routing Globally, page 18-10](#)
- [Enabling IP PIM on Layer 3 Interfaces, page 18-10](#)
- [Enabling IP Multicast Layer 3 Switching on Layer 3 Interfaces, page 18-11](#)
- [Configuring the Layer 3 Switching Global Threshold, page 18-11](#)
- [Enabling Installation of Directly Connected Subnets, page 18-12](#)
- [Enabling NetFlow-Based Rate Limiting of RPF Failures, page 18-12](#)
- [Enabling CEF-Based Rate Limiting of RPF Failures, page 18-13](#)
- [Enabling Shortcut-Consistency Checking, page 18-13](#)
- [Configuring ACL-Based Filtering of RPF Failures, page 18-14](#)
- [Displaying RPF Failure Rate-Limiting Information, page 18-14](#)
- [Displaying IP Multicast Layer 3 Hardware Switching Summary, page 18-14](#)
- [Displaying the IP Multicast Routing Table, page 18-16](#)
- [Displaying IP Multicast Layer 3 Switching Statistics, page 18-17](#)
- [Using Debug Commands, page 18-18](#)
- [Clearing IP Multicast Layer 3 Switching Statistics, page 18-19](#)



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Source Specific Multicast with IGMPv3, IGMP v3lite, and URD

For complete information and procedures about source specific multicast with IGMPv3, IGMP v3lite, and URL Rendezvous Directory (URD), refer to this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtssm5t.htm>

Enabling IP Multicast Routing Globally

You must enable IP multicast routing globally before you can enable IP multicast Layer 3 switching on Layer 3 interfaces.

For complete information and procedures, refer to these publications:

- *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/index.htm
- *Cisco IOS IP and IP Routing Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/index.htm

To enable IP multicast routing globally, perform this task:

Command	Purpose
Router(config)# ip multicast-routing	Enables IP multicast routing globally.
Router(config)# no ip multicast-routing	Disables IP multicast routing globally.

This example shows how to enable multicast routing globally:

```
Router(config)# ip multicast-routing
Router(config)#
```

Enabling IP PIM on Layer 3 Interfaces

You must enable PIM on the Layer 3 interfaces before IP multicast Layer 3 switching functions on those interfaces.

To enable IP PIM on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port}}	Selects an interface to configure.
Step 2	Router(config-if)# ip pim {dense-mode sparse-mode sparse-dense-mode}	Enables IP PIM on a Layer 3 interface.
	Router(config-if)# no ip pim [dense-mode sparse-mode sparse-dense-mode]	Disables IP PIM on a Layer 3 interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable PIM on an interface using the default mode (**sparse-dense-mode**):

```
Router(config-if)# ip pim
Router(config-if)#
```


This example shows how to enable PIM sparse mode on an interface:

```
Router(config-if)# ip pim sparse-mode
Router(config-if)#
```

Enabling IP Multicast Layer 3 Switching on Layer 3 Interfaces

IP multicast Layer 3 switching is enabled by default on the Layer 3 interface when you enable PIM on the interface. Perform this task only if you disabled IP multicast Layer 3 switching on the interface and you want to reenabling it.

PIM can be enabled on any Layer 3 interface, including VLAN interfaces.



Note

You must enable PIM on all participating Layer 3 interfaces before IP multicast Layer 3 switching will function. For information on configuring PIM on Layer 3 interfaces, see the [“Enabling IP PIM on Layer 3 Interfaces”](#) section on page 18-10.

To enable IP multicast Layer 3 switching on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port}}	Selects an interface to configure.
Step 2	Router(config-if)# mls ip multicast	Enables IP multicast Layer 3 switching on a Layer 3 interface.
Step 3	Router(config-if)# no mls ip multicast	Disables IP multicast Layer 3 switching on a Layer 3 interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable IP multicast Layer 3 switching on a Layer 3 interface:

```
Router(config-if)# mls ip multicast
Router(config-if)#
```

Configuring the Layer 3 Switching Global Threshold

You can configure a global multicast rate threshold, specified in packets per second, below which all multicast traffic is routed by the MSFC, which prevents creation of switching cache entries for low-rate Layer 3 flows.



Note

This command does not affect flows that are already being routed. To apply the threshold to existing routes, clear the route and let it reestablish.

To configure the Layer 3 switching threshold, perform this task:

Command	Purpose
Router(config)# mls ip multicast threshold <i>ppsec</i>	Configures the IP MMLS threshold.
Router(config)# no mls ip multicast threshold	Reverts to the default IP MMLS threshold.

This example shows how to configure the Layer 3 switching threshold to 10 packets per second:

```
Router(config)# mls ip multicast threshold 10
Router(config)#
```

Enabling Installation of Directly Connected Subnets

In PIM sparse mode, a first-hop router that is the designated router for the interface may need to encapsulate the source traffic in a PIM register message and unicast it to the rendezvous point. To prevent new sources for the group from being learned in the routing table, the (*,G) flows should remain as completely hardware-switched flows. (subnet/mask, 224/4) entries installed in the hardware FIB allows both (*,G) flows to remain completely hardware-switched flows, and new, directly connected sources to be learned correctly. Installing of directly connected subnets is enabled globally by default. One (subnet/mask, 224/4) is installed per PIM-enabled interface.

To view FIB entries, enter the **show mls ip multicast connected** command.

To enable installation of directly connected subnets, perform this task:

Command	Purpose
Router(config)# mls ip multicast connected	Enables installation of directly connected subnets.
Router(config)# no mls ip multicast connected	Disables installation of directly connected subnets.

This example shows how to enable installation of directly connected subnets:

```
Router(config)# mls ip multicast connected
Router(config)#
```

Enabling NetFlow-Based Rate Limiting of RPF Failures

You can enable NetFlow-based rate limiting of RPF failures globally and on a per-Layer 3 interface basis. When enabled on a global level, the feature is automatically enabled on all eligible Layer 3 interfaces.



Note

To enable NetFlow-based rate limiting of RPF failures on a PFC2, you must first disable CEF-based rate limiting of RPF failures, which is enabled by default.

To enable NetFlow-based rate limiting of RPF failures, perform this task:

	Command	Purpose
Step 1	Router(config)# mls ip multicast non-rpf netflow	Enables NetFlow-based rate limiting of RPF failures globally.
	Router(config)# no mls ip multicast non-rpf netflow	Disables NetFlow-based rate limiting of RPF failures globally.
Step 2	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel channel_ID}}	Selects the Layer 3 interface to be configured.

	Command	Purpose
Step 3	Router(config-if)# mls ip multicast non-rpf netflow	Enables NetFlow-based rate limiting of RPF failures on the Layer 3 interface.
	Router(config-if)# no mls ip multicast non-rpf netflow	Disables NetFlow-based rate limiting of RPF failures on the Layer 3 interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable NetFlow-based rate limiting of non-RPF failures globally:

```
Router(config)# mls ip multicast non-rpf netflow
Router(config)#
```

Enabling CEF-Based Rate Limiting of RPF Failures

CEF-based rate limiting of RPF failures is enabled by default on systems with PFC2. CEF-based rate limiting of RPF failures can be configured globally only.

To enable CEF-based rate limiting of RPF failures, perform this task:

Command	Purpose
Router(config)# mls ip multicast non-rpf cef	Enables CEF-based rate limiting of RPF failures globally.
Router(config)# no mls ip multicast non-rpf cef	Disables CEF-based rate limiting of RPF failures globally.

This example shows how to enable CEF-based rate limiting of RPF failures globally:

```
Router(config)# mls ip multicast non-rpf CEF
Router(config)#
```

Enabling Shortcut-Consistency Checking

When you enable the shortcut-consistency checking feature, the multicast route table and the multicast-hardware entries are checked for consistency, and any inconsistencies are corrected. You can view inconsistencies by entering the **show mls ip multicast consistency-check** command.

If consistency checking is enabled, the multicast route table will be scanned every two seconds and a full scan is completed within 4 minutes.

To enable shortcut-consistency checking, perform this task:

Command	Purpose
Router(config)# mls ip multicast consistency-check	Enables shortcut-consistency checking.
Router(config)# no mls ip multicast consistency-check num	Restores the default.

This example shows how to enable the hardware shortcut-consistency checker:

```
Router (config)# mls ip multicast consistency-check
Router (config)#
```

Configuring ACL-Based Filtering of RPF Failures

When you configure ACL-based filtering of RPF failures, ACLs that filter RPF failures in hardware are downloaded to the hardware-based ACL engine and applied on the interface you specify.

To enable ACL-based filtering of RPF failures on an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects an interface to configure.
Step 2	Router(config-if)# mls ip multicast stub	Enables ACL-based filtering of RPF failures on an interface.
	Router(config-if)# no mls ip multicast stub	Disables ACL-based filtering of RPF failures on an interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

Displaying RPF Failure Rate-Limiting Information

To display RPF failure rate-limiting information, perform this task:

Command	Purpose
Router# show mls ip multicast summary	Displays RPF failure rate-limiting information.

This example shows how to display RPF failure rate-limiting information:

```
Router# show mls ip multicast summary
10004 MMLS entries using 1280464 bytes of memory
Number of partial hardware-switched flows:4
Number of complete hardware-switched flows:10000
Router#
```

Displaying IP Multicast Layer 3 Hardware Switching Summary



Note

The **show interface statistics** command does not display hardware-switched packets, only packets switched by software.

The **show ip pim interface count** command displays the IP multicast Layer 3 switching enable state on IP PIM interfaces and the number of packets received and sent on the interface.

To display IP multicast Layer 3 switching information for an IP PIM Layer 3 interface, perform one of these tasks:

Command	Purpose
Router# show ip pim interface [{{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}] count	Displays IP multicast Layer 3 switching enable state information for all MSFC IP PIM Layer 3 interfaces.
Router# show ip interface	Displays the IP multicast Layer 3 switching enable state on the Layer 3 interfaces.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

These examples show how to display the IP PIM configuration of the interfaces:

```
Router# show ip pim interface count
```

```
State:* - Fast Switched, D - Distributed Fast Switched
        H - Hardware Switching Enabled
Address      Interface          FS  Mpackets In/Out
10.15.1.20   GigabitEthernet4/8 * H 952/4237130770
10.20.1.7    GigabitEthernet4/9 * H 1385673757/34
10.25.1.7    GigabitEthernet4/10* H 0/34
10.11.1.30   FastEthernet6/26   * H 0/0
10.37.1.1    FastEthernet6/37   * H 0/0
1.22.33.44   FastEthernet6/47   * H 514/68
```

```
Router# show ip mroute count
```

```
IP Multicast Statistics
56 routes using 28552 bytes of memory
13 groups, 3.30 average sources per group
Forwarding Counts:Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
→ Other counts:Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group:224.2.136.89, Source count:1, Group pkt count:29051
  Source:132.206.72.28/32, Forwarding:29051/-278/1186/0, Other:85724/8/56665
Router#
```



Note

The -ive counter means that the outgoing interface list of the corresponding entry is NULL, and this indicates that this flow is still active.

This example shows how to display the IP multicast Layer 3 switching configuration of interface VLAN 10:

```
Router# show ip interface vlan 10
```

```
Vlan10 is up, line protocol is up
  Internet address is 10.0.0.6/8
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.2 224.0.0.13 224.0.0.10
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable are never sent
  ICMP mask replies are never sent
```

```

IP fast switching is enabled
IP fast switching on the same interface is disabled
IP Flow switching is disabled
IP CEF switching is enabled
IP Fast switching turbo vector
IP Normal CEF switching turbo vector
IP multicast fast switching is enabled
IP multicast distributed fast switching is disabled
IP route-cache flags are Fast, CEF
Router Discovery is disabled
IP output packet accounting is disabled
IP access violation accounting is disabled
TCP/IP header compression is disabled
RTP/IP header compression is disabled
Probe proxy name replies are disabled
Policy routing is disabled
Network address translation is disabled
WCCP Redirect outbound is disabled
WCCP Redirect exclude is disabled
BGP Policy Mapping is disabled
→ IP multicast multilayer switching is enabled
→ IP mls switching is enabled
Router#

```

Displaying the IP Multicast Routing Table

The **show ip mroute** command displays the IP multicast routing table.

To display the IP multicast routing table, perform this task:

Command	Purpose
Router# show ip mroute [<i>hostname</i> <i>group_number</i>]	Displays the IP multicast routing table and the hardware-switched interfaces.

This example shows how to display the IP multicast routing table:

```

Router# show ip mroute 230.13.13.1
IP Multicast Routing Table
Flags:D - Dense, S - Sparse, s - SSM Group, C - Connected, L - Local,
      P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
      J - Join SPT, M - MSDP created entry, X - Proxy Join Timer Running
      A - Advertised via MSDP, U - URD, I - Received Source Specific Host
      Report
Outgoing interface flags:H - Hardware switched
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode

(*, 230.13.13.1), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20
  Outgoing interface list:
→ GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(*, 230.13.13.2), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
→ Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(10.20.1.15, 230.13.13.1), 00:14:31/00:01:40, flags:CJT
→ Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD

```

```

Outgoing interface list:
  GigabitEthernet4/9, Forward/Sparse-Dense, 00:14:31/00:00:00, H
(132.206.72.28, 224.2.136.89), 00:14:31/00:01:40, flags:CJT
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
→ Outgoing interface list:Null
Router#

```

**Note**

The RPF-MFD flag indicates the flow is completely hardware switched. The H flag indicates the flow is hardware switched on the outgoing interface.

Displaying IP Multicast Layer 3 Switching Statistics

The **show mls ip multicast** command displays detailed information about IP multicast Layer 3 switching.

To display detailed IP multicast Layer 3 switching information, perform these tasks:

Command	Purpose
Router# show mls ip multicast group <i>ip_address</i> [interface <i>type slot/port</i> statistics]	Displays IP multicast Layer 3 switching group information.
Router# show mls ip multicast interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }} [statistics summary]	Displays IP multicast Layer 3 switching details for all interfaces.
Router# show mls ip multicast source <i>ip_address</i> [interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }}] statistics]	Displays IP multicast Layer 3 switching source information.
Router# show mls ip multicast summary	Displays a summary of IP multicast Layer 3 switching information.
Router# show mls ip multicast statistics	Displays IP multicast Layer 3 switching statistics.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display information on a specific IP multicast Layer 3 switching entry:

```

Router# show mls ip multicast group 10.1.0.11
Multicast hardware switched flows:
Total shortcut installed: 0

```

This example shows how to display IP multicast group information:

```

Router# show mls ip multicast group 230.13.13.1 source 10.20.1.15
Multicast hardware switched flows:
(10.20.1.15, 230.13.13.1) Incoming interface:Gi4/8, Packets switched:0
Hardware switched outgoing interfaces:Gi4/9
→ RPF-MFD installed

```

```

Total hardware switched flows :1
Router#

```

This example shows how to display IP multicast Layer 3 switching information for VLAN 10:

```

Router# show mls ip multicast interface vlan 10
Multicast hardware switched flows:
(10.1.0.15, 224.2.2.15) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
MFD installed: Vlan10

```

```
(10.1.0.19, 224.2.2.19) Incoming interface: Vlan10, Packets switched: 1970
Hardware switched outgoing interfaces:
MFD installed: Vlan10
```

```
(10.1.0.11, 224.2.2.11) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
MFD installed: Vlan10
```

```
(10.1.0.10, 224.2.2.10) Incoming interface: Vlan10, Packets switched: 2744
Hardware switched outgoing interfaces:
MFD installed: Vlan10
```

```
(10.1.0.17, 224.2.2.17) Incoming interface: Vlan10, Packets switched: 3340
Hardware switched outgoing interfaces:
MFD installed: Vlan10
```

```
(10.1.0.13, 224.2.2.13) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
```

This example shows how to display the IP multicast Layer 3 switching statistics:

```
Router# show mls ip multicast statistics

MLS Multicast Operation Status:
MLS Multicast configuration and state:
  Router Mac: 00e0.b0ff.7b00, Router IP: 33.0.33.24
  MLS multicast operating state: ACTIVE
  Shortcut Request Queue size 4
  Maximum number of allowed outstanding messages: 1
  Maximum size reached from feQ: 3096
  Feature Notification sent: 1
  Feature Notification Ack received: 1
  Unsolicited Feature Notification received: 0
  MSM sent: 205170
  MSM ACK received: 205170
  Delete notifications received: 0
  Flow Statistics messages received: 35211

MLS Multicast statistics:
  Flow install Ack: 996508
  Flow install Nack: 1
  Flow update Ack: 1415959
  Flow update Nack: 0
  Flow delete Ack: 774953
  Complete flow install Ack: 958469

Router#
```

Using Debug Commands

Table 18-2 describes IP multicast Layer 3 switching-related debug commands that you can use to troubleshoot IP multicast Layer 3 switching problems.

Table 18-2 IP Multicast Layer 3 Switching Debug Commands

Command	Description
[no] debug mls ip multicast events	Displays IP multicast Layer 3 switching events.
[no] debug mls ip multicast errors	Turns on debug messages for multicast MLS-related errors.

Table 18-2 IP Multicast Layer 3 Switching Debug Commands (continued)

Command	Description
[no] debug mls ip multicast group <i>group_id</i> <i>group_mask</i>	Turns on debugging for a subset of flows.
[no] debug mls ip multicast messages	Displays IP multicast Layer 3 switching messages from and to hardware switching engine.
[no] debug mls ip multicast all	Turns on all IP multicast Layer 3 switching messages.
[no] debug mdss errors	Turns on MDSS ¹ error messages.
[no] debug mdss events	Turns on MDSS-related events.
[no] debug mdss all	Turns on all MDSS messages.

1. MDSS = Multicast Distributed Switching Services

Clearing IP Multicast Layer 3 Switching Statistics

To clear IP multicast Layer 3 switching statistics, perform this task:

Command	Purpose
Router# clear mls ip multicast statistics	Clears IP multicast Layer 3 switching statistics.

This example shows how to clear IP multicast Layer 3 switching statistics:

```
Router# clear mls ip multicast statistics
```

The **show mls multicast statistics** command displays a variety of information about the multicast flows being handled by the PFC. You can display entries based on any combination of the participating MSFC, the VLAN, the multicast group address, or the multicast traffic source. For an example of the **show mls ip multicast statistics** command, see the [“Displaying IP Multicast Layer 3 Switching Statistics”](#) section on page 18-17.



Configuring IP Unicast Layer 3 Switching on Supervisor Engine 1



Note

The features described in this chapter are supported only on Supervisor Engine 1, the policy feature card (PFC), and the Multilayer Switch Feature Card (MSFC or MSFC2). For information about Supervisor Engine 2, PFC2, and MSFC2, see [Chapter 17, “Configuring IP Unicast Layer 3 Switching on Supervisor Engine 2.”](#)

Supervisor Engine 1 with PFC and MSFC or MSFC2 provide hardware Layer 3 switching with Multilayer Switching (MLS). This chapter describes how to configure IP unicast Layer 3 switching on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How IP MLS Works, page 19-2](#)
- [Default IP MLS Configuration, page 19-6](#)
- [IP MLS Configuration Guidelines and Restrictions, page 19-6](#)
- [Configuring IP MLS, page 19-6](#)
- [Displaying IP MLS Cache Entries, page 19-9](#)
- [Clearing IP MLS Cache Entries, page 19-11](#)
- [Troubleshooting IP MLS, page 19-14](#)



Note

To configure the MSFC to support MLS on a Catalyst 5000 series switch, refer to the *Layer 3 Switching Software Configuration Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_2/layer3/index.htm.

Understanding How IP MLS Works

These sections provide an overview of IP MLS and describe how IP MLS works:

- [IP MLS Overview, page 19-2](#)
- [IP MLS Flows, page 19-2](#)
- [Layer 3 MLS Cache, page 19-3](#)
- [Flow Masks, page 19-3](#)
- [Layer 3-Switched Packet Rewrite, page 19-4](#)
- [IP MLS Operation, page 19-5](#)

IP MLS Overview

IP MLS provides high-performance hardware-based Layer 3 switching for Catalyst 6500 series switches. IP MLS switches unicast IP data packet flows between IP subnets using advanced application-specific integrated circuit (ASIC) switching hardware, which offloads the processor-intensive packet routing from network routers.

The packet forwarding function is moved onto Layer 3 switches whenever a complete switched path exists between two hosts. Standard routing protocols, such as Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Routing Information Protocol (RIP), and Intermediate System-to-Intermediate System (IS-IS), are used for route determination.

In addition, IP MLS provides traffic statistics you can use to identify traffic characteristics for administration, planning, and troubleshooting. IP MLS uses NetFlow Data Export (NDE) to export flow statistics.

**Note**

For more information about NDE, see [Chapter 33, “Configuring NDE.”](#)

IP MLS Flows

Layer 3 protocols, such as IP and Internetwork Packet Exchange (IPX), are connectionless—they deliver every packet independently of every other packet. However, actual network traffic consists of many end-to-end conversations, or flows, between users or applications.

A flow is a unidirectional sequence of packets between a particular source and destination that share the same protocol and transport-layer information. Communication from a client to a server and from the server to the client are separate flows. For example, Telnet traffic transferred from a particular source to a particular destination comprises a separate flow from File Transfer Protocol (FTP) packets between the same source and destination.

Flows are based only on Layer 3 addresses, which allow IP traffic from multiple users or applications to a particular destination to be carried on a single flow if only the destination IP address is used to identify a flow.

Layer 3 MLS Cache

The PFC maintains a Layer 3 switching table (the Layer 3 MLS cache) for Layer 3-switched flows. The cache also includes entries for traffic statistics that are updated in tandem with the switching of packets. After the MLS cache is created, packets identified as belonging to an existing flow can be Layer 3 switched based on the cached information. The MLS cache maintains flow information for all active flows.

An MLS cache entry is created for the initial packet of each flow. Upon receipt of a packet that does not match any flow currently in the MLS cache, a new IP MLS entry is created.

The state and identity of the flow are maintained while packet traffic is active; when traffic for a flow ceases, the entry ages out. You can configure the aging time for MLS entries kept in the MLS cache. If an entry is not used for the specified period of time, the entry ages out and statistics for that flow can be exported to a flow collector application.

The maximum MLS cache size is 128K entries. However, an MLS cache larger than 32K entries increases the probability that a flow will not be switched by the PFC and will get forwarded to the Catalyst 6500 series switch.

Flow Masks

A flow mask is a filter configured by a network administrator that is used by the PFC to determine how MLS entries are created. The more detailed the flow-mask criteria, the deeper into the packet the MLS process must look in order to verify if the packet meets those criteria.

The PFC supports only one flow mask, and when the PFC flow mask changes, the entire MLS cache is purged. When the PFC exports cached entries, flow records are created based on the current flow mask.

Depending on the current flow mask, some fields in the flow record might not have values. Unsupported fields are filled with a zero (0).

There are three types of IP MLS flow-mask modes: destination-ip, source-destination-ip, and full-flow-ip. This section describes how these three flow-mask modes work.

- destination-ip—The least-specific flow mask. The PFC maintains one MLS entry for each destination IP address. All flows to a given destination IP address use this MLS entry. In destination-ip mode, the destination IP address of the switched flows are displayed, along with the packet rewrite information: rewritten destination MAC, rewritten VLAN, and egress interface.
- source-destination-ip—The PFC maintains one MLS entry for each source and destination IP address pair. All flows between a given source and destination use this MLS entry regardless of the protocol-specific Layer 4 port information.
- full-flow-ip—The most-specific flow mask. The PFC creates and maintains a separate MLS cache entry for each IP flow. A full-flow-ip entry includes the source IP address, destination IP address, protocol, and protocol-specific Layer 4 port information.

**Note**

The flow mask mode affects the screen output of the **show mls ip** command.

Interaction Between Software Features and Flow Mask Behavior

This section describes the flow mask used when different software features are configured in a system with a Supervisor Engine 1.

- Security ACLs—Does not affect flow mask.
- Reflexive ACLs—Does not affect flow mask.
- TCP intercept—Does not affect flow mask.
- Policy Based Routing (PBR)—Does not affect flow mask.
- ISLB (IOS Server Load Balancing)—When packets are processed by the ISLB process, a full-flow-ip mask is used.
- WCCP (Web Cache Control Protocol)—When packets are processed by WCCP, a full-flow-ip mask is used.



Note A full-flow-ip mask is used if the if the Web Cache engines are Layer-2 adjacent to the switch. If the Web Cache engines are not Layer-2 adjacent, then GRE encapsulation needs to be configured to send packets to the Web Cache engines and in that the flow mask is not affected because the packets are processed in software.

- CBAC (Context-Based Access Control)—Does not affect flow mask.
- Unicast RPF—When unicast RPF is configured with the **ip verify unicast** command, the flow mask is changed by the Layer 3 manager to source-destination-ip mask.
- Netflow Data export (NDE)—The flow mask used is determined by the **mls flow ip** command.
- QoS Microflow policing—When packets are processed by microflow policing, a full-flow-ip mask is used.

Layer 3-Switched Packet Rewrite

When a packet is Layer 3 switched from a source host to a destination host, the PFC performs a packet rewrite based on information learned from the MSFC and stored in the MLS cache.

If Host A and Host B are on different VLANs and Host A sends a packet to the MSFC to be routed to Host B, the PFC recognizes that the packet was sent to the MAC address of the MSFC. The PFC checks the MLS cache and finds the entry matching the flow in question.

When the PFC receives the packet, it is formatted (conceptually) as follows:

Frame Header		IP Header				Payload	
Destination	Source	Destination	Source	TTL	Checksum	Data	Checksum
<i>MSFC MAC</i>	<i>Host A MAC</i>	<i>Host B IP</i>	<i>Host A IP</i>	<i>n</i>	<i>calculation1</i>		

The PFC rewrites the Layer 2 frame header, changing the destination MAC address to the MAC address of Host B and the source MAC address to the MAC address of the MSFC (these MAC addresses are stored in the MLS cache entry for this flow). The Layer 3 IP addresses remain the same, but the IP header Time to Live (TTL) is decremented and the checksum is recomputed. The PFC rewrites the switched Layer 3 packets so that they appear to have been routed by a router.

The PFC forwards the rewritten packet to Host B's VLAN (the destination VLAN is stored in the MLS cache entry) and Host B receives the packet.

After the PFC performs the packet rewrite, the packet is formatted (conceptually) as follows:

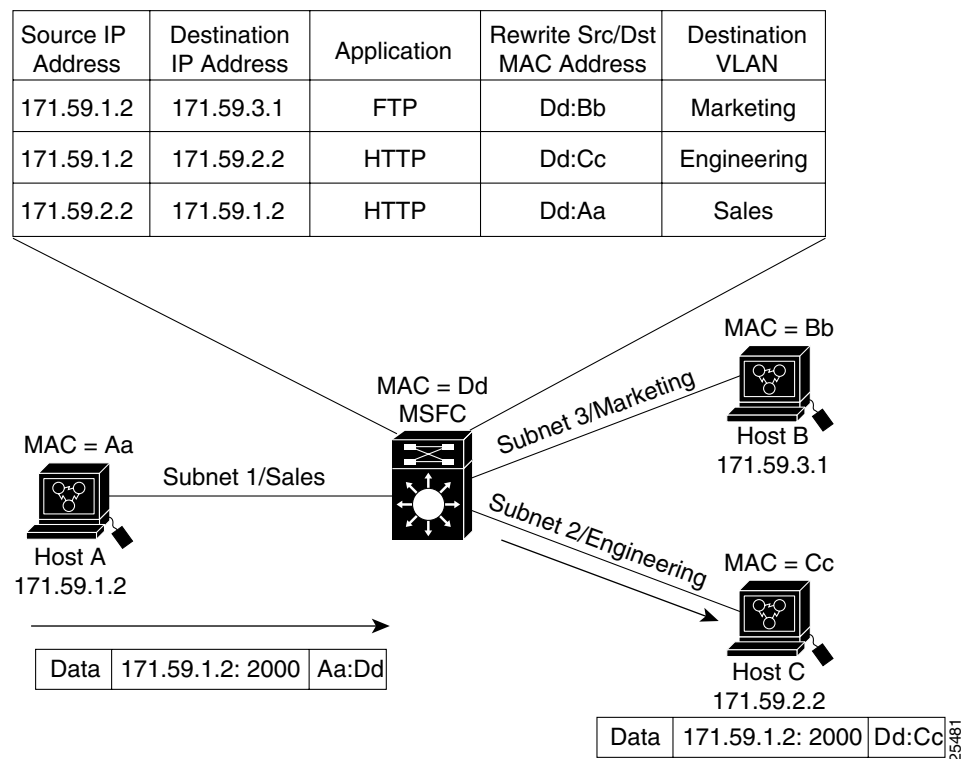
Frame Header		IP Header				Payload	
Destination	Source	Destination	Source	TTL	Checksum	Data	Checksum
<i>Host B MAC</i>	<i>MSFC MAC</i>	<i>Host B IP</i>	<i>Host A IP</i>	<i>n-1</i>	<i>calculation2</i>		

IP MLS Operation

Figure 19-1 shows a simple IP MLS network topology. In this example, Host A is on the Sales VLAN (IP subnet 171.59.1.0), Host B is on the Marketing VLAN (IP subnet 171.59.3.0), and Host C is on the Engineering VLAN (IP subnet 171.59.2.0).

When Host A initiates an HTTP file transfer to Host C, an MLS entry for this flow is created (this entry is the second item in the MLS cache shown in Figure 19-1). The PFC stores the MAC addresses of the MSFC and Host C in the MLS entry when the MSFC forwards the first packet from Host A through the switch to Host C. The PFC uses this information to rewrite subsequent packets from Host A to Host C.

Figure 19-1 IP MLS Example Topology



Default IP MLS Configuration

Table 19-1 shows the default IP MLS configuration.

Table 19-1 Default IP MLS Configuration

Feature	Default Value
IP MLS enable state	Enabled
IP MLS aging time	256 seconds
IP MLS fast aging time	32 seconds
IP MLS fast aging-time packet threshold	100 packets
IP MLS long aging time	900 seconds

IP MLS Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring IP MLS:

- The **clear ip route** command clears all IP MLS cache entries.
- The **no ip routing** command purges all IP MLS cache entries and disables IP MLS.
- The **ip security** interface command disables IP MLS on the interface.

Configuring IP MLS

These sections describe how to configure the MSFC for IP MLS:

- [Enabling IP MLS Globally, page 19-6](#)
- [Disabling and Enabling IP MLS on a Layer 3 Interface, page 19-7](#)
- [Displaying the Interface IP MLS Configuration, page 19-7](#)
- [Configuring the MLS Aging-Time, page 19-8](#)
- [Setting the Minimum IP MLS Flow Mask, page 19-8](#)



Note

The MSFC can be specified as the MLS route processor (MLS-RP) for Catalyst 5000 family switches using MLS. Refer to the *Layer 3 Switching Configuration Guide—Catalyst 5000 Family, 4000 Family, 2926G Series, 2926 Series, and 2948G* for procedures.



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling IP MLS Globally

IP MLS is enabled globally and cannot be disabled.

Disabling and Enabling IP MLS on a Layer 3 Interface

IP MLS is permanently enabled globally but can be disabled and enabled on a specified interface.

To enable IP MLS on a specific interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects an interface to configure.
Step 2	Router(config-if)# mls ip	Enables IP MLS on an interface.
	Router(config-if)# no mls ip	Disables IP MLS on an interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to disable IP MLS for Fast Ethernet port 5/5:

```
Router(config)# interface fastethernet 5/5
Router(config-if)# no mls ip
Router(config-if)#
```



Note

IP MLS is enabled by default; you only need to enable it if you have previously disabled it.

Displaying the Interface IP MLS Configuration

To display the IP MLS configuration on a Layer 3 interface, perform this task:

Command	Purpose
Router# show ip { interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}} nde }	Displays IP MLS configuration for an interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display the IP MLS configuration for Fast Ethernet port 5/4:

```
Router# show ip interface fastethernet 5/4
FastEthernet5/4 is up, line protocol is up
  Internet address is 172.20.52.106/29
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
```

```

IP Flow switching is disabled
IP CEF switching is enabled
IP Fast switching turbo vector
IP Normal CEF switching turbo vector
IP multicast fast switching is enabled
IP multicast distributed fast switching is disabled
Router Discovery is disabled
IP output packet accounting is disabled
IP access violation accounting is disabled
TCP/IP header compression is disabled
RTP/IP header compression is disabled
Probe proxy name replies are disabled
Policy routing is disabled
Network address translation is disabled
WCCP Redirect outbound is disabled
WCCP Redirect exclude is disabled
BGP Policy Mapping is disabled
IP multicast multilayer switching is disabled
IP mls switching is enabled
Router#

```

Configuring the MLS Aging-Time

The MLS aging time applies to all MLS cache entries. See the [“Configuring the MLS Aging Time” section on page 33-10](#).

Setting the Minimum IP MLS Flow Mask

You can set the minimum granularity of the flow mask for the MLS cache on the PFC. The actual flow mask used will be at least of the granularity specified by this command. For information on how the different flow masks work, see the [“Flow Masks” section on page 19-3](#).

For example, if you do not configure access lists on any MSFC, then the IP MLS flow mask on the PFC is destination-ip by default. However, you can force the PFC to use the source-destination-ip flow mask by setting the minimum IP MLS flow mask using the **mls flow destination-source** command.



Caution

Changing the flow mask purges all existing shortcuts in the MLS cache and affects the number of active shortcuts on the PFC. Be careful when using this command.

To set the minimum IP MLS flow mask, perform this task:

Command	Purpose
Router(config)# mls flow [ip { destination destination-source full }]	Sets the minimum IP MLS flow mask for the protocol.
Router(config)# no mls flow ip	Reverts to the default IP MLS flow mask.

This example shows how to set the minimum IP MLS flow mask:

```

Router(config)# mls flow ip destination
Router(config)#

```

To display the IP MLS flow mask configuration, perform this task:

Command	Purpose
Router# show mls netflow flowmask	With Release 12.1(8a)E and later releases, displays the flow mask configuration.
Router# show mls flowmask	With releases earlier than Release 12.1(8a)E, displays the flow mask configuration.

This example shows how to display the MLS flow mask configuration:

```
Router# show mls netflow flowmask
current ip flowmask for unicast: destination address
current ipx flowmask for unicast: destination address
Router#
```

Displaying IP MLS Cache Entries

These sections describe how to display IP MLS cache entries:

- [Displaying IP MLS Information, page 19-9](#)
- [Displaying IP MLS Cache Entries for a Specific Destination Address, page 19-10](#)
- [Displaying Cache Entries for a Specific Source IP Address, page 19-10](#)
- [Displaying Entries for a Specific IP Flow, page 19-11](#)



Note

For a description of how the flow mask mode affects the screen displays when showing MLS entries, see the “Flow Masks” section on page 19-3.

Displaying IP MLS Information

To display IP MLS information, perform this task:

Command	Purpose
Router# show mls ip [any destination <i>ip_address</i> detail flow [tcp udp] interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }} macd <i>destination_mac_address</i> macs <i>source_mac_address</i> multicast source <i>ip_address</i>]	Displays IP MLS information.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display IP MLS information:

```
Router# show mls ip
DstIP          SrcIP          DstVlan-DstMAC      Pkts      Bytes
-----
SrcDstPorts  SrcDstEncap  Age      LastSeen
-----
172.20.52.122  0.0.0.0      5       : 00e0.4fac.b3ff 155      6290
5 /9 ,5 /9  ARPA,ARPA  661     15:09:32

Number of Entries Found = 1

Router#
```

Displaying IP MLS Cache Entries for a Specific Destination Address

To display MLS entries for a specific destination IP address, perform this task:

Command	Purpose
Router# show mls ip destination <i>ip_address</i> [any detail flow [tcp udp] interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }} macd <i>destination_mac_address</i> macs <i>source_mac_address</i> multicast source <i>ip_address</i>]	Displays the IP MLS cache entries for a specific IP destination address.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display MLS entries for a specific destination IP address:

```
Router# show mls ip destination 127.1.1.1
DstIP/SrcIP    Prot/SrcPt/DstPt  DstMAC/DstVlan    Pkts  Bytes
-----
127.1.1.1/127.1.1.1  udp/  0040.0bd0.29fc/4095  92  111C
127.1.1.1/0.0.0.0    0040.0bd0.29fc/4095  0  0

Number of Entries Found = 2

Router#
```

Displaying Cache Entries for a Specific Source IP Address

To display MLS entries for a specific source IP address, perform this task:

Command	Purpose
Router# show mls ip source <i>ip_address</i> [any destination <i>ip_address</i> detail flow [tcp udp] interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }} macd <i>destination_mac_address</i> macs <i>source_mac_address</i> multicast]	Displays the IP MLS source cache entries for a specific IP source address.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display MLS entries for a specific source IP address:

```
Router# show mls ip source 172.20.52.122 any
DstIP          SrcIP          DstVlan-DstMAC      Pkts          Bytes
-----
SrcDstPorts   SrcDstEncap   Age      LastSeen
-----
172.20.52.122  0.0.0.0       5       : 00e0.4fac.b3ff 157          6370
5 /9 ,5 /9   ARPA,ARPA    901     15:15:30

Number of Entries Found = 1

Router#
```

Displaying Entries for a Specific IP Flow

To display MLS cache entries for a specific IP flow (when the flow mask mode is IP-flow), perform this task:

Command	Purpose
<pre>Router# show mls ip flow [tcp [any detail interface type slot/port macd destination_mac_address macs source_mac_address] udp [any detail interface {{vlan vlan_ID} {type¹ slot/port} {port-channel number}} macd destination_mac_address macs source_mac_address]]</pre>	Displays the IP MLS cache entries for a specific IP flow.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display MLS cache entries for a specific IP flow:

```
Router# show mls ip flow tcp detail
IP Destination  IP Source      Vlan Xtag L3-protocol Encapsulation RW-Vlan
-----+-----+-----+-----+-----+-----+-----+
RW-MACSource   RW-MACDestination  Bytes      Packets    Age  Last Seen
-----+-----+-----+-----+-----+-----+
QoS           Police Count Threshold  Leak      Drop Bucket  Use-Tbl Use-Enable
-----+-----+-----+-----+-----+-----+

Number of Entries Found = 0

Router#
```

Clearing IP MLS Cache Entries

The **clear mls ip** command removes specific MLS cache entries. If none of the following parameters are entered, all IP Layer 3 entries in the table are cleared:

- **destination** or **source**—Describes the IP addresses of the origin and termination point being purged.
- **interface** and its arguments—Limits the purge to entries for that interface.
- **macd** (MAC **destination**) or **macs** (MAC **source**)—Specifies the source or destination parameters to use when searching for entries to purge.
- **exclude protocol**—Specifies **all port**, **tcp port**, or **udp port** and interface to allow a entries to remain in the table.

- **slot**—Clears only the entries associated with a specific slot number.

The **flow** keyword specifies the following additional flow information:

- Protocol family (*protocol*)—Specifies **tcp** or **udp**.
- TCP or UDP source and destination port numbers—If the protocol you specify is Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), specify the source and destination TCP or UDP port numbers.

To clear an IP MLS cache entry, perform this task:

Command	Purpose
<pre>Router# clear mls [exclude protocol [all port 1-96 [tcp port 1-96 udp port 1-96]] ip [any destination ip_address flow [tcp [any interface macd macs] udp [any interface macd macs]] interface {{vlan vlan_ID} {type¹ slot/port} {port-channel number}} macd dest_mac_address macs source_mac_address multicast source]]</pre>	Clears MLS cache entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to clear MLS cache entries with destination IP address 172.20.26.22:

```
Router# clear mls ip destination 172.20.26.22
Router#
```

This example shows how to clear MLS cache entries with destination IP address 172.20.26.22, source 172.20.22.113, and flow TCP port 23:

```
Router# clear mls ip destination 172.20.26.22 source 172.20.22.113 flow tcp 23
Router#
```

To display the MLS entries and confirm they have been cleared see the [“Displaying IP MLS Cache Entries” section on page 19-9](#).

Displaying IP MLS Contention Table and Statistics

These sections describe how to display the MLS IP contention table and statistics:

- [Displaying the IP MLS Contention Table, page 19-12](#)
- [Displaying IP MLS VLAN Statistics, page 19-13](#)

Displaying the IP MLS Contention Table

The **show mls table-contention** command displays the flow contention level. The table contention level (TCL) is indicated with a number ranging from 0 (normal) to 3 (maximum). When reaching levels 1 through 3, accelerated aging starts, and begins to age out entries at a rate suitable to reduce the current contention rate. The detailed option displays the breakdown of contention between different flows.

To show the MLS contention table and VLAN statistics, perform this task:

Command	Purpose
Router# show mls table-contention [detailed summary]	Displays the MLS contention table.

This example shows how to display the MLS contention table:

```
Router# show mls table-contention detailed
Detailed Table Contention Level Information
=====
Layer 3
-----
L3 Contention Level:      0
Page Hits Requiring 1 Lookup   =      10
Page Hits Requiring 2 Lookups  =       0
Page Hits Requiring 3 Lookups  =       0
Page Hits Requiring 4 Lookups  =       0
Page Hits Requiring 5 Lookups  =       0
Page Hits Requiring 6 Lookups  =       0
Page Hits Requiring 7 Lookups  =       0
Page Hits Requiring 8 Lookups  =       0
Page Misses                   =       0

Router#
```

Displaying IP MLS VLAN Statistics

The **show mls vlan-statistics** command displays VLAN-based statistics for MLS cache entries. Specifying a VLAN identifier results in a display with only the shortcuts for that VLAN. If you specify a slot, only the information about that specific slot is shown; otherwise, all entries are displayed.

To display the MLS VLAN statistics, perform this task:

Command	Purpose
Router# show mls vlan-statistics 1-1024	Displays the MLS VLAN statistics.

This example shows how to display the VLAN statistics for VLAN 1 for every slot:

```
Router# show mls vlan-statistics 1
Slot 0
=====
Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 1
=====
Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 2
=====
```

```
Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 3
=====

(Information Deleted)

Router#
```

Troubleshooting IP MLS

Table 19-2 describes IP MLS-related debugging commands that you can use to troubleshoot IP MLS problems.

Table 19-2 IP MLS Debugging Commands

Command	Description
[no] <code>debugging 13-mgr events</code>	Displays Layer 3 manager-related events.
[no] <code>debugging 13-mgr packets</code>	Displays Layer 3 manager packets.
[no] <code>debugging 13-mgr global</code>	Displays bug trace of IP global purge events.
[no] <code>debugging 13-mgr all</code>	Turns on all Layer 3 manager debugging messages.

To configure the IP MLS-related debugging commands, perform this task:

Command	Purpose
Router# <code>debugging mls {ip {all error events messages multicast} rp {all error events ip locator packets verbose}}</code>	Configures IP-MLS debugging.
Router# <code>{no debugging undebug} {all {mls {ip {all error events messages multicast} rp {all error events ip locator packets verbose}}}}</code>	Disables IP-MLS debugging.

This example shows how to configure all IP debugging:

```
Router# debugging mls ip all
mls ip all debugging is on
Router#
```



Note

Enter the **show tech-support** command to display system information.



Configuring IPX Unicast Layer 3 Switching on Supervisor Engine 1



Note

The features described in this chapter are supported only on Supervisor Engine 1, PFC, and MSFC or MSFC2. For information about Supervisor Engine 2, PFC2, and MSFC2 see [Chapter 17, “Configuring IP Unicast Layer 3 Switching on Supervisor Engine 2.”](#)

Supervisor Engine 1 with PFC and MSFC or MSFC2 provide Layer 3 switching with Multilayer Switching (MLS). This chapter describes how to configure Internetwork Packet Exchange (IPX) Layer 3 switching on the Catalyst 6500 series switch.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How IPX MLS Works, page 20-2](#)
- [Default IPX MLS Configuration, page 20-5](#)
- [Configuration Guidelines and Restrictions, page 20-5](#)
- [Configuring IPX MLS, page 20-6](#)
- [Displaying IPX MLS Information, page 20-8](#)
- [Clearing IPX MLS Cache Entries, page 20-13](#)
- [Troubleshooting IPX MLS, page 20-14](#)



Note

The MSFC can be specified as the MLS route processor (MLS-RP) for Catalyst 5000 family switches using MLS. Refer to the *Layer 3 Switching Configuration Guide—Catalyst 5000 Family, 4000 Family, 2926G Series, 2926 Series, and 2948G* for MLS configuration procedures.

Understanding How IPX MLS Works

These sections provide an overview of MLS and describe how MLS works:

- [IPX MLS Overview, page 20-2](#)
- [IPX MLS Flows, page 20-2](#)
- [Layer 3 MLS Cache, page 20-2](#)
- [Flow Masks, page 20-3](#)
- [Layer 3-Switched Packet Rewrite, page 20-3](#)
- [IPX MLS Operation, page 20-4](#)

IPX MLS Overview

IPX MLS provides high-performance hardware-based Layer 3 switching for Catalyst 6500 series switches. IPX MLS switches unicast IPX data packet flows between networks using advanced application-specific integrated circuit (ASIC) switching hardware, offloading processor-intensive packet routing from network routers.

The packet forwarding function is moved onto Layer 3 switches whenever a partial or complete switched path exists between two hosts. Packets that do not have a partial or complete switched path to reach their destinations are still forwarded by routers. Standard routing protocols, such as Routing Information Protocol (RIP), Enhanced Interior Gateway Protocol (EIGRP), and NetWare Link Services Protocol (NLSP), are used for route determination.

IPX MLS Flows

Layer 3 protocols, such as IP and IPX, are connectionless—they deliver every packet independently of every other packet. However, actual network traffic consists of many end-to-end conversations, or flows, between users or applications.

A flow is a unidirectional sequence of packets between a particular source and destination that share the same protocol and network-layer information. Communication from a client to a server and from the server to the client are separate flows.

Flows are based only on Layer 3 addresses, which allow IPX traffic from multiple users or applications to a particular destination to be carried on a single flow if only the destination IPX address is used to identify a flow.

Layer 3 MLS Cache

The Policy Feature Card (PFC) maintains a Layer 3 switching table (MLS cache) for the Layer 3-switched flows. The cache includes entries for traffic statistics that are updated as packets are switched. After the MLS cache is created, packets identified as belonging to an existing flow can be Layer 3 switched based on the cached information. The MLS cache maintains flow information for all active flows.

An IPX MLS cache entry is created for the initial packet of each flow. Upon receipt of a packet that does not match any flow currently in the MLS cache, a new IPX MLS entry is created.

The state and identity of the flow are maintained while packet traffic is active; when traffic for a flow ceases, the entry ages out. You can configure the aging time for IPX MLS entries kept in the MLS cache. If an entry is not used for the specified period of time, the entry ages out and statistics for that flow can be exported to a flow collector application.

The maximum MLS cache size is 128K entries. However, an MLS cache larger than 32K entries increases the probability that a flow will not be switched by the PFC and will get forwarded to the MSFC.

Flow Masks

The PFC uses flow mask modes to determine how IPX MLS entries are created. The flow mask mode is based on the access lists configured on the IPX MLS router interfaces.

These sections describe how the flow mask modes work:

- [Flow Mask Modes, page 20-3](#)
- [Flow Mask Mode and show mls entry Command Output, page 20-3](#)

Flow Mask Modes

The PFC supports only one flow mask (the most specific one). When the PFC flow mask changes, the entire MLS cache is purged. When the PFC exports cached entries, flow records are created based on the current flow mask mode. Depending on the current mode, some fields in the flow record might not have values. Unsupported fields are filled with a dash (-).

The flow mask modes for IPX MLS are as follows:

- **destination mode**—The least-specific flow mask mode. The PFC maintains one IPX MLS entry for each destination IPX address (network and node). All flows to a given destination IPX address use this IPX MLS entry. This mode is used if there are no access lists configured based on source IPX addresses on any of the IPX MLS router interfaces.
- **destination-source mode**—The PFC maintains one MLS entry for each destination (network and node) and source (network only) IPX address pair. All flows between a given source and destination use this MLS entry regardless of the IPX sockets. This mode is used if there is an access list on any of the IPX MLS interfaces that filters on source network.

Flow Mask Mode and show mls entry Command Output

The flow mask mode impacts the screen output of the **show mls ipx** command. In destination mode, the destination IPX address of the switched flows are displayed, along with the packet rewrite information: rewritten destination MAC, rewritten VLAN, and egress interface.

Layer 3-Switched Packet Rewrite

When a packet is Layer 3 switched from a source host to a destination host, the PFC performs a packet rewrite, based on information learned from the MSFC and stored in the MLS cache.

If Host A and Host B are on different VLANs and Host A sends a packet to the MSFC to be routed to Host B, the PFC recognizes that the packet was sent to the MAC address of the MSFC. The PFC checks the MLS cache and finds the entry matching the flow in question.

Received IPX packets are formatted (conceptually) as follows:

Layer 2 Frame Header		Layer 3 IPX Header			Data	FCS
Destination	Source	Checksum/ IPX Length/ Transport Control	Destination Net/ Node/ Socket	Source Net/ Node/ Socket		
<i>MSFC MAC</i>	<i>Source A MAC</i>	<i>n</i>	<i>Destination B IPX</i>	<i>Source A IPX</i>		

The PFC rewrites the Layer 2 frame header, changing the destination MAC address to the MAC address of Host B and the source MAC address to the MAC address of the MSFC (these MAC addresses are stored in the IPX MLS cache entry for this flow). The Layer 3 IPX addresses remain the same. The PFC rewrites the switched Layer 3 packets so that they appear to have been routed by a router.

The PFC forwards the rewritten packet to Host B's VLAN (the destination VLAN is saved in the IPX MLS cache entry) and Host B receives the packet.

After the switch rewrites an IPX packet, it is formatted (conceptually) as follows:

Layer 2 Frame Header		Layer 3 IPX Header			Data	FCS
Destination	Source	Checksum/ IPX Length/ Transport Control	Destination Net/ Node/ Socket	Source Net/ Node/ Socket		
<i>Destination B MAC</i>	<i>MSFC MAC</i>	<i>n+1</i>	<i>Destination B IPX</i>	<i>Source A IPX</i>		

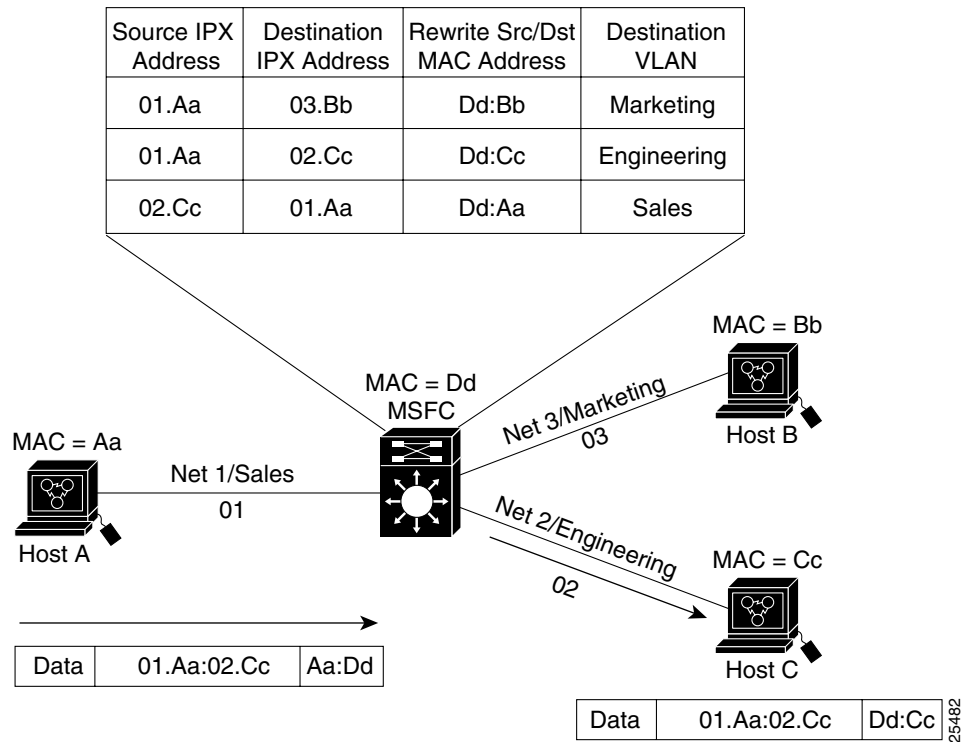
IPX MLS Operation

Figure 20-1 displays a conceptual IPX MLS network topology. In this example, Host A is on the Sales VLAN (IPX address 01.Aa), Host B is on the Marketing VLAN (IPX address 03.Bb), and Host C is on the Engineering VLAN (IPX address 02.Cc).

When Host A initiates a file transfer to Host C, an IPX MLS entry for this flow is created (this entry is the second item in the table shown in Figure 20-1). The PFC stores the MAC addresses of the MSFC and Host C in the IPX MLS entry when the MSFC forwards the first packet from Host A through the switch to Host C. The PFC uses this information to rewrite subsequent packets from Station A to Station C.

Similarly, a separate IPX MLS entry is created in the MLS cache for the traffic from Host A to Host B, and for the traffic from Host B to Host A. The destination VLAN is stored as part of each IPX MLS entry so that the correct VLAN identifier is used when encapsulating traffic on trunk links.

Figure 20-1 IPX MLS Example Topology



Default IPX MLS Configuration

Table 20-1 displays the default IPX MLS configuration.

Table 20-1 Default IPX MLS Configuration

Feature	Default Value
IPX MLS enable state	Enabled
IPX MLS aging time	256 seconds
IPX MLS fast aging time	32 seconds
IPX MLS fast aging time packet threshold	100 packets
IPX MLS long aging time	900 seconds

Configuration Guidelines and Restrictions

- These Cisco IOS software features and commands affect IPX MLS:
 - IPX accounting—IPX accounting cannot be enabled on an IPX MLS-enabled interface.
 - IPX EIGRP—MLS is supported for EIGRP interfaces if the Transport Control (TC) maximum is set to a value greater than the default (16).
 - The **clear ipx route** command clears all IPX MLS cache entries.

- The **no ipx routing** command purges all IPX MLS cache entries and disables IPX MLS.
- The **ipx security** interface command disables IPX MLS on the interface.
- In IPX, the two end points of communication negotiate the maximum transmission unit (MTU) to be used. MTU size is limited by media type.

Configuring IPX MLS

These sections describe how to configure IPX MLS:

- [Enabling IPX MLS Globally, page 20-6](#)
- [Enabling IPX MLS on a Layer 3 Interface, page 20-6](#)
- [Configuring the MLS Aging Time, page 20-7](#)
- [Configuring the Minimum IPX MLS Flow Mask, page 20-8](#)



Note

For information on configuring VLANs on the switch, see [Chapter 7, “Configuring LAN Ports for Layer 2 Switching.”](#)



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling IPX MLS Globally

IPX MLS is enabled globally and cannot be disabled.

Enabling IPX MLS on a Layer 3 Interface

To enable IPX MLS on an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# mls ipx	Enables IPX MLS.
	Router(config-if)# no mls ipx	Disables IPX MLS.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable IPX MLS for Fast Ethernet interface 5/5:

```
Router(config)# interface fastethernet 5/5
Router(config-if)# mls ipx
Router(config-if)#
```

To display the IPX MLS interface configuration, perform this task:

Command	Purpose
<code>Router# show [ipx [interface {{vlan vlan_ID} {type¹ slot/port} {port-channel number}}] nde]</code>	Displays MLS details for an interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display interface IPX MLS information for interface VLAN 200:

```
Router# show ipx interface vlan 200
Vlan200 is up, line protocol is up
  IPX address is 2.0050.3e8d.6400, NOVELL-ETHER [up]
  Delay of this IPX network, in ticks is 1 throughput 0 link delay 0
  IPXWAN processing not enabled on this interface.
  IPX SAP update interval is 60 seconds
  IPX type 20 propagation packet forwarding is disabled
  Incoming access list is not set
  Outgoing access list is not set
  IPX helper access list is not set
  SAP GGS output filter list is not set
  SAP GNS processing enabled, delay 0 ms, output filter list is not set
  SAP Input filter list is not set
  SAP Output filter list is not set
  SAP Router filter list is not set
  Input filter list is not set
  Output filter list is not set
  Router filter list is not set
  Netbios Input host access list is not set
  Netbios Input bytes access list is not set
  Netbios Output host access list is not set
  Netbios Output bytes access list is not set
  Updates each 60 seconds aging multiples RIP: 3 SAP: 3
  SAP interpacket delay is 55 ms, maximum size is 480 bytes
  RIP interpacket delay is 55 ms, maximum size is 432 bytes
  RIP response delay is not set
  IPX accounting is disabled
  IPX fast switching is configured (enabled)
  RIP packets received 0, RIP packets sent 1, 0 Throttled
  RIP specific requests received 0, RIP specific replies sent 0
  RIP general requests received 0, 0 ignored, RIP general replies sent 0
  SAP packets received 0, SAP packets sent 1, 0 Throttled
  SAP GNS packets received 0, SAP GNS replies sent 0
  SAP GGS packets received 0, 0 ignored, SAP GGS replies sent 0
  IPX mls switching is enabled
Router#
```

Configuring the MLS Aging Time

The MLS aging time applies to all MLS cache entries. See the [“Configuring the MLS Aging Time” section on page 33-10](#).



Note

IPX MLS does not use fast aging.

Configuring the Minimum IPX MLS Flow Mask

You can configure the minimum granularity of the flow mask for the MLS cache on the PFC. The actual flow mask used will be at least of the granularity specified by this command. For information on how the different flow masks work, see the [“Flow Masks” section on page 20-3](#).



Caution

This command purges all existing shortcuts in the MLS cache and affects the number of active shortcuts on the PFC. Be careful when using this command.

To configure the minimum IPX MLS flow mask, perform this task:

Command	Purpose
Router(config)# mls flow ipx {destination destination-source}	Configures the minimum IPX MLS flow mask.
Router(config)# no mls flow ipx	Reverts to the default IPX MLS flow mask.

This example displays how to configure the minimum IPX MLS flow mask to destination:

```
Router(config)# mls flow ipx destination
Router(config)#
```

To display the IPX MLS flow mask configuration, perform this task:

Command	Purpose
Router# show mls flowmask	Displays the flow mask configuration.

This example displays the MLS flow mask configuration:

```
Router# show mls flowmask
current ip flowmask for unicast: destination only
current ipx flowmask for unicast: destination only
```

Displaying IPX MLS Information

This section describes the commands used to display IPX MLS configuration and statistics for the switch and the various interfaces and is separated into the following:

- [Displaying IPX MLS Cache Entries, page 20-9](#)
- [Displaying the IPX MLS Contention Table, page 20-11](#)
- [Displaying IPX MLS VLAN Statistics, page 20-12](#)

Displaying IPX MLS Cache Entries

The **show mls ipx** command displays IPX shortcut cache entries. You can specify the following parameters to focus the information displayed:

- **source** and **destination** parameters display the source and or destination IPX network addresses associated with those entries.
- **interface** arguments display only entries associated with a specific interface number.
- **slot** displays only the cache entries associated with a specific slot number.
- **macs** and **macd** arguments and the IPX network address display the source and destination MAC addresses associated with those entries.

To display all IPX MLS entries on the switch, perform this task:

Command	Purpose
Router# show mls ipx [destination <i>ipx_network_address</i> interface {{ <i>vlan vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }} macd <i>destination_mac_address</i> macs <i>source_mac_address</i> source <i>ipx_network_address</i>]	Displays various IPX MLS cache entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

These sections provide examples of how to display specific IPX MLS cache entries on the switch:

- [Displaying All IPX MLS Cache Entries, page 20-9](#)
- [Displaying IPX MLS Cache Entries for a Specific Destination Address, page 20-10](#)
- [Displaying IPX MLS Cache Entries for a Specific Source Address, page 20-10](#)
- [Displaying IPX MLS Cache Entries for a Specific Interface, page 20-11](#)
- [Displaying IPX MLS Cache Entries for a Specific MAC Destination or Source Address, page 20-11](#)

Displaying All IPX MLS Cache Entries

To display all IPX MLS cache entries on the switch, perform this task:

Command	Purpose
Router# show mls ipx	Displays all IPX MLS entries.

This example shows how to display all IPX MLS entries on the switch:

```
Router# show mls ipx
DstNet-DstNode          SrcNet  DstVlan-DstMac      Pkts      Bytes
-----
SrcDstPorts   SrcDstEncap Age   LastSeen
-----
Number of Entries Found = 0

Router#
```

Displaying IPX MLS Cache Entries for a Specific Destination Address

To display IPX MLS cache entries for a specific destination IPX address, perform this task:

Command	Purpose
Router# show mls ipx [destination <i>ipx_addr</i>]	Displays IPX MLS entries for a specific destination address (<i>net_address.node_address</i>).

This example shows how to display IPX MLS entries for a specific destination address:

```
Router# show mls ipx destination 1.2.2.2
DstNet-DstNode          SrcNet  DstVlan-DstMac      Pkts      Bytes
-----
SrcDstPorts  SrcDstEncap Age  LastSeen
-----

Number of Entries Found = 0

Router#
```

Displaying IPX MLS Cache Entries for a Specific Source Address

To display IPX MLS cache entries for a specific source network address, perform this task:

Command	Purpose
Router# show mls ipx source <i>ipx_address</i>	Displays IPX MLS entries for a specific source network address (<i>net_address</i>).



Note

This task should be performed with IPX flow in destination-source mode. For more information, see the [“Flow Mask Modes”](#) section on page 20-3.

This example shows how to display IPX MLS entries for a specific source IPX address:

```
Router# show mls ipx source 1.2.2.2
DstNet-DstNode          SrcNet  DstVlan-DstMac      Pkts      Bytes
-----
SrcDstPorts  SrcDstEncap Age  LastSeen
-----

Number of Entries Found = 0

Router#
```

Displaying IPX MLS Cache Entries for a Specific Interface

To display IPX MLS entries for a specific interface, perform this task:

Command	Purpose
Router# show mls ipx interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Displays IPX MLS cache entries for a specific interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display IPX MLS entries for IPX Fast Ethernet interface 5/4:

```
Router# show mls ipx interface fastethernet 5/4
DstNet-DstNode          SrcNet  DstVlan-DstMac      Pkts      Bytes
-----
SrcDstPorts   SrcDstEncap Age   LastSeen
-----

Number of Entries Found = 0

Router#
```

Displaying IPX MLS Cache Entries for a Specific MAC Destination or Source Address

To display IPX MLS entries for a specific MAC destination or source address, perform this task:

Command	Purpose
Router# show mls ipx [macd destination_address macs source_address]	Displays IPX MLS cache entries for a specific destination or source MAC address.

This example shows how to display IPX MLS entries for a specific MAC destination address:

```
Router# show mls ipx macd aaaa.bbbb.bbbb
DstNet-DstNode          SrcNet  DstVlan-DstMac      Pkts      Bytes
-----
SrcDstPorts   SrcDstEncap Age   LastSeen
-----

Number of Entries Found = 0

Router#
```

Displaying the IPX MLS Contention Table

The **show mls table-contention** command displays the flow contention level for the switch. The table contention level (TCL) is indicated with a number ranging from 0 (normal) to 3 (maximum). When reaching levels 1 through 3, accelerated aging starts, which begins to age out entries at a rate suitable to reduce the current contention rate. The detailed option displays the breakdown of contention between different flows.

To show the MLS contention table and VLAN statistics for the switch, perform this task:

Command	Purpose
Router# show mls table-contention [detailed summary]	Displays the IPX MLS contention table.

This example displays the IPX MLS contention table for the switch:

```
Router# show mls table-contention detailed
Detailed Table Contention Level Information
=====
Layer 3
-----
L3 Contention Level:      0
Page Hits Requiring 1 Lookup   =      10
Page Hits Requiring 2 Lookups  =       0
Page Hits Requiring 3 Lookups  =       0
Page Hits Requiring 4 Lookups  =       0
Page Hits Requiring 5 Lookups  =       0
Page Hits Requiring 6 Lookups  =       0
Page Hits Requiring 7 Lookups  =       0
Page Hits Requiring 8 Lookups  =       0
Page Misses                   =       0

Router#
```

Displaying IPX MLS VLAN Statistics

The **show mls vlan-statistics** command displays VLAN-based statistics for IPX MLS cache entries. Specifying a VLAN identifier results in a display with only the shortcuts for that VLAN. If you specify a slot, only the information about that slot is shown; otherwise, all entries are displayed.

To display the IPX MLS VLAN statistics for the switch, perform this task:

Command	Purpose
Router# show mls vlan-statistics 1-1024	Displays the IPX MLS VLAN statistics.

This example displays the VLAN statistics for VLAN 1 for every slot in the switch:

```
Router# show mls vlan-statistics 1
Slot 0
=====
Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 1
=====
Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 2
=====
```

```

Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 3
=====
Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 4
=====
Vlan 1 Statistics Information:
-----
65280 Layer 2 Packets Bridged, 0 Bytes
65280 Layer 3 Packets Input, 0 Bytes
65280 Layer 3 Packets Output, 0 Bytes
Slot 5

(Information Deleted)

```

Clearing IPX MLS Cache Entries

Clear IPX shortcut entries in the Layer 3 table based on the entered criteria. The **clear mls ipx** command clears shortcut entries in the Layer 3 tables matching configured parameters. If none of the following parameters are entered, all IPX Layer 3 entries in the table are cleared:

- **destination** or **source**—Describes the IPX addresses of the origin and termination points being purged.
- **interface** and its arguments must be specified, which limits the purge to entries associated with the specified interface.
- **macd** (MAC **destination**) or **macs** (MAC **source**)—Specifies the source port or destination interface arguments to consider when searching for entries to purge.
- **slot**—Clears only the entries associated with a specific slot number.

To clear the IPX MLS statistics, perform this task:

Command	Purpose
<pre>Router# clear mls ipx [exclude protocol [all port 1-96 tcp port 1-96 udp port 1-96] [destination [hostname ipx_address] interface {{vlan vlan_ID} {type¹ slot/port} {port-channel number}} macd destination_mac_address macs source_mac_address]]</pre>	Clears the IPX MLS statistics.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to clear the MLS cache entries in the Layer 3 table IPX interface, Fast Ethernet interface 5/5:

```
Router# clear mls ipx interface fastethernet 5/5
Router#
```

To display the MLS entries and confirm they have been cleared, see the [“Displaying IPX MLS Information”](#) section on page 20-8.

Troubleshooting IPX MLS

Table 20-2 describes debug commands that you can use to troubleshoot IPX MLS problems.

Table 20-2 IPX MLS Debug Commands

Command	Description
[no] debug l3-mgr events	Displays Layer 3 manager-related events.
[no] debug l3-mgr packets	Displays Layer 3 manager packets.
[no] debug l3-mgr global	Displays bug trace of IP global purge events.
[no] debug l3-mgr all	Turns on all Layer 3 manager debugging messages.
[no] debug mls ipx	Turns on IPX-related events for MLS, including route purging and changes of access lists and flow masks.
[no] debug mls locator	Identifies which switch is switching a particular flow by using MLS explorer packets.
[no] debug mls all	Turns on all MLS debugging events.

To configure the debug commands that you can use to troubleshoot IPX MLS problems, perform this task:

Command	Purpose
Router(config)# debug mls {ipx {all error events messages} rp {all error events ip ipx locator packets verbose}}	Configures IPX MLS debugging.
Router(config)# {no undebug} mls {all ipx {all error events messages} rp {all error events ipx locator packets verbose}}	Disables MLS debugging.

This example displays how to configure all IPX debugging:

```
Router# debug mls ipx all
mls ip all debugging is on
Router#
```



Note

The **show tech-support** command displays switch system information. Use application-specific commands to get more information about particular applications.

Table 20-3 describes the Serial Control Protocol (SCP)-related debug commands to troubleshoot the SCP that runs over the Ethernet out-of-band channel (EOBC).

Table 20-3 SCP Debug Commands

Command	Description
[no] debug scp async	Displays trace for async data in and out of the SCP system.
[no] debug scp data	Displays packet data trace.
[no] debug scp errors	Displays errors and warnings in SCP.
[no] debug scp packets	Displays packet data in and out of the SCP system.
[no] debug scp timeouts	Reports timeouts.
[no] debug scp all	Turns on all SCP debugging messages.



Configuring IGMP Snooping

This chapter describes how to configure Internet Group Management Protocol (IGMP) snooping on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How IGMP Snooping Works, page 21-1](#)
- [Default IGMP Snooping Configuration, page 21-6](#)
- [IGMP Snooping and IGMP Snooping Querier Configuration Guidelines and Restrictions, page 21-6](#)
- [Enabling the IGMP Snooping Querier, page 21-7](#)
- [Configuring IGMP Snooping, page 21-8](#)



Note

-
- To support Cisco Group Management Protocol (CGMP) client devices, configure the Multilayer Switch Feature Card (MSFC) as a CGMP server. Refer to the *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.1, “IP Multicast,” “Configuring IP Multicast Routing,” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt3/1cdmulti.htm
 - For more information on IP multicast and IGMP, refer to RFC 1112 and RFC 2236.
-

Understanding How IGMP Snooping Works

These sections describe IGMP snooping:

- [IGMP Snooping Overview, page 21-2](#)
- [Joining a Multicast Group, page 21-2](#)
- [Leaving a Multicast Group, page 21-4](#)
- [Understanding IGMP Version 3 Support, page 21-6](#)

IGMP Snooping Overview

You can configure the switch to use IGMP snooping in subnets that receive IGMP queries from either IGMP or the IGMP snooping querier. IGMP snooping constrains multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward multicast traffic only to those ports that want to receive it.

IGMP, which runs at Layer 3 on a multicast router, generates Layer 3 IGMP queries in subnets where the multicast traffic needs to be routed. For information about IGMP, see [Chapter 18, “Configuring IP Multicast Layer 3 Switching.”](#)

With Release 12.1(8a)E and later releases, you can configure the IGMP snooping querier on the switch to support IGMP snooping in subnets that do not have any multicast router interfaces because the multicast traffic does not need to be routed. For more information about the IGMP snooping querier, see the [“Enabling the IGMP Snooping Querier” section on page 21-7.](#)

IGMP (on a multicast router) or the IGMP snooping querier (on the supervisor engine) sends out periodic general IGMP queries that the switch forwards through all ports in the VLAN and to which hosts respond. IGMP snooping monitors the Layer 3 IGMP traffic.

**Note**

If a multicast group has only sources and no receivers in a VLAN, IGMP snooping constrains the multicast traffic to only the multicast router ports.

Joining a Multicast Group

Hosts join multicast groups either by sending an unsolicited IGMP join message or by sending an IGMP join message in response to a general query from a multicast router (the switch forwards general queries from multicast routers to all ports in a VLAN).

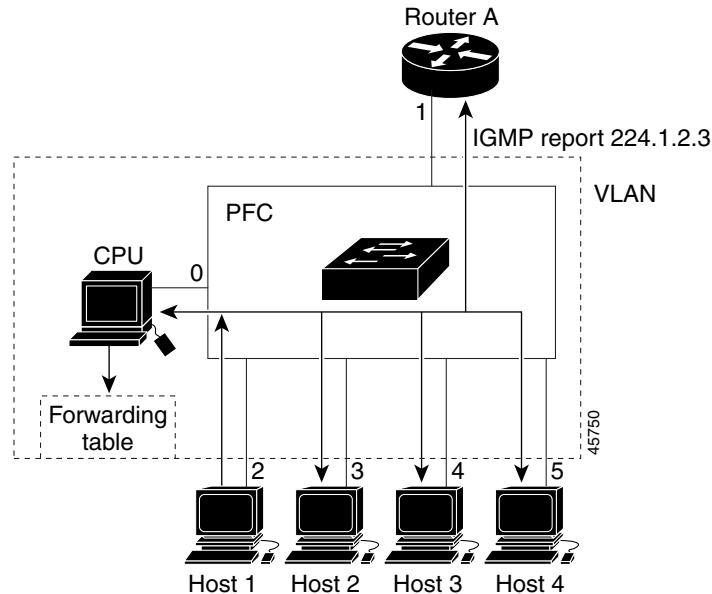
In response to an IGMP join request, the switch creates an entry in its Layer 2 forwarding table for the VLAN on which the join request was received. When other hosts interested in this multicast traffic send IGMP join requests, the switch adds them to the existing Layer 2 forwarding table entry. The switch creates only one entry per VLAN in the Layer 2 forwarding table for each multicast group for which it receives an IGMP join request.

IGMP snooping suppresses all but one of the host join messages per multicast group and forwards this one join message to the multicast router.

The switch forwards multicast traffic for the multicast group specified in the join message to the interfaces where join messages were received. See [Figure 21-1.](#)

Layer 2 multicast groups learned through IGMP snooping are dynamic. However, you can statically configure Layer 2 multicast groups using the **mac-address-table static** command. When you specify group membership for a multicast group address statically, the static setting supersedes any IGMP snooping learning. Multicast group membership lists can consist of both static and IGMP snooping-learned settings.

Figure 21-1 Initial IGMP Join Message



Multicast router A sends a general query to the switch, which forwards the query to ports 2 through 5, all members of the same VLAN. Host 1 wants to join multicast group 224.1.2.3 and multicasts an IGMP membership report (IGMP join message) to the group with the equivalent MAC destination address of 0x0100.5E01.0203. When the CPU receives the IGMP report multicast by Host 1, the CPU uses the information in the IGMP report to set up a forwarding-table entry, as shown in [Table 21-1](#), that includes the port numbers of Host 1, the multicast router, and the switch internal CPU.

Table 21-1 IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1, 2

The switch hardware can distinguish IGMP information packets from other packets for the multicast group. The first entry in the table tells the switching engine to send only IGMP packets to the CPU. This prevents the CPU from becoming overloaded with multicast frames. The second entry tells the switching engine to send frames addressed to the 0x0100.5E01.0203 multicast MAC address that are not IGMP packets (!IGMP) to the multicast router and to the host that has joined the group.

If another host (for example, Host 4) sends an unsolicited IGMP join message for the same group ([Figure 21-2](#)), the CPU receives that message and adds the port number of Host 4 to the forwarding table as shown in [Table 21-2](#). Because the forwarding table directs IGMP messages only to the CPU, the message is not flooded to other ports. Any known multicast traffic is forwarded to the group and not to the CPU.

Figure 21-2 Second Host Joining a Multicast Group

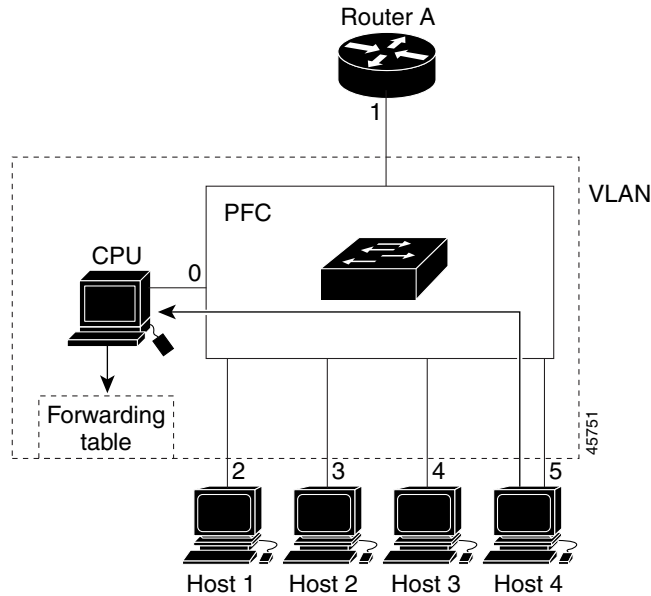


Table 21-2 Updated IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1, 2, 5

Leaving a Multicast Group

These sections describe leaving a multicast group:

- [Normal Leave Processing, page 21-4](#)
- [Fast-Leave Processing, page 21-5](#)

Normal Leave Processing

Interested hosts must continue to respond to the periodic general IGMP queries. As long as at least one host in the VLAN responds to the periodic general IGMP queries, the multicast router continues forwarding the multicast traffic to the VLAN. When hosts want to leave a multicast group, they can either ignore the periodic general IGMP queries (called a “silent leave”), or they can send a group-specific IGMPv2 leave message.

When IGMP snooping receives a group-specific IGMPv2 leave message from a host, it sends out a MAC-based a group-specific query to determine if any other devices connected to that interface are interested in traffic for the specific multicast group. If IGMP snooping does not receive an IGMP Join message in response to the general query, it assumes that no other devices connected to the interface are interested in receiving traffic for this multicast group, and it removes the interface from its Layer 2 forwarding table entry for that multicast group. If the leave message was from the only remaining interface with hosts interested in the group and IGMP snooping does not receive an IGMP Join in

response to the general query, it removes the group entry and relays the IGMP leave to the multicast router. If the multicast router receives no reports from a VLAN, the multicast router removes the group for the VLAN from its IGMP cache.

The interval for which the switch waits before updating the table entry is called the “last member query interval.” Enter the **ip igmp snooping last-member-query-interval** *interval* command to configure the interval.

Fast-Leave Processing

IGMP snooping fast-leave processing allows IGMP snooping to remove a Layer 2 LAN interface from the forwarding-table entry without first sending out IGMP group-specific queries to the interface. Upon receiving a group-specific IGMPv2 leave message, IGMP snooping immediately removes the interface from the Layer 2 forwarding table entry for that multicast group, unless a multicast router was learned on the port. Fast-leave processing improves bandwidth management for all hosts on a switched network.



Note

Use fast-leave processing only on VLANs where only one host is connected to each Layer 2 LAN port. If fast-leave is enabled in VLANs where more than one host is connected to a Layer 2 LAN port, some hosts might be dropped inadvertently. Fast-leave processing is supported only with IGMP version 2 hosts.

Understanding IGMP Snooping Querier

IGMP snooping querier should be used to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not need to be routed.

In a network with IP multicast routing, the IP multicast router acts as the IGMP querier. If the IP-multicast traffic in a VLAN needs to be Layer 2 switched only, an IP-multicast router is not required, but without an IP-multicast router on a VLAN, you must configure another switch as the IGMP querier so that it can send queries.

When IGMP snooping querier is enabled, the IGMP snooping querier sends out periodic IGMP queries that trigger IGMP report messages from the switch that wants to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

You can enable the IGMP snooping querier on all the Catalyst 6500 series switches in the VLAN, but for each VLAN that is connected to switches that use IGMP to report interest in IP multicast traffic, you must set at least one switch as the IGMP snooping querier.

You can use Cisco IOS commands to configure the Catalyst 6500 series switches to generate such IGMP queries on a VLAN regardless of whether or not IP multicast routing is enabled.



Note

To enable IP multicast routing on the Catalyst 6500 series switches on a specific VLAN, enter the **ip pim sparse-mode** command, the **ip pim sparse-dense-mode** command, or the **ip pim dense-mode** command on that interface. See [Chapter 18, “Configuring IP Multicast Layer 3 Switching”](#) for more details.

Understanding IGMP Version 3 Support

With Release 12.1(8a)E and later releases, IGMP snooping supports IGMP version 3. Because the Layer 2 table is (MAC-group, VLAN) based, with IGMPv3 hosts it is preferable to have only a single multicast source per MAC-group. A single multicast source per group allows IGMPv3 hosts connected to specific ports to receive traffic from a specific (source, group).

Default IGMP Snooping Configuration

Table 21-3 shows the default IGMP snooping configuration.

Table 21-3 IGMP Snooping Default Configuration

Feature	Default Values
IGMP snooping querier ¹	Disabled
IGMP snooping	Enabled
Multicast routers	None configured
IGMP snooping learning method	PIM/DVMRP ²

1. Supported in Release 12.1(8a)E and later releases.

2. PIM/DVMRP = Protocol Independent Multicast/Distance Vector Multicast Routing Protocol

IGMP Snooping and IGMP Snooping Querier Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring IGMP snooping and IGMP snooping querier:

Guidelines

- IGMP snooping supports private VLANs. Private VLANs do not impose any restrictions on IGMP snooping.
- IGMP snooping constrains traffic in MAC multicast groups 0100.5e00.0001 to 0100.5eff.ffff.
- IGMP snooping does not constrain Layer 2 multicasts generated by routing protocols.
- The IGMP snooping querier supports IGMP version 2.
- When enabled, the IGMP snooping querier does not start if it detects IGMP traffic from a multicast router.
- When enabled, the IGMP snooping querier starts after 60 seconds with no IGMP traffic detected from a multicast router.
- When enabled, the IGMP snooping querier disables itself if it detects IGMP traffic from a multicast router.
- You can enable the IGMP snooping querier on all the Catalyst 6500 series switches in the VLAN. On each VLAN that is connected to switches that use IGMP to report interest in IP multicast traffic, you must set one switch as the IGMP querier.

Periodically, the IGMP querier sends IGMP queries that trigger IGMP report messages from the switch that wants to receive IP multicast traffic.

IGMP snooping listens to these IGMP reports to establish appropriate forwarding. In a normal network with IP multicast routing, the IP multicast router acts as the IGMP querier.



Note To enable IP multicast routing on the Catalyst 6500 series switches on a specific VLAN, enter the **ip pim sparse-mode** command, the **ip pim sparse-dense-mode** command, or the **ip pim dense-mode** command on that interface. See [Chapter 18, “Configuring IP Multicast Layer 3 Switching”](#) for more details.

- If the IP-multicast traffic in a VLAN needs to be Layer 2 switched only, an IP-multicast router is not required, but without an IP-multicast router on a VLAN, you must configure another switch as the IGMP querier so that it can send queries. You can use Cisco IOS commands to configure the Catalyst 6500 series switches to generate such IGMP queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Restrictions

- IGMP snooping querier requires Release 12.1(8a)E and later.
- When configuring the IGMP snooping querier, configure the VLAN in the VLAN database or, with Release 12.1(11b)E and later releases, configure the VLAN in global configuration mode (see [Chapter 9, “Configuring VLANs”](#)).
- QoS does not support IGMP packets when IGMP snooping is enabled.
- You must configure an IP address on the VLAN interface for the IGMP snooping querier to start. (See [Chapter 12, “Configuring Layer 3 Interfaces”](#)). When enabled, the IGMP snooping querier uses the IP address as the query source address. The IGMP snooping querier disables itself if the IP address is cleared and restarts when you configure an IP address.



Note With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling the IGMP Snooping Querier

With Release 12.1(8a)E and later, use the IGMP snooping querier to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not need to be routed.

To enable the IGMP snooping querier in a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects the VLAN interface.
Step 2	Router(config-if)# ip igmp snooping querier	Enables the IGMP snooping querier.
	Router(config-if)# no ip igmp snooping querier	Disables the IGMP snooping querier.

	Command	Purpose
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show ip igmp interface vlan <i>vlan_ID</i> include querier	Verifies the configuration.

This example shows how to enable the IGMP snooping querier on VLAN 200 and verify the configuration:

```
Router# interface vlan 200
Router(config-if)# igmp snooping querier
Router(config-if)# end
Router# show ip igmp interface vlan 200 | include querier
IGMP snooping querier is enabled on this interface
Router#
```

Configuring IGMP Snooping



Note

To use IGMP snooping, configure a Layer 3 interface in the subnet for multicast routing (see [Chapter 18, “Configuring IP Multicast Layer 3 Switching”](#)) or enable the IGMP snooping querier in the subnet (see the “[Enabling the IGMP Snooping Querier](#)” section on page 21-7).

IGMP snooping allows Catalyst 6500 series switches to examine IGMP packets and make forwarding decisions based on their content.

These sections describe how to configure IGMP snooping:

- [Enabling IGMP Snooping, page 21-9](#)
- [Configuring IGMP Snooping Learning, page 21-10](#)
- [Configuring a Multicast Router Port Statically, page 21-10](#)
- [Configuring the IGMP Query Interval, page 21-11](#)
- [Enabling IGMP Fast-Leave Processing, page 21-11](#)
- [Configuring a Host Statically, page 21-12](#)
- [Displaying IGMP Snooping Information, page 21-12](#)



Note

Except for the global enable command, all IGMP snooping commands are supported only on VLAN interfaces.

Enabling IGMP Snooping

To enable IGMP snooping globally, perform this task:

	Command	Purpose
Step 1	Router(config)# ip igmp snooping	Enables IGMP snooping.
	Router(config)# no ip igmp snooping	Disables IGMP snooping.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show ip igmp interface vlan vlan_ID include globally	Verifies the configuration.

This example shows how to enable IGMP snooping globally and verify the configuration:

```
Router(config)# ip igmp snooping
Router(config)# end
Router# show ip igmp interface vlan 200 | include globally
IGMP snooping is globally enabled
Router#
```

To enable IGMP snooping in a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan vlan_ID	Selects a VLAN interface.
Step 2	Router(config-if)# ip igmp snooping	Enables IGMP snooping.
	Router(config-if)# no ip igmp snooping	Disables IGMP snooping.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show ip igmp interface vlan vlan_ID include snooping	Verifies the configuration.

This example shows how to enable IGMP snooping on VLAN 200 and verify the configuration:

```
Router# interface vlan 200
Router(config-if)# ip igmp snooping
Router(config-if)# end
Router# show ip igmp interface vlan 200 | include snooping
IGMP snooping is globally enabled
IGMP snooping is enabled on this interface
IGMP snooping fast-leave is enabled on this interface
IGMP snooping querier is disabled on this interface
Router#
```

Configuring IGMP Snooping Learning

To configure IGMP snooping learning, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects a VLAN interface.
Step 2	Router(config-if)# ip igmp snooping mrouter learn { cgmp pim-dvmrp }	Configures the learning method.
	Router(config-if)# no ip igmp snooping mrouter learn { cgmp pim-dvmrp }	Reverts to the default learning method.

This example shows how to configure IP IGMP snooping to learn from PIM/DVMRP packets:

```
Router(config)# interface vlan 1
Router(config-if)# ip igmp snooping mrouter learn pim-dvmrp
Router(config-if)# end
Router#
```

This example shows how to configure IP IGMP snooping to learn from CGMP self-join packets:

```
Router(config)# interface vlan 1
Router(config-if)# ip igmp snooping mrouter learn cgmp
Router(config-if)# end
Router#
```

Configuring a Multicast Router Port Statically

To configure a static connection to a multicast router, perform this task:

	Command	Purpose
Step 1	Router(config)# mac-address-table static <i>mac_addr</i> vlan <i>vlan_id</i> interface <i>type</i> ¹ <i>slot/port</i> [disable-snooping]	Configures a static connection to a multicast router.
	Router(config)# no mac-address-table static <i>mac_addr</i> vlan <i>vlan_id</i>	Clears a static connection to a multicast router.
Step 2	Router(config-if)# end	Exits configuration mode.
Step 3	Router# show mac-address-table address <i>mac_addr</i>	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

With Release 12.1(11b)E2 and later releases, you can enter the **disable-snooping** keyword to prevent multicast traffic addressed to the statically configured multicast MAC address from also being sent to other multicast router ports in the same VLAN.

This example shows how to configure a static connection to a multicast router:

```
Router(config)# mac-address-table static 0050.3e8d.6400 vlan 12 interface fastethernet 5/7
```

All releases support the **mac-address-table static** command. The **ip igmp snooping mrouter interface** command, which was available in earlier releases and which provided the same functionality as the **mac-address-table static** command, is deprecated in Release 12.1(13)E and later releases.

Configuring the IGMP Query Interval

You can configure the interval for which the switch waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group.



Note

When both IGMP fast-leave processing and the IGMP query interval are configured, fast-leave processing takes precedence.

To configure the interval for the IGMP queries sent by the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects a VLAN interface.
Step 2	Router(config-if)# ip igmp snooping last-member-query-interval <i>interval</i>	Configures the interval for the IGMP queries sent by the switch. Default is 1 second. Valid range is 100 to 999 milliseconds.
	Router(config-if)# no ip igmp snooping last-member-query-interval	Reverts to the default value.

This example shows how to configure the IGMP query interval:

```
Router(config-if)# ip igmp snooping last-member-query-interval 200
Router(config-if)# exit
Router# show ip igmp interface vlan 200 | include last-member-query-interval
IGMP snooping last member query interval on this interface is 200 ms
```

Enabling IGMP Fast-Leave Processing

To enable IGMP fast-leave processing in a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects a VLAN interface.
Step 2	Router(config-if)# ip igmp snooping fast-leave	Enables IGMP fast-leave processing in the VLAN.
	Router(config-if)# no ip igmp snooping fast-leave	Disables IGMP fast-leave processing in the VLAN.

This example shows how to enable IGMP fast-leave processing on the VLAN 200 interface and verify the configuration:

```
Router# interface vlan 200
Router(config-if)# ip igmp snooping fast-leave
Configuring fast leave on vlan 200
Router(config-if)# end
Router# show ip igmp interface vlan 200 | include fast-leave
IGMP snooping fast-leave is enabled on this interface
Router(config-if)#
```

Configuring a Host Statically

Hosts normally join multicast groups dynamically, but you can also configure a host statically for a Layer 2 LAN port.

To configure a host statically for a Layer 2 LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects a VLAN interface.
Step 2	Router(config)# mac-address-table static <i>mac_addr</i> vlan <i>vlan_id</i> interface <i>type</i> ¹ <i>slot/port</i> [disable-snooping]	Configures a static connection to a multicast router.
	Router(config)# no mac-address-table static <i>mac_addr</i> vlan <i>vlan_id</i>	Clears a static connection to a multicast router.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

With Release 12.1(11b)E2 and later releases, you can enter the **disable-snooping** keyword to prevent multicast traffic addressed to the statically configured multicast MAC address from being sent to multicast router ports in the same VLAN.

This example shows how to configure a host statically in VLAN 12 on FastEthernet port 5/7:

```
Router(config)# mac-address-table static 0050.3e8d.6400 vlan 12 interface fastethernet 5/7
```

All releases support the **mac-address-table static** command. The **ip igmp snooping static** command, which was available in earlier releases and which provided the same functionality as the **mac-address-table static** command, is deprecated in Release 12.1(13)E and later releases.

Displaying IGMP Snooping Information

These sections describe displaying IGMP snooping information:

- [Displaying Multicast Router Interfaces](#), page 21-12
- [Displaying MAC Address Multicast Entries](#), page 21-13
- [Displaying IGMP Snooping Information for a VLAN Interface](#), page 21-13

Displaying Multicast Router Interfaces

When you enable IGMP snooping, the switch automatically learns to which interface multicast routers are connected.

To display multicast router interfaces, perform this task:

Command	Purpose
Router# show ip igmp snooping mrouter interface <i>vlan_ID</i>	Displays multicast router interfaces.

This example shows how to display the multicast router interfaces in VLAN 1:

```
Router# show ip igmp snooping mrouter interface vlan 1
vlan          ports
-----+-----
 1           Gi1/1,Gi2/1,Fa3/48,Router
Router#
```

Displaying MAC Address Multicast Entries

To display MAC address multicast entries for a VLAN, perform this task:

Command	Purpose
Router# show mac-address-table multicast <i>vlan_ID</i> [count]	Displays MAC address multicast entries for a VLAN.

This example shows how to display MAC address multicast entries for VLAN 1:

```
Router# show mac-address-table multicast vlan 1
vlan  mac address      type    qos          ports
-----+-----
 1  0100.5e02.0203  static  --  Gi1/1,Gi2/1,Fa3/48,Router
 1  0100.5e00.0127  static  --  Gi1/1,Gi2/1,Fa3/48,Router
 1  0100.5e00.0128  static  --  Gi1/1,Gi2/1,Fa3/48,Router
 1  0100.5e00.0001  static  --  Gi1/1,Gi2/1,Fa3/48,Router,Switch
Router#
```

This example shows how to display a total count of MAC address entries for a VLAN:

```
Router# show mac-address-table multicast 1 count

Multicast MAC Entries for vlan 1:    4
Router#
```

Displaying IGMP Snooping Information for a VLAN Interface

To display IGMP snooping information for a VLAN interface, perform this task:

Command	Purpose
Router# show ip igmp interface <i>vlan_ID</i>	Displays IGMP snooping information on a VLAN interface.

This example shows how to display IGMP snooping information on the VLAN 200 interface:

```
Router# show ip igmp interface vlan 200
Vlan200 is up, line protocol is up
 Internet address is 172.20.52.94/27
 IGMP is enabled on interface
 Current IGMP version is 2
 CGMP is disabled on interface
 IGMP query interval is 60 seconds
 IGMP querier timeout is 120 seconds
 IGMP max query response time is 10 seconds
 Last member query response interval is 1000 ms
 Inbound IGMP access group is not set
 IGMP activity: 0 joins, 0 leaves
 Multicast routing is enabled on interface
```

```
Multicast TTL threshold is 0
Multicast designated router (DR) is 172.20.52.94 (this system)
IGMP querying router is 172.20.52.94 (this system)
No multicast groups joined
IGMP snooping is globally enabled
IGMP snooping is enabled on this interface
IGMP snooping fast-leave is enabled on this interface
IGMP snooping querier is disabled on this interface
Router#
```



Configuring RGMP

This chapter describes how to configure Router-Port Group Management Protocol (RGMP). Release 12.1(3a)E3 and later releases support RGMP. This chapter consists of these sections:

- [Understanding How RGMP Works, page 22-1](#)
- [Default RGMP Configuration, page 22-2](#)
- [RGMP Configuration Guidelines and Restrictions, page 22-2](#)
- [Enabling RGMP on Layer 3 Interfaces, page 22-3](#)



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

Understanding How RGMP Works

RGMP constrains multicast traffic that exits the Catalyst 6500 series switch through ports to which only disinterested multicast routers are connected. RGMP reduces network congestion by forwarding multicast traffic to only those routers that are configured to receive it.



Note

To use RGMP, you must enable IGMP snooping on the Catalyst 6500 series switch. IGMP snooping constrains multicast traffic that exits through LAN ports to which hosts are connected. IGMP snooping does not constrain traffic that exits through LAN ports to which one or more multicast routers are connected.



Note

You must enable Protocol Independent Multicast (PIM) on all routers and switches for RGMP to work. Only PIM sparse mode is currently supported.

All routers on the network must be RGMP-capable. RGMP-capable routers send RGMP hello messages periodically. The RGMP hello message tells the Catalyst 6500 series switch not to send multicast data to the router unless an RGMP join message has also been sent to the Catalyst 6500 series switch from that router. When an RGMP join message is sent, the router is able to receive multicast data.

To stop receiving multicast data, a router must send an RGMP leave message to the Catalyst 6500 series switch. To disable RGMP on a router, the router must send an RGMP bye message to the Catalyst 6500 series switch.

Table 22-1 provides a summary of the RGMP packet types.

Table 22-1 RGMP Packet Types

Description	Action
Hello	When RGMP is enabled on the router, no multicast data traffic is sent to the router by the Catalyst 6500 series switch unless an RGMP join is specifically sent for a group.
Bye	When RGMP is disabled on the router, all multicast data traffic is sent to the router by the Catalyst 6500 series switch.
Join	Multicast data traffic for a multicast MAC address from the Layer 3 group address G is sent to the router. These packets have group G in the Group Address field of the RGMP packet.
Leave	Multicast data traffic for the group G is not sent to the router. These packets have group G in the group address field of the RGMP packet.

Default RGMP Configuration

RGMP is permanently enabled on Layer 2 LAN ports. RGMP is disabled by default on Layer 3 interfaces.

RGMP Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring RGMP:

- RGMP supports PIM sparse mode. RGMP does not support PIM dense mode. RGMP explicitly supports the two AutoRP groups in dense mode by not restricting traffic to those groups but by flooding it to all router ports. For this reason, you should configure PIM sparse-dense mode. If you configure groups other than the AutoRP groups for dense mode, their traffic will not be correctly forwarded through router ports that have been enabled for RGMP.
- To effectively constrain multicast traffic with RGMP, connect RGMP-enabled routers to separate ports on RGMP-enabled Catalyst 6500 series switches. (VLAN interfaces satisfy this restriction.)
- RGMP only constrains traffic that exits through LAN ports on which it detects an RGMP-enabled router. If a non-RGMP enabled router is detected on a LAN port, that port receives all multicast traffic.
- RGMP does not support directly connected multicast sources in the network. A directly connected multicast source will send multicast traffic into the network without signaling through RGMP or PIM. This multicast traffic will not be received by an RGMP-enabled router unless the router already requested receipt of that multicast group through RGMP. This restriction applies to hosts and to functions in routers that source multicast traffic, such as the **ping** and **mtrace** commands and multicast applications that source multicast traffic, such as UDPTN.
- RGMP supports directly connected receivers in the network. Traffic to these receivers will be constrained by IGMP snooping, or if the receiver is a router itself, by PIM and RGMP.
- CGMP is not supported in networks where RGMP is enabled on routers. You cannot enable both RGMP and CGMP on a Layer 3 interface. If RGMP is enabled on a Layer 3 interface, CGMP is silently disabled and vice versa.

- The following properties of RGMP are the same as for IGMP snooping:
 - RGMP constrains traffic based on the multicast group, not on the sender’s IP address.
 - If spanning tree topology changes occur in the network, the state is not flushed as it is with Cisco Group Management Protocol (CGMP).
 - RGMP does not constrain traffic for multicast groups 224.0.0.x (x = 0...255), which allows use of PIMv2 bootstrap router (BSR) in an RGMP-controlled network.
 - RGMP in Cisco network devices operates on MAC addresses, not on IP multicast addresses. Because multiple IP multicast addresses can map to one MAC address (see RFC 1112), RGMP cannot differentiate between the IP multicast groups that might map to a MAC address.
 - The capability of the Catalyst 6500 series switch to constrain traffic is limited by its content-addressable memory (CAM) table capacity.

**Note**

With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling RGMP on Layer 3 Interfaces

To enable RGMP on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects an interface to configure.
Step 2	Router(config-if)# ip rgmp Router(config-if)# no ip rgmp	Enables RGMP on the Layer 3 interface. Disables RGMP on the Layer 3 interface.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# debug ip rgmp [name_or_group_address]	(Optional) Monitors RGMP.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

This example shows how to configure RGMP on FastEthernet port 3/3:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 3/3
Router(config-if)# ip rgmp
Router(config-if)# end
Router#
```




Configuring Network Security

This chapter contains network security information unique to the Catalyst 6500 series switches, which supplements the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/index.htm
- *Cisco IOS Security Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

This chapter consists of these sections:

- [ACL Configuration Guidelines](#), page 23-1
- [Hardware and Software ACL Support](#), page 23-2
- [Guidelines and Restrictions for Using Layer 4 Operators in ACLs](#), page 23-3
- [Configuring the Cisco IOS Firewall Feature Set](#), page 23-5
- [Configuring MAC Address-Based Traffic Blocking](#), page 23-8
- [Configuring VLAN ACLs](#), page 23-8
- [Configuring TCP Intercept](#), page 23-18
- [Configuring Unicast Reverse Path Forwarding](#), page 23-19
- [Configuring Unicast Flood Protection](#), page 23-21
- [Configuring MAC Move Notification](#), page 23-22



Note

With Releases 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

ACL Configuration Guidelines

The following guidelines apply to ACL configurations:

- Each type of ACL (IP, IPX, and MAC) filters only traffic of the corresponding type. A MAC ACL never matches IP or IPX traffic.
- By default, the MSFC sends Internet Control Message Protocol (ICMP) unreachable messages when a packet is denied by an access group.

With the **ip unreachable** command enabled (which is the default), a Supervisor Engine 2 drops most of the denied packets in hardware and sends only a small number of packets to the MSFC2 to be dropped (10 packets per second, maximum), which generates ICMP-unreachable messages.

With the **ip unreachable** command enabled, a Supervisor Engine 1 sends all the denied packets to the MSFC to be dropped, which generates ICMP-unreachable messages. With a Supervisor Engine 1, to drop access list-denied packets in hardware, you must disable ICMP-unreachable messages using the **no ip unreachable** interface configuration command.

To eliminate the load imposed on the MSFC CPU by the task of dropping denied packets and generating ICMP-unreachable messages, do the following:

- With Supervisor Engine 1, enter the **no ip unreachable** interface configuration command.
- With Supervisor Engine 2, enter the **no ip unreachable** and the **no ip redirects** interface configuration commands. (CSCdr33918)
- ICMP unreachable messages are not sent if a packet is denied by a VACL.

Hardware and Software ACL Support

Access control lists (ACLs) can be processed in hardware by the Policy Feature Card (PFC or PFC2), the Distributed Forwarding Card (DFC), or in software by the Multilayer Switch Feature Card (MSFC or MSFC2). The following behavior describes software and hardware handling of ACLs:

- ACL flows that match a “deny” statement in standard and extended ACLs (input and output) are dropped in hardware if “ip unreachable” is disabled.
- ACL flows that match a “permit” statement in standard and extended ACLs (input and output) are processed in hardware.
- VLAN ACL (VACL) flows are processed in hardware. If a field specified in a VACL is not supported by hardware processing that field is ignored (for example, the **log** keyword in an ACL) or the whole configuration is rejected (for example, a VACL containing unsupported IPX ACL parameters).
- VACL logging is processed in software.
- Dynamic ACL flows are processed in the hardware; however, idle timeout is processed in software.
- IP accounting for an ACL access violation on a given port is supported by forwarding all denied packets for that port to the MSFC for software processing without impacting other flows.
- Extended name-based MAC address ACLs are supported in hardware.
- The following ACL types are processed in software:
 - Standard XNS access list
 - Extended XNS access list
 - DECnet access list
 - Internetwork Packet Exchange (IPX) access lists
 - Extended MAC address access list
 - Protocol type-code access list



Note

IP packets with a header length of less than five will not be access controlled.

- Flows that require logging are processed in software without impacting nonlogged flow processing in hardware.
- The forwarding rate for software-processed flows is substantially less than for hardware-processed flows.
- When you enter the **show ip access-list** command, the match count displayed does not include packets processed in hardware.

Guidelines and Restrictions for Using Layer 4 Operators in ACLs

These sections describe guidelines and restrictions when configuring ACLs that include Layer 4 port operations:

- [Determining Layer 4 Operation Usage, page 23-3](#)
- [Determining Logical Operation Unit Usage, page 23-4](#)

Determining Layer 4 Operation Usage

You can specify these types of operations:

- gt (greater than)
- lt (less than)
- neq (not equal)
- eq (equal)
- range (inclusive range)

We recommend that you do not specify more than *nine different* operations on the same ACL. If you exceed this number, each new operation might cause the affected ACE to be translated into more than one ACE.

Use the following two guidelines to determine Layer 4 operation usage:

- Layer 4 operations are considered different if the operator or the operand differ. For example, in this ACL there are three different Layer 4 operations (“gt 10” and “gt 11” are considered two different Layer 4 operations):

```
... gt 10 permit
... lt 9 deny
... gt 11 deny
```

**Note**

There is no limit to the use of “eq” operators as the “eq” operator does not use a logical operator unit (LOU) or a Layer 4 operation bit. See the [“Determining Logical Operation Unit Usage” section on page 23-4](#) for a description of LOUs.

- Layer 4 operations are considered different if the same operator/operand couple applies once to a source port and once to a destination port. For example, in this ACL there are two different Layer 4 operations because one ACE applies to the source port and one applies to the destination port.

```
... Src gt 10 ...
... Dst gt 10
```

Determining Logical Operation Unit Usage

Logical operation units (LOUs) are registers that store operator-operand couples. All ACLs use LOUs. There can be up to 32 LOUs; each LOU can store two different operator-operand couples with the exception of the range operator. LOU usage per Layer 4 operation is as follows:

- gt uses 1/2 LOU
- lt uses 1/2 LOU
- neq uses 1/2 LOU
- range uses 1 LOU
- eq does not require a LOU

For example, this ACL would use a single LOU to store two different operator-operand couples:

```
... Src gt 10 ...
... Dst gt 10
```

A more detailed example follows:

```
ACL1
... (dst port) gt 10 permit
... (dst port) lt 9 deny
... (dst port) gt 11 deny
... (dst port) neq 6 permit
... (src port) neq 6 deny
... (dst port) gt 10 deny

ACL2
... (dst port) gt 20 deny
... (src port) lt 9 deny
... (src port) range 11 13 deny
... (dst port) neq 6 permit
```

The Layer 4 operations and LOU usage is as follows:

- ACL1 Layer 4 operations: 5
- ACL2 Layer 4 operations: 4
- LOUs: 4

An explanation of the LOU usage follows:

- LOU 1 stores “gt 10” and “lt 9”
- LOU 2 stores “gt 11” and “neq 6”
- LOU 3 stores “gt 20” (with space for one more)
- LOU 4 stores “range 11 13” (range needs the entire LOU)

Configuring the Cisco IOS Firewall Feature Set

**Note**

Release 12.1(11b)E and later releases include firewall feature set images.

These sections describe configuring the Cisco IOS firewall feature set on the Catalyst 6500 series switches:

- [Cisco IOS Firewall Feature Set Support Overview, page 23-5](#)
- [Firewall Configuration Guidelines and Restrictions, page 23-6](#)
- [Configuring CBAC on Catalyst 6500 Series Switches, page 23-7](#)

Cisco IOS Firewall Feature Set Support Overview

The firewall feature set images support these Cisco IOS firewall features:

- Context-based Access Control (CBAC)
- Port-to-Application Mapping (PAM)
- Authentication Proxy

These are the firewall feature set image names:

- c6sup22-jo3sv-mz
- c6sup22-po3sv-mz
- c6sup12-jo3sv-mz
- c6sup12-po3sv-mz

For more information about Cisco IOS firewall features, refer to the *Cisco IOS Security Configuration Guide, Release 12.1*, “Traffic Filtering and Firewalls” online publications:

- The “Cisco IOS Firewall Overview” chapter at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdfirwl.htm
- The “Configuring Context-Based Access Control” chapter at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdebac.htm
- The “Configuring Authentication Proxy” chapter at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdauthp.htm
- Cisco IOS Security Command Reference publication at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

The following features are supported with and without the use of a Cisco IOS firewall image:

- Standard access lists and static extended access lists
- Lock-and-key (dynamic access lists)
- IP session filtering (reflexive access lists)
- TCP intercept

- Security server support
- Network address translation
- Neighbor router authentication
- Event logging
- User authentication and authorization

**Note**

Catalyst 6500 series switches support the Intrusion Detection System Module (IDS) (WS-X6381-IDS). Catalyst 6500 series switches do not support the Cisco IOS firewall IDS feature, which is configured with the **ip audit** command.

Firewall Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring the Cisco IOS firewall features:

Restrictions

- On other platforms, if you enter the **ip inspect** command on a port, CBAC modifies ACLs on other ports to permit the inspected traffic to flow through the network device. On Catalyst 6500 series switches, you must enter the **mls ip inspect** commands to permit traffic through any ACLs that would deny the traffic through other ports. See the “[Configuring CBAC on Catalyst 6500 Series Switches](#)” section on page 23-7.
- With Supervisor Engine 2 and PFC2, reflexive ACLs and CBAC have conflicting flow mask requirements. When you configure CBAC on a switch with Supervisor Engine 2 and PFC2, reflexive ACLs are processed in software on the MSFC2.
- CBAC is incompatible with VACLs. You can configure both CBAC and VACLs on the switch but not in the same subnet (VLAN) or on the same interface.

**Note**

The Intrusion Detection System Module (IDS) uses VACLs to select traffic. To use the IDS in a subnet where CBAC is configured, enter the **mls ip ids acl_name** interface command, where *acl_name* is configured to select traffic for the IDS.

Guidelines

- To inspect Microsoft NetMeeting (2.0 or greater) traffic, turn on both **h323** and **tcp** inspection.
- To inspect web traffic, turn on **tcp** inspection. To avoid reduced performance, do not turn on **http** inspection to block Java.
- You can configure CBAC on physical ports configured as Layer 3 interfaces and on VLAN interfaces.
- QoS and CBAC do not interact or interfere with each other.

Configuring CBAC on Catalyst 6500 Series Switches

You need to do additional CBAC configuration on the Catalyst 6500 series switches. On a network device other than a Catalyst 6500 series switch, when ports are configured to deny traffic, CBAC permits traffic to flow bidirectionally through the port if it is configured with the **ip inspect** command. The same behavior applies to any other port that the traffic needs to go through, as shown in this example:

```
Router(config)# ip inspect name permit_ftp ftp
Router(config)# interface vlan 100
Router(config-if)# ip inspect permit_ftp in
Router(config-if)# ip access-group deny_ftp_a in
Router(config-if)# ip access-group deny_ftp_b out
Router(config-if)# exit
Router(config)# interface vlan 200
Router(config-if)# ip access-group deny_ftp_c in
Router(config-if)# ip access-group deny_ftp_d out
Router(config-if)# exit
Router(config)# interface vlan 300
Router(config-if)# ip access-group deny_ftp_e in
Router(config-if)# ip access-group deny_ftp_f out
Router(config-if)# end
```

If the FTP session enters on VLAN 100 and needs to leave on VLAN 200, CBAC permits the FTP traffic through ACLs deny_ftp_a, deny_ftp_b, deny_ftp_c, and deny_ftp_d. If another FTP session enters on VLAN 100 and needs to leave on VLAN 300, CBAC permits the FTP traffic through ACLs deny_ftp_a, deny_ftp_b, deny_ftp_e, and deny_ftp_f.

On a Catalyst 6500 series switch, when ports are configured to deny traffic, CBAC permits traffic to flow bidirectionally only through the port configured with the **ip inspect** command. You must configure other ports with the **mls ip inspect** command.

If the FTP session enters on VLAN 100 and needs to leave on VLAN 200, CBAC on a Catalyst 6500 series switch permits the FTP traffic only through ACLs deny_ftp_a and deny_ftp_b. To permit the traffic through ACLs deny_ftp_c and deny_ftp_d, you must enter the **mls ip inspect deny_ftp_c** and **mls ip inspect deny_ftp_d** commands, as shown in this example:

```
Router(config)# mls ip inspect deny_ftp_c
Router(config)# mls ip inspect deny_ftp_d
```

With the example configuration, FTP traffic cannot leave on VLAN 300 unless you enter the **mls ip inspect deny_ftp_e** and **mls ip inspect deny_ftp_f** commands. Enter the **show fm insp [detail]** command to verify the configuration.

The **show fm insp [detail]** command displays the list of ACLs and ports on which CBAC is configured and the status (**ACTIVE** or **INACTIVE**), as shown in this example:

```
Router# show fm insp
      interface:Vlan305(in) status :ACTIVE
      acl name:deny
      interfaces:
          Vlan305(out):status ACTIVE
```

On VLAN 305, inspection is active in the inbound direction and no ACL exists. ACL **deny** is applied on VLAN 305 in the outbound direction and inspection is active.

To display all of the flow information, use the **detail** keyword.

If a VACL is configured on the port before configuring CBAC, the status displayed is **INACTIVE**; otherwise, it is **ACTIVE**. If PFC resources are exhausted, the command displays the word “BRIDGE” followed by the number of currently active NetFlow requests that failed, which have been sent to the MSFC2 for processing.

Configuring MAC Address-Based Traffic Blocking

With 12.1(13)E and later releases, to block all traffic to or from a MAC address in a specified VLAN, perform this task:

Command	Purpose
Router(config)# mac-address-table static <i>mac_address</i> vlan <i>vlan_ID</i> drop	Blocks all traffic to or from the configured MAC address in the specified VLAN.
Router(config)# no mac-address-table static <i>mac_address</i> vlan <i>vlan_ID</i>	Clears MAC address-based blocking.

This example shows how to block all traffic to or from MAC address 0050.3e8d.6400 in VLAN 12:

```
Router# configure terminal
Router(config)# mac-address-table static 0050.3e8d.6400 vlan 12 drop
```

Configuring VLAN ACLs



Note

Releases 12.1(11b)E or later supports VLAN ACLs (VACLs).

The following sections describe VACLs:

- [Understanding VACLs, page 23-8](#)
- [Configuring VACLs, page 23-11](#)
- [Configuring VACL Logging, page 23-17](#)

Understanding VACLs

These sections describe VACLs:

- [VACL Overview, page 23-8](#)
- [Bridged Packets, page 23-9](#)
- [Routed Packets, page 23-10](#)
- [Multicast Packets, page 23-11](#)

VACL Overview

VACLs can provide access control for all packets that are bridged within a VLAN or that are routed into or out of a VLAN or, with releases 12.1(13)E or later, a WAN interface for VACL capture. Unlike regular Cisco IOS standard or extended ACLs that are configured on router interfaces only and are applied on routed packets only, VACLs apply to all packets and can be applied to any VLAN or WAN interface. VACLs are processed in hardware. VACLs use Cisco IOS ACLs. VACLs ignore any Cisco IOS ACL fields that are not supported in hardware.

You can configure VACLs for IP, IPX, and MAC-Layer traffic. VACLs applied to WAN interfaces support only IP traffic for VACL capture.

When you configure a VACL and apply it to a VLAN, all packets entering the VLAN are checked against this VACL. If you apply a VACL to the VLAN and an ACL to a routed interface in the VLAN, a packet coming in to the VLAN is first checked against the VACL and, if permitted, is then checked against the input ACL before it is handled by the routed interface. When the packet is routed to another VLAN, it is first checked against the output ACL applied to the routed interface and, if permitted, the VACL configured for the destination VLAN is applied. If a VACL is configured for a packet type and a packet of that type does not match the VACL, the default action is deny.

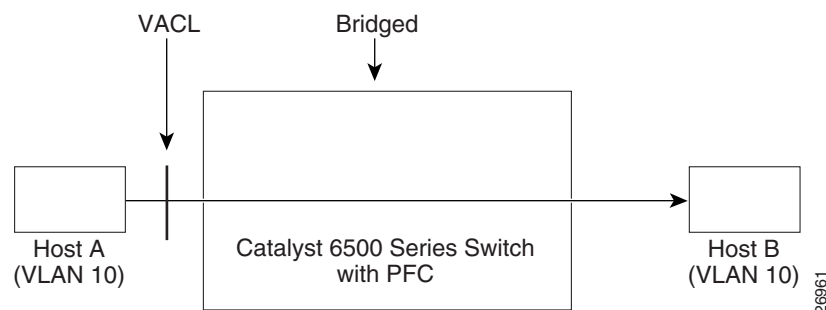
**Note**

- VACLs and CBAC cannot be configured on the same interface.
- TCP Intercepts and Reflexive ACLs take precedence over a VACL action if these are configured on the same interface.
- IGMP packets are not checked against VACLs.

Bridged Packets

Figure 23-1 shows a VACL applied on bridged packets.

Figure 23-1 Applying VACLs on Bridged Packets

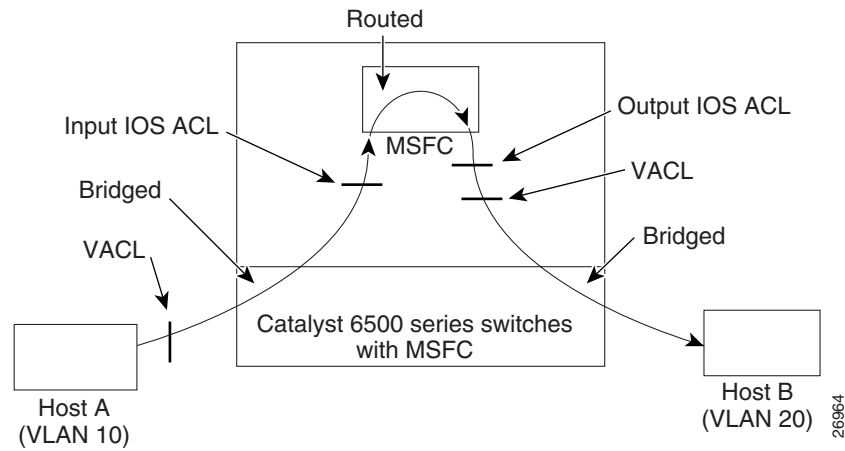


Routed Packets

Figure 23-2 shows how ACLs are applied on routed and Layer 3-switched packets. For routed or Layer 3-switched packets, the ACLs are applied in the following order:

1. VACL for input VLAN
2. Input Cisco IOS ACL
3. Output Cisco IOS ACL
4. VACL for output VLAN

Figure 23-2 Applying VACLs on Routed Packets

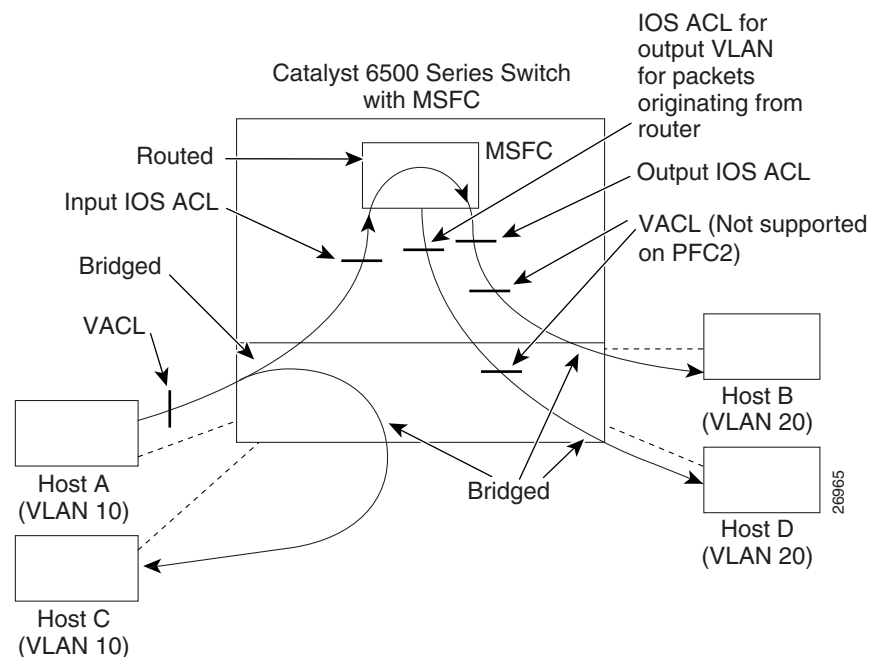


Multicast Packets

Figure 23-3 shows how ACLs are applied on packets that need multicast expansion. For packets that need multicast expansion, the ACLs are applied in the following order:

1. Packets that need multicast expansion:
 - a. VACL for input VLAN
 - b. Input Cisco IOS ACL
2. Packets after multicast expansion:
 - a. Output Cisco IOS ACL
 - b. VACL for output VLAN (not supported with PFC2)
3. Packets originating from router—VACL for output VLAN

Figure 23-3 Applying VACLs on Multicast Packets



Configuring VACLs

These sections describe configuring VACLs:

- [VACL Configuration Overview](#), page 23-12
- [Defining a VLAN Access Map](#), page 23-12
- [Configuring a Match Clause in a VLAN Access Map Sequence](#), page 23-13
- [Configuring an Action Clause in a VLAN Access Map Sequence](#), page 23-14
- [Applying a VLAN Access Map](#), page 23-14
- [Verifying VLAN Access Map Configuration](#), page 23-15

- [VLAN Access Map Configuration and Verification Examples, page 23-15](#)
- [Configuring a Capture Port, page 23-16](#)

VACL Configuration Overview

VACLs use standard and extended Cisco IOS IP and IPX ACLs, and MAC-Layer named ACLs (see the “[Configuring MAC-Layer Named Access Lists \(Optional\)](#)” section on page 31-39) and VLAN access maps.

VLAN access maps can be applied to VLANs or, with releases 12.1(13)E or later, to WAN interfaces for VACL capture. VACLs attached to WAN interfaces support only standard and extended Cisco IOS IP ACLs for VACL capture.

Each VLAN access map can consist of one or more map sequences, each sequence with a match clause and an action clause. The match clause specifies IP, IPX, or MAC ACLs for traffic filtering and the action clause specifies the action to be taken when a match occurs. When a flow matches a permit ACL entry, the associated action is taken and the flow is not checked against the remaining sequences. When a flow matches a deny ACL entry, it will be checked against the next ACL in the same sequence or the next sequence. If a flow does not match any ACL entry and at least one ACL is configured for that packet type, the packet is denied.

To use access-control for both bridged and routed traffic, you can use VACLs alone or a combination of VACLs and ACLs. You can define ACLs on the VLAN interfaces to use access-control for both the input and output routed traffic. You can define a VACL to use access-control for the bridged traffic.

The following caveats apply to ACLs when used with VACLs:

- Packets that require logging on the outbound ACLs are not logged if they are denied by a VACL.
- VACLs are applied on packets before NAT translation. If the translated flow is not subject to access control, the flow might be subject to access control after the translation because of the VACL configuration.

The action clause in a VACL can be forward, drop, capture, or redirect. Traffic can also be logged. VACLs applied to WAN interfaces do not support the redirect or log actions.



Note

VACLs have an implicit deny at the end of the map; a packet is denied if it does not match any ACL entry, and at least one ACL is configured for the packet type.



Note

If an empty or undefined ACL is specified in a VACL, any packets will match the ACL and the associated action is taken.

Defining a VLAN Access Map

To define a VLAN access map, perform this task:

Command	Purpose
Router(config)# vlan access-map <i>map_name</i> [0-65535]	Defines the VLAN access map. Optionally, you can specify the VLAN access map sequence number.
Router(config)# no vlan access-map <i>map_name</i> 0-65535	Deletes a map sequence from the VLAN access map.
Router(config)# no vlan access-map <i>map_name</i>	Deletes the VLAN access map.

When defining a VLAN access map, note the following syntax information:

- To insert or modify an entry, specify the map sequence number.
- If you do not specify the map sequence number, a number is automatically assigned.
- You can specify only one match clause and one action clause per map sequence.
- Use the **no** keyword with a sequence number to remove a map sequence.
- Use the **no** keyword without a sequence number to remove the map.

See the “VLAN Access Map Configuration and Verification Examples” section on page 23-15.

Configuring a Match Clause in a VLAN Access Map Sequence

To configure a match clause in a VLAN access map sequence, perform this task:

Command	Purpose
Router(config-access-map)# match {ip address {1-199 1300-2699 acl_name} ipx address {800-999 acl_name} mac address acl_name}	Configures the match clause in a VLAN access map sequence.
Router(config-access-map)# no match {ip address {1-199 1300-2699 acl_name} ipx address {800-999 acl_name} mac address acl_name}	Deletes the match clause in a VLAN access map sequence.

When configuring a match clause in a VLAN access map sequence, note the following syntax information:

- You can select one or more ACLs.
- VACLs attached to WAN interfaces support only standard and extended Cisco IOS IP ACLs.
- Use the **no** keyword to remove a match clause or specified ACLs in the clause.
- For information about named MAC-Layer ACLs, refer to the “Configuring MAC-Layer Named Access Lists (Optional)” section on page 31-39.
- For information about Cisco IOS ACLs, refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, “Traffic Filtering and Firewalls,” “Access Control Lists: Overview and Guidelines,” at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/index.htm

See the “VLAN Access Map Configuration and Verification Examples” section on page 23-15.

Configuring an Action Clause in a VLAN Access Map Sequence

To configure an action clause in a VLAN access map sequence, perform this task:

Command	Purpose
<pre>Router(config-access-map)# action {drop [log]} {forward [capture]} {redirect {{ethernet fastethernet gigabitethernet tengigabitethernet} slot/port} {port-channel channel_id}}</pre>	Configures the action clause in a VLAN access map sequence.
<pre>Router(config-access-map)# no action {drop [log]} {forward [capture]} {redirect {{ethernet fastethernet gigabitethernet tengigabitethernet} slot/port} {port-channel channel_id}}</pre>	Deletes the action clause in from the VLAN access map sequence.

When configuring an action clause in a VLAN access map sequence, note the following syntax information:

- You can set the action to drop, forward, forward capture, or redirect packets.
- VACLs applied to WAN interfaces support only the forward capture action. VACLs applied to WAN interfaces do not support the drop, forward, or redirect actions.
- Forwarded packets are still subject to any configured Cisco IOS security ACLs.
- The **capture** action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured. For more information about the **capture** action, see the “[Configuring a Capture Port](#)” section on page 23-16.
- The **log** action is supported only on Supervisor Engine 2.
- VACLs applied to WAN interfaces do not support the **log** action.
- When the **log** action is specified, dropped packets are logged in software. Only dropped IP packets can be logged.
- The **redirect** action allows you to specify up to five interfaces, which can be physical interfaces or EtherChannels. You cannot specify packets to be redirected to an EtherChannel member or a VLAN interface.
- For systems with a Supervisor Engine 2, the redirect interface must be in the VLAN for which the VACL access map is configured. For systems with Supervisor Engine 1, the redirect interface must be in the redirected packet’s source VLAN.
- Use the **no** keyword to remove an action clause or specified redirect interfaces.

See the “[VLAN Access Map Configuration and Verification Examples](#)” section on page 23-15.

Applying a VLAN Access Map

To apply a VLAN access map, perform this task:

Command	Purpose
<pre>Router(config)# vlan filter map_name {vlan-list vlan_list interface type¹ number²} CP_CmdPlain</pre>	Applies the VLAN access map to the specified VLANs or WAN interfaces.

Command	Purpose
Router(config)# no vlan filter <i>map_name</i> [vlan-list <i>vlan_list</i> interface <i>type</i> ¹ <i>number</i> ²]	Removes the VLAN access map from the specified VLANs or WAN interfaces.

1. *type* = **pos**, **atm**, or **serial**
2. *number* = *slot/port* or *slot/port_adapter/port*; can include a subinterface or channel group descriptor

When applying a VLAN access map, note the following syntax information:

- You can apply the VLAN access map to one or more VLANs or WAN interfaces.
- The *vlan_list* parameter can be a single VLAN ID or a comma-separated list of VLAN IDs or VLAN ID ranges (*vlan_ID-vlan_ID*).
- If you delete a WAN interface that has a VACL applied, the VACL configuration on the interface is also removed.
- You can apply only one VLAN access map to each VLAN or WAN interface.
- VACLs applied to VLANs are active only for VLANs with a Layer 3 VLAN interface configured. VACLs applied to VLANs without a Layer 3 VLAN interface are inactive. With releases 12.1(13)E and later, applying a VLAN access map to a VLAN without a Layer 3 VLAN interface creates an administratively down Layer 3 VLAN interface to support the VLAN access map. If creation of the Layer 3 VLAN interface fails, the VACL is inactive.
- You cannot apply a VACL to a secondary private VLAN. VACLs applied to primary private VLANs also apply to secondary private VLANs.
- Use the **no** keyword to clear VLAN access maps from VLANs or WAN interfaces.

See the “[VLAN Access Map Configuration and Verification Examples](#)” section on page 23-15.

Verifying VLAN Access Map Configuration

To verify VLAN access map configuration, perform this task:

Command	Purpose
Router# show vlan access-map [<i>map_name</i>]	Verifies VLAN access map configuration by displaying the content of a VLAN access map.
Router# show vlan filter [access-map <i>map_name</i> vlan <i>vlan_id</i> interface <i>type</i> ¹ <i>number</i> ²]	Verifies VLAN access map configuration by displaying the mappings between VACLs and VLANs.

1. *type* = **pos**, **atm**, or **serial**
2. *number* = *slot/port* or *slot/port_adapter/port*; can include a subinterface or channel group descriptor

VLAN Access Map Configuration and Verification Examples

Assume IP-named ACL **net_10** and **any_host** are defined as follows:

```
Router# show ip access-lists net_10
Extended IP access list net_10
    permit ip 10.0.0.0 0.255.255.255 any

Router# show ip access-lists any_host
Standard IP access list any_host
    permit any
```

This example shows how to define and apply a VLAN access map to forward IP packets. In this example, IP traffic matching net_10 is forwarded and all other IP packets are dropped due to the default drop action. The map is applied to VLAN 12 to 16.

```
Router(config)# vlan access-map thor 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action forward
Router(config-access-map)# exit
Router(config)# vlan filter thor vlan-list 12-16
```

This example shows how to define and apply a VLAN access map to drop and log IP packets. In this example, IP traffic matching net_10 is dropped and logged and all other IP packets are forwarded:

```
Router(config)# vlan access-map ganymede 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action drop log
Router(config-access-map)# exit
Router(config)# vlan access-map ganymede 20
Router(config-access-map)# match ip address any_host
Router(config-access-map)# action forward
Router(config-access-map)# exit
Router(config)# vlan filter ganymede vlan-list 7-9
```

This example shows how to define and apply a VLAN access map to forward and capture IP packets. In this example, IP traffic matching net_10 is forwarded and captured and all other IP packets are dropped:

```
Router(config)# vlan access-map mordred 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action forward capture
Router(config-access-map)# exit
Router(config)# vlan filter mordred vlan-list 2, 4-6
```

Configuring a Capture Port

A port configured to capture VACL-filtered traffic is called a capture port.



Note

To apply IEEE 802.1Q or ISL tags to the captured traffic, configure the capture port to trunk unconditionally (see the “[Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk](#)” section on page 7-8 and the “[Configuring the Layer 2 Trunk Not to Use DTP](#)” section on page 7-9).

To configure a capture port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port}}	Specifies the interface to configure.
Step 2	Router(config-if)# switchport capture allowed vlan {add all except remove} <i>vlan_list</i>	(Optional) With Release 12.1(13)E and later releases, filters the captured traffic on a per-destination-VLAN basis. The default is all .
	Router(config-if)# no switchport capture allowed vlan	Clears the configured destination VLAN list and returns to the default value (all).
Step 3	Router(config-if)# switchport capture	Configures the port to capture VACL-filtered traffic.
	Router(config-if)# no switchport capture	Disables the capture function on the interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When configuring a capture port, note the following syntax information:

- With Release 12.1(13)E and later releases, you can configure any port as a capture port. With earlier releases, only the Gigabit Ethernet monitor port on the IDS module can be configured as a capture port.
- When configuring a capture port with Release 12.1(13)E and later releases, note the following syntax information:
 - The *vlan_list* parameter can be a single VLAN ID or a comma-separated list of VLAN IDs or VLAN ID ranges (*vlan_ID–vlan_ID*).
 - To encapsulate captured traffic, configure the capture port with the **switchport trunk encapsulation** command (see the “[Configuring a Layer 2 Switching Port as a Trunk](#)” section on page 7-8) before you enter the **switchport capture** command.
 - To not encapsulate captured traffic, configure the capture port with the **switchport mode access** command (see the “[Configuring a LAN Interface as a Layer 2 Access Port](#)” section on page 7-14) before you enter the **switchport capture** command.
 - The capture port supports only egress traffic. No traffic can enter the switch through a capture port.

This example shows how to configure a Fast Ethernet interface 5/1 as a capture port:

```
Router(config)# interface gigabitEthernet 5/1
Router(config-if)# switchport capture
Router(config-if)# end
```

This example shows how to display VLAN access map information:

```
Router# show vlan access-map mordred
Vlan access-map "mordred" 10
    match: ip address net_10
    action: forward capture
Router#
```

This example shows how to display mappings between VACLs and VLANs. For each VACL map, there is information about the VLANs that the map is configured on and the VLANs that the map is active on. A VACL is not active if the VLAN does not have an interface.

```
Router# show vlan filter
VLAN Map mordred:
    Configured on VLANs: 2,4-6
    Active on VLANs: 2,4-6
Router#
```

Configuring VACL Logging

When you configure VACL logging, IP packets that are denied generate log messages in these situations:

- When the first matching packet is received
- For any matching packets received during the last 5-minute interval
- If the threshold is reached before the 5-minute interval

Log messages are generated on a per-flow basis. A flow is defined as packets with the same IP addresses and Layer 4 (UDP or TCP) port numbers. When a log message is generated, the timer and packet count is reset.

These restrictions apply to VACL logging:

- Supported only with Supervisor Engine 2.
- Because of the rate-limiting function for redirected packets, VACL logging counters may not be accurate.
- Only denied IP packets are logged.

To configure VACL logging, use the **action drop log** command action in VLAN access map submode (see the “Configuring VACLs” section on page 23-11 for configuration information) and perform this task in global configuration mode to specify the global VACL logging parameters:

	Command	Purpose
Step 1	Router(config)# vlan access-log maxflow <i>max_number</i>	Sets the log table size. The content of the log table can be deleted by setting the maxflow number to 0. The default is 500 with a valid range of 0 to 2048. When the log table is full, logged packets from new flows are dropped by the software.
Step 2	Router(config)# vlan access-log ratelimit <i>pps</i>	Sets the maximum redirect VACL logging packet rate. The default packet rate is 2000 packets per second with a valid range of 0 to 5000. Packets exceeding the limit are dropped by the hardware.
Step 3	Router(config)# vlan access-log threshold <i>pkt_count</i>	Sets the logging threshold. A logging message is generated if the threshold for a flow is reached before the 5-minute interval. By default, no threshold is set.
Step 4	Router(config)# exit	Exits VLAN access map configuration mode.
Step 5	Router# show vlan access-log config	(Optional) Displays the configured VACL logging properties.
Step 6	Router# show vlan access-log flow protocol { <i>src_addr src_mask</i> } any { host <i>hostname host_ip</i> }} { <i>dst_addr dst_mask</i> } any { host <i>hostname host_ip</i> }} [vlan <i>vlan_id</i>]	(Optional) Displays the content of the VACL log table.
Step 7	Router# show vlan access-log statistics	(Optional) Displays packet and message counts and other statistics.

This example shows how to configure global VACL logging in hardware:

```
Router(config)# vlan access-log maxflow 800
Router(config)# vlan access-log ratelimit 2200
Router(config)# vlan access-log threshold 4000
```

Configuring TCP Intercept

With Supervisor Engine 2 and PFC2, TCP intercept flows are processed in hardware.

With Supervisor Engine 1 and PFC, TCP intercept flows are processed in software.

For configuration procedures, refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, “Traffic Filtering and Firewalls,” “Configuring TCP Intercept,” at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/secur_c/scprt3/scdden1.htm

Configuring Unicast Reverse Path Forwarding

These sections describe configuring Cisco IOS Unicast Reverse Path Forwarding (Unicast RPF):

- [Understanding Unicast RPF Support, page 23-19](#)
- [Configuring Unicast RPF, page 23-19](#)
- [Enabling Self-Pinging, page 23-19](#)
- [Configuring the Unicast RPF Checking Mode, page 23-20](#)

Understanding Unicast RPF Support

The PFC2 supports Unicast RPF with hardware processing for packets that have a single return path. The MSFC2 processes traffic in software that has multiple return paths (for example, load sharing).

With a PFC2, if you configure Unicast RPF to filter with an ACL, the PFC2 determines whether or not traffic matches the ACL. The PFC2 sends the traffic denied by the RPF ACL to the MSFC2 for the Unicast RPF check.



Note

- Because the packets in a denial-of-service attack typically match the deny ACE and are sent to the MSFC2 for the unicast RPF check, they can overload the MSFC2.
- The PFC2 provides hardware support for traffic that does not match the Unicast RPF ACL, but that does match an input security ACL.

With Supervisor Engine 1 and PFC, the MSFC or MSFC 2 supports Unicast RPF in software.

Configuring Unicast RPF

For configuration procedures, refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, “Other Security Features,” “Configuring Unicast Reverse Path Forwarding” at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/secr_c/scprt5/scdrpf.htm

Enabling Self-Pinging

With Unicast RPF enabled, the switch cannot ping itself. To enable self-pinging, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# ip verify unicast source reachable-via any allow-self-ping Router(config-if)# no ip verify unicast source reachable-via any allow-self-ping	Enables the switch to ping itself or a secondary address. Disables self-pinging.
Step 3	Router(config-if)# exit	Exits interface configuration mode.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable self-pinging:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip verify unicast source reachable-via any allow-self-ping
Router(config-if)# end
```

Configuring the Unicast RPF Checking Mode

There are two Unicast RPF checking modes:

- Strict checking mode, which verifies that the source IP address exists in the FIB table and verifies that the source IP address is reachable through the input port.
- Exist-only checking mode, which only verifies that the source IP address exists in the FIB table.



Note

The most recently configured mode is automatically applied to all ports configured for Unicast RPF checking.

To configure Unicast RPF checking mode, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects an interface to configure. Note Based on the input port, Unicast RPF verifies the best return path before forwarding the packet on to the next destination.
Step 2	Router(config-if)# ip verify unicast source reachable-via {rx any} [allow-default] [list] Router(config-if)# no ip verify unicast	Configures the Unicast RPF checking mode. Reverts to the default Unicast RPF checking mode.
Step 3	Router(config-if)# exit	Exits interface configuration mode.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When configuring the Unicast RPF checking mode, note the following syntax information:

- Use the **rx** keyword to enable strict checking mode.
- Use the **any** keyword to enable exist-only checking mode.
- Use the **allow-default** keyword to allow use of the default route for RPF verification.
- Use the *list* option to identify an access list.
 - If the access list denies network access, spoofed packets are dropped at the port.
 - If the access list permits network access, spoofed packets are forwarded to the destination address. Forwarded packets are counted in the interface statistics.
 - If the access list includes the logging action, information about the spoofed packets is sent to the log server.



Note

When you enter the **ip verify unicast source reachable-via** command, the Unicast RPF checking mode changes on all ports in the switch.

This example shows how to enable Unicast RPF exist-only checking mode on Gigabit Ethernet port 4/1:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip verify unicast source reachable-via any
Router(config-if)# end
Router#
```

This example shows how to enable Unicast RPF strict checking mode on Gigabit Ethernet port 4/2:

```
Router(config)# interface gigabitethernet 4/2
Router(config-if)# ip verify unicast source reachable-via rx
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show running-config interface gigabitethernet 4/2
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/2
ip address 42.0.0.1 255.0.0.0
ip verify unicast reverse-path
no cdp enable
end
Router# show running-config interface gigabitethernet 4/1
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/1
ip address 41.0.0.1 255.0.0.0
→ ip verify unicast reverse-path (RPF mode on g4/1 also changed to strict-check RPF mode)
no cdp enable
end
Router#
```

Configuring Unicast Flood Protection

The unicast flood protection feature protects the system from disruptions caused by unicast flooding. The Catalyst 6500 series switches use forwarding tables to direct traffic to specific ports based on the VLAN number and the destination MAC address of the frame. When there is no entry corresponding to the frame's destination MAC address in the incoming VLAN, the frame is sent to all forwarding ports within the respective VLAN, which causes flooding. Limited flooding is part of the normal switching process, but continuous flooding can cause adverse performance effects on the network.

When you enable the unicast flood protection feature, the system sends an alert when the rate limit has been exceeded, filters the traffic, or shuts down the port generating the floods when it detects unknown unicast floods exceeding a threshold.

To configure unicast flood protection, perform this task:

	Command	Purpose
Step 1	Router(config)# [no] mac-address-table unicast-flood {limit <i>kfps</i> } {vlan <i>vlan</i> } {filter <i>timeout</i> alert shutdown}	Enables unicast flood protection globally.
Step 2	Router# show mac-address-table unicast-flood	Displays unicast flood protection information.

When configuring unicast flood protection, note the following syntax information:

- Use the **limit** keyword to specify the unicast floods on a per source MAC address and per VLAN basis; valid values are from 1 to 4000 floods per second (fps).
- Use the **filter** keyword to specify how long to filter unicast flood traffic; valid values are from 1 to 34560 minutes.
- Use the **alert** keyword to configure the system to send an alert message when frames of unicast floods exceed the flood rate limit.
- Use the **shutdown** keyword to configure the system to shut down the ingress port generating the floods when frames of unicast floods exceed the flood rate limit.

This example shows how to configure the system to filter unicast flood traffic for 5 minutes and set the flood rate limit to 3000 fps:

```
Router(config)# mac-address-table unicast-flood limit 3 vlan 100 filter 5
Router # show mac-address-table unicast-flood
Unicast Flood Protection status: enabled

Configuration:
vlan      Kfps      action      timeout
-----+-----+-----+-----
   100         3          filter        5

Mac filters:
No.  vlan  source mac addr.      installed on      time left (mm:ss)
-----+-----+-----+-----+-----+-----
Router(config)#
```

Configuring MAC Move Notification

When you configure MAC move notification, a message is generated when a MAC address moves from one port to another.



Note

The MAC address move notification feature does not generate a notification when a new MAC address is added to the CAM or when a MAC address is removed from the CAM.

To configure MAC move notification, perform this task:

	Command	Purpose
Step 1	Router(config)# [no] mac-address-table notification mac-move	Enables MAC move notification globally.
Step 2	Router# show mac-address-table notification mac-move	Displays MAC move notification information.

This example shows how to enable the MAC move notification feature:

```
Router(config)# mac-address-table notification mac-move
Router# show mac-address-table notification mac-move
MAC Move Notification: enabled
Router#
```




Configuring Denial of Service Protection

This chapter contains information on how to protect your system against Denial of Service (DoS) attacks. The information covered in this chapter is unique to the Catalyst 6500 series switches, and it supplements the network security information and procedures in the “[Configuring Network Security](#)” in this publication as well as the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/index.htm
- *Cisco IOS Security Command Reference*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_r/index.htm

This chapter consists of these sections:

- [DoS Protection Overview](#), page 24-1
- [Configuring DoS Protection](#), page 24-2

DoS Protection Overview

The DoS protection available on the Catalyst 6500 series switch provides support against two types of DoS attack scenarios:

- Data-packet processing that starves routing-protocol processing may result in DoS attacks such as the following:
 - Routing peer loss due to hello timeouts
 - HSRP peer loss due to hello timeouts
 - Routing protocol slow convergence
- Data packets congesting a CPU inband datapath may result in DoS attacks such as the following:
 - Routing peer loss due to hello packet drops
 - HSRP peer loss due to hello packet drops



Note

DoS protection used at the local router may not prevent peer loss caused by data-packet congestion on the external link.

Configuring DoS Protection

The following sections describe the different DoS protection implementations and give configuration examples:

- [Supervisor Engine DoS Protection, page 24-2](#)
- [Security ACLs, page 24-2](#)
- [QoS ACLs, page 24-4](#)
- [Forwarding Information Base Rate-Limiting, page 24-5](#)
- [ARP Throttling, page 24-5](#)
- [Monitoring Packet Drop Statistics, page 24-6](#)

Supervisor Engine DoS Protection

The supervisor engine has built-in mechanisms that limit the rate of traffic in hardware and prevent flooding of the route processor and denial of service. Rate-limiting allows most of the traffic to be dropped in hardware and only a small percentage of the traffic to be forwarded to the route processor at a nonconfigurable rate of 0.5 packets per second. Rate-limiting of packets in hardware exists for the following traffic conditions:

- ICMP unreachable messages for ACL deny

This condition allows most ACL-denied packets to be dropped in hardware, and some packets to be forwarded to the route processor for monitoring purposes.



Note Because the system is programmed to bridge all ACL-deny log packets to the route processor, we do not recommend that you configure deny log ACEs in a security ACL.

- ICMP redirect messages

ICMP redirect messages are used by routers to notify the hosts on the data link that a better route is available for a particular destination. Most of these messages are dropped in hardware and only a few messages need to reach the route processor.

- Forwarding Information Base (FIB) Failures

If the FIB does not know how to route traffic for a specific IP destination address, some packets will be forwarded to the route processor to generate ICMP redirect messages.

- Reverse Path Forwarding (RPF) Failures

If the FIB IP source address lookup results in an RPF failure, some packets will be forwarded to the route processor to generate ICMP unreachable messages.

Security ACLs

The Catalyst 6500 series switch can deny packets in hardware using security ACLs and can drop DoS packets before they reach the CPU inband datapath. Because security ACLs are applied in hardware using the TCAM, long security ACLs can be used without impacting the throughput of other traffic. Security ACLs can also be applied after a DoS attack has been identified.

When using security ACLs to drop DoS packets, note the following information:

- The security ACL must specify the traffic flow to be dropped.
- When adding a security ACL to block DoS packets to an interface that already has a security ACL configured, you must merge the DoS security ACL with the existing security ACL.
- Security ACLs need to be configured on all external interfaces that require protection. Use the interface range command to configure a security ACL on multiple interfaces.

The following example shows how a security ACL is used to drop DoS packets:

```
Router# clear mls ip mod 9
Router# show mls ip mod 9
Displaying Netflow entries in module 9
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
Pkts          Bytes          Age  LastSeen  Attributes
-----
199.1.1.1     199.2.1.1     0   :0       :0       0   : 0
1843          84778         2   02:30:17  L3 - Dynamic
199.2.1.1     199.1.1.1     0   :0       :0       0   : 0
2742416      126151136    2   02:30:17  L3 - Dynamic          traffic flow identified
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# no access-list 199
Router(config)# access-list 199 deny ip host 199.1.1.1 any
Router(config)# access-list 199 permit ip any any
Router(config)# interface g9/1
Router(config-if)# ip access 199 in          security ACL applied
Router(config-if)# end
Router#
1w6d: %SYS-5-CONFIG_I: Configured from console by console
Router# clear mls ip mod 9
Router# show mls ip mod 9
Displaying Netflow entries in module 9
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
Pkts          Bytes          Age  LastSeen  Attributes
-----
199.1.1.1     199.2.1.1     0   :0       :0       0   : 0
1542          70932         2   02:31:56  L3 - Dynamic
199.2.1.1     199.1.1.1     0   :0       :0       0   : 0
0             0             2   02:31:56  L3 - Dynamic          hardware-forwarded
                                                                traffic stopped

Extended IP access list 199
  deny ip host 199.1.1.1 any (100 matches)
  permit ip any any
Router# show access-list 199
Extended IP access list 199
  deny ip host 199.1.1.1 any (103 matches)          rate limiting at 0.5 pps
  permit ip any any
Router #
```

QoS ACLs

Unlike Security ACLs, QoS ACLs can be used to limit the rate of traffic without denying access to all the traffic in a flow.

When using QoS ACLs to limit the rate of packets, note the following information:

- The QoS ACL must specify the traffic flow to be rate-limited.
- When adding a QoS ACL to limit the rate of packets to an interface that already has a QoS ACL configured, you must merge the rate-limiting ACL with the existing QoS ACL.
- QoS ACLs need to be configured on all external interfaces that require protection. Use the interface range command to configure an ACL on multiple interfaces.

The following example shows how to use a QoS ACL to prevent a ping attack on a router. A QoS ACL is configured and applied on all interfaces to limit the rate of incoming ICMP echo packets.

```
Router# show ip ospf neighbors

Neighbor ID      Pri   State           Dead Time   Address        Interface
6.6.6.122        1    FULL/BDR        00:00:30   6.6.6.122     Vlan46
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                Interface    Hold Uptime   SRTT   RTO  Q  Seq Type
                               (sec)      (ms)          Cnt  Num
0   4.4.4.122                V144        11 00:06:07   4     200  0  6555
Router#


ping attack starts


Router# show proc cpu | include CPU utilization
CPU utilization for five seconds: 99%/90%; one minute: 48%; five minutes: 25%
Router#
2w0d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from FULL to DOWN, Neighbor
Down: Dead timer expired
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
Router#
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# access-list 199 permit icmp any any echo
Router(config)# class-map match-any icmp
Router(config-cmap)# match access-group 199
Router(config-cmap)# exit
Router(config)# policy-map icmp
Router(config-pmap)# class icmp
Router(config-pmap-c)# police 96000 16000 16000 conform-action transmit exceed-action drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface range g4/1 - 9
Router(config-if-range)# service-policy input icmp


policy applied


Router(config-if-range)# end
2w0d: %SYS-5-CONFIG_I: Configured from console by console
2w0d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from LOADING to FULL, Loading
Done
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                Interface    Hold Uptime   SRTT   RTO  Q  Seq Type
                               (sec)      (ms)          Cnt  Num
0   4.4.4.122                V144        13 00:00:48   8     200  0  6565
Router#
```

Forwarding Information Base Rate-Limiting

The forwarding information base (FIB) rate-limiting allows all packets that require software processing to be rate limited.

The following FIB rate-limiting usage guidelines apply:

- FIB rate-limiting does not limit the rate of multicast traffic.
- FIB rate-limiting does not differentiate between legitimate and illegitimate traffic (for example, tunnels, Telnet).
- FIB rate-limiting applies aggregate rate-limiting and not per flow rate-limiting.

The following example shows traffic destined for a nonexistent host address on a locally connected subnet. Normally, the ARP request would result in an ARP reply and the installation of a FIB adjacency for this traffic. However, the adjacency in the FIB for the destination subnet would continue to receive traffic that would, in turn, be forwarded for software processing. By applying rate-limiting to this traffic, the rate of traffic forwarded for software processing can be limited to a manageable amount.

```
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                Interface    Hold Uptime    SRTT   RTO   Q   Seq Type
                               (sec)          (ms)          Cnt Num
0   4.4.4.122                V144        11 00:00:26    8     200  0   6534
Router# show ip ospf neighbors

Neighbor ID      Pri   State           Dead Time   Address        Interface
6.6.6.122        1    FULL/BDR        00:00:36   6.6.6.122     Vlan46
→ Router#                               attack starts
Router# show arp | include 199.2.250.250
Internet 199.2.250.250      0    Incomplete     ARPA
Router#
1w6d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from FULL to DOWN, Neighbor
Down: Dead timer expired
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
→ Router(config)# mls ip cef rate-limit 1000          traffic rate limited to 1000 pps
Router(config)# end
Router#
1w6d: %SYS-5-CONFIG_I: Configured from console by console
Router#
1w6d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from LOADING to FULL, Loading
Done
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                Interface    Hold Uptime    SRTT   RTO   Q   Seq Type
                               (sec)          (ms)          Cnt Num
0   4.4.4.122                V144        12 00:00:07    12    200  0   6536
Router#
```

ARP Throttling

ARP throttling limits the rate at which packets destined to a connected network are forwarded to the route processor. Most of these packets are dropped, but a small number are sent to the router (rate limited).

Monitoring Packet Drop Statistics

Because the rate-limiting mechanism allows a certain number of packets to be forwarded for software processing, you can view the packet drop statistics by entering NetFlow **show** commands from the CLI. You can also capture the incoming or outgoing traffic on an interface and send a copy of this traffic to an external interface for monitoring by, for example, a traffic analyzer. To capture traffic and forward it to an external interface, use the **monitor session** commands.

Monitoring Dropped Packets Using NetFlow Commands

The following NetFlow commands display flows that are destined to the router MAC that are either hardware switched or forwarded to the route processor.

Displaying statistics based on source or flow only works if the MLS NetFlow flowmask is set to a value greater than destination-only.

```
Router# show mls ip
Displaying Netflow entries in Supervisor Earl
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
200.2.5.3      0.0.0.0        0 :0             :0             0 : 0

Pkts          Bytes          Age  LastSeen  Attributes
-----
0             0              1   01:52:25  L3 - Dynamic
```

```
Router# show mls netflow flowmask
current ip flowmask for unicast: destination only
current ipx flowmask for unicast: destination only
```

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls flow ip destination-source
Router(config)# exit
1w6d: %SYS-5-CONFIG_I: Configured from console by console
Router# show mls ip
Displaying Netflow entries in Supervisor Earl
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
200.2.5.3      223.255.254.226 0 :0             :0             0 : 0

Pkts          Bytes          Age  LastSeen  Attributes
-----
0             0              2   01:54:05  L3 - Dynamic
```

```
Router#
```

When you use the **show mls ip** command to display information about flows for a specific source or destination address, the command accepts 32 host prefixes only. When you use the output modifiers, you might see all flows from a specific subnet.

```
Router# show mls ip source 9.9.9.2 mod 4
Displaying Netflow entries in module 4
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
9.9.9.177      9.9.9.2        0 :0             :0             0 : 0

Pkts          Bytes          Age  LastSeen  Attributes
-----
0             0              28  01:56:59  L3 - Dynamic

Router# show mls ip mod 4 | include 9.9.9
9.9.9.177      9.9.9.2        0 :0             :0             0 : 0
```

```
9.9.9.177      9.9.9.1      0   :0      :0      0   : 0
```

Monitoring Dropped Packets Using Monitor Session Commands

This example shows how to use the **monitor session** command to capture and forward traffic to an external interface:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# monitor session 1 source vlan 44 both
Router(config)# monitor session 1 destination interface g9/1
Router(config)# end
Router#
2w0d: %SYS-5-CONFIG_I: Configured from console by console
Router# show monitor session 1
Session 1
-----
Source Ports:
  RX Only:      None
  TX Only:      None
  Both:         None
Source VLANs:
  RX Only:      None
  TX Only:      None
  Both:         44
Destination Ports: Gi9/1
Filter VLANs:   None
```




Configuring IEEE 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication to prevent unauthorized devices (clients) from gaining access to the network. Release 12.1(13)E and later releases support 802.1X port-based authentication.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding 802.1X Port-Based Authentication, page 25-1](#)
- [Default 802.1X Port-Based Authentication Configuration, page 25-5](#)
- [802.1X Port-Based Authentication Guidelines and Restrictions, page 25-6](#)
- [Configuring 802.1X Port-Based Authentication, page 25-7](#)
- [Displaying 802.1X Status, page 25-15](#)

Understanding 802.1X Port-Based Authentication

The IEEE 802.1X standard defines a client-server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a switch port and assigns the port to a VLAN before making available any services offered by the switch or the LAN.

Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

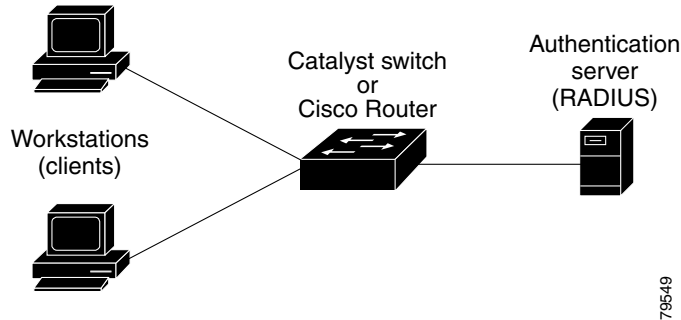
These sections describe IEEE 802.1X port-based authentication:

- [Device Roles, page 25-2](#)
- [Authentication Initiation and Message Exchange, page 25-3](#)
- [Ports in Authorized and Unauthorized States, page 25-4](#)
- [Supported Topologies, page 25-4](#)

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles as shown in Figure 25-1.

Figure 25-1 802.1X Device Roles



The specific roles shown in Figure 25-1 are as follows:

- *Client*—The device (workstation) that requests access to the LAN and switch services and responds to requests from the switch. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating system. (The client is the *supplicant* in the IEEE 802.1X specification.)



Note To resolve Windows XP network connectivity and 802.1X port-based authentication issues, read the Microsoft Knowledge Base article at this URL:
<http://support.microsoft.com/support/kb/articles/Q303/5/97.ASP>

- *Authentication server*—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the authentication service is transparent to the client. The Remote Authentication Dial-In User Service (RADIUS) security system with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server, version 3.0. RADIUS uses a client-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.
- *Switch* (also called the *authenticator* and *back-end authenticator*)—Controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client. The switch includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is reencapsulated in the RADIUS format. The EAP frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the server's frame header is removed, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

Authentication Initiation and Message Exchange

The switch or the client can initiate authentication. If you enable authentication on a port by using the **dot1x port-control auto** interface configuration command, the switch must initiate authentication when it determines that the port link state transitions from down to up. The switch then sends an EAP-request/identity frame to the client to request its identity (typically, the switch sends an initial identity/request frame followed by one or more requests for authentication information). When the client receives the frame, it responds with an EAP-response/identity frame.

If the client does not receive an EAP-request/identity frame from the switch during bootup, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.



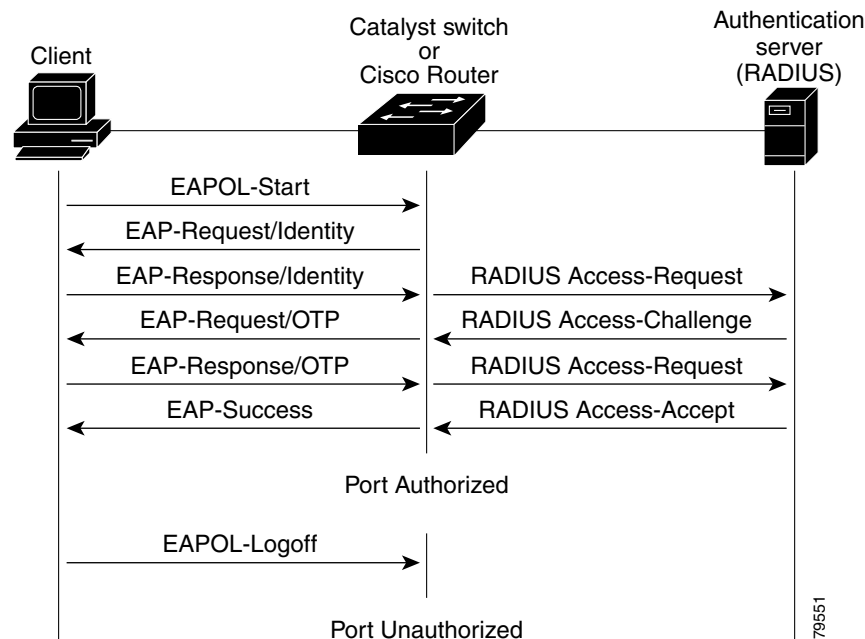
Note

If 802.1X is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client transmits frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated. For more information, see the [“Ports in Authorized and Unauthorized States”](#) section on page 25-4.

When the client supplies its identity, the switch begins its role as the intermediary, passing EAP frames between the client and the authentication server until authentication succeeds or fails. If the authentication succeeds, the switch port becomes authorized. For more information, see the [“Ports in Authorized and Unauthorized States”](#) section on page 25-4.

The specific exchange of EAP frames depends on the authentication method being used. [Figure 25-2](#) shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

Figure 25-2 Message Exchange



Ports in Authorized and Unauthorized States

The switch port state determines whether or not the client is granted access to the network. The port starts in the *unauthorized* state. While in this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a client is successfully authenticated, the port transitions to the *authorized* state, allowing all traffic for the client to flow normally.

If a client that does not support 802.1X is connected to an unauthorized 802.1X port, the switch requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

You control the port authorization state by using the **dot1x port-control** interface configuration command and these keywords:

- **force-authorized**—Disables 802.1X port-based authentication and causes the port to transition to the authorized state without any authentication exchange required. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This is the default setting.
- **force-unauthorized**—Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client through the interface.
- **auto**—Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received. The switch requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each client attempting to access the network is uniquely identified by the switch by using the client's MAC address.

If the client is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated client are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the switch can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and network access is not granted.

When a client logs off, it sends an EAPOL-logoff message, causing the switch port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

Supported Topologies

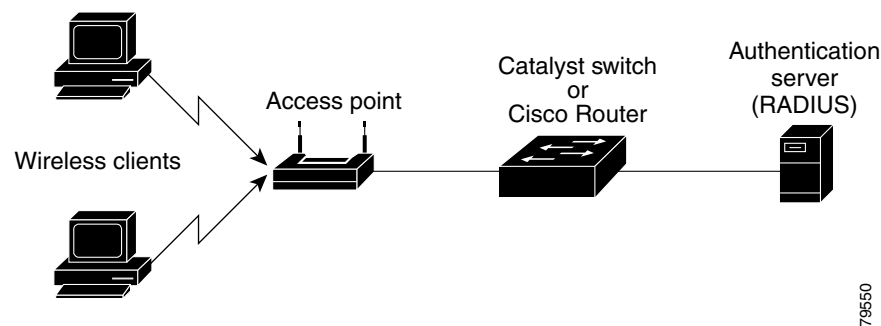
The 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see [Figure 25-1 on page 25-2](#)), only one client can be connected to the 802.1X-enabled switch port. The switch detects the client when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

[Figure 25-3](#) shows 802.1X port-based authentication in a wireless LAN. The 802.1X port is configured as a multiple-host port that becomes authorized as soon as one client is authenticated. When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the switch denies access to the network to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and the wireless access point acts as a client to the switch.

Figure 25-3 Wireless LAN Example



Default 802.1X Port-Based Authentication Configuration

[Table 25-1](#) shows the default 802.1X configuration.

Table 25-1 Default 802.1X Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled
RADIUS server IP address	None specified
RADIUS server UDP authentication port	1812
RADIUS server key	None specified
Per-interface 802.1X protocol enable state	Disabled (force-authorized) Note The port transmits and receives normal traffic without 802.1X-based authentication of the client.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet period	60 seconds (number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client)

Table 25-1 Default 802.1X Configuration (continued)

Feature	Default Setting
Retransmission time	30 seconds (number of seconds that the switch should wait for a response to an EAP request/identity frame from the client before retransmitting the request)
Maximum retransmission number	2 times (number of times that the switch will send an EAP-request/identity frame before restarting the authentication process)
Multiple host support	Disabled
Client timeout period	30 seconds (when relaying a request from the authentication server to the client, the amount of time the switch waits for a response before retransmitting the request to the client)
Authentication server timeout period	30 seconds (when relaying a response from the client to the authentication server, the amount of time the switch waits for a reply before retransmitting the response to the server)

802.1X Port-Based Authentication Guidelines and Restrictions

Follow these guidelines and restrictions when configuring 802.1X port-based authentication:

- When 802.1X is enabled, ports are authenticated before any other Layer 2 or Layer 3 features are enabled.
- The 802.1X protocol is supported on both Layer 2 static-access ports and Layer 3 routed ports, but it is not supported on these port types:
 - Trunk port—If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.
 - EtherChannel port—Before enabling 802.1X on the port, you must first remove it from the EtherChannel port-channel interface. If you try to enable 802.1X on an EtherChannel port-channel interface or on an individual active port in an EtherChannel, an error message appears, and 802.1X is not enabled. If you enable 802.1X on a not-yet active individual port of an EtherChannel, the port does not join the EtherChannel.
 - Secure port—You cannot configure a secure port as an 802.1X port. If you try to enable 802.1X on a secure port, an error message appears, and 802.1X is not enabled. If you try to change an 802.1X-enabled port to a secure port, an error message appears, and the security settings are not changed.
 - Switch Port Analyzer (SPAN) destination port—You can enable 802.1X on a port that is a SPAN destination port; however, 802.1X is disabled until the port is removed as a SPAN destination port. You can enable 802.1X on a SPAN source port.

Configuring 802.1X Port-Based Authentication

These sections describe how to configure 802.1X port-based authentication:

- [Enabling 802.1X Port-Based Authentication, page 25-7](#)
- [Configuring Switch-to-RADIUS-Server Communication, page 25-8](#)
- [Enabling Periodic Reauthentication, page 25-10](#)
- [Manually Reauthenticating the Client Connected to a Port, page 25-11](#)
- [Initializing Authentication for the Client Connected to a Port, page 25-11](#)
- [Changing the Quiet Period, page 25-11](#)
- [Changing the Switch-to-Client Retransmission Time, page 25-12](#)
- [Setting the Switch-to-Client Frame Retransmission Number, page 25-14](#)
- [Enabling Multiple Hosts, page 25-14](#)
- [Resetting the 802.1X Configuration to the Default Values, page 25-15](#)

Enabling 802.1X Port-Based Authentication

To enable 802.1X port-based authentication, you must enable AAA and specify the authentication method list. A method list describes the sequence and authentication methods to be queried to authenticate a user.

The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle, the authentication process stops, and no other authentication methods are attempted.

To configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	Router(config)# aaa new-model	Enables AAA.
	Router(config)# no aaa new-model	Disables AAA.
Step 2	Router(config)# aaa authentication dot1x { default } <i>method1</i> [<i>method2...</i>]	Creates an 802.1X port-based authentication method list.
	Router(config)# no aaa authentication dot1x { default <i>list_name</i> }	Clears the configured method list.
Step 3	Router(config)# dot1x system-auth-control	Globally enables 802.1X port-based authentication.
	Router(config)# no dot1x system-auth-control	Globally disables 802.1X port-based authentication.
Step 4	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Enters interface configuration mode and specifies the interface to be enabled for 802.1X port-based authentication.
Step 5	Router(config-if)# dot1x port-control auto	Enables 802.1X port-based authentication on the interface.
	Router(config-if)# no dot1x port-control auto	Disables 802.1X port-based authentication on the interface.

	Command	Purpose
Step 6	Router(config)# end	Returns to privileged EXEC mode.
Step 7	Router# show dot1x all	Verifies your entries. Check the Status column in the 802.1X Port Summary section of the display. An <i>enabled</i> status means the port-control value is set either to auto or to force-unauthorized .

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When you enable 802.1X port-based authentication, note the following syntax information:

- To create a default list that is used when a named list is *not* specified in the **authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.
- Enter at least one of these keywords:
 - **group radius**—Use the list of all RADIUS servers for authentication.
 - **none**—Use no authentication. The client is automatically authenticated by the switch without using the information supplied by the client.

This example shows how to enable AAA and 802.1X on Fast Ethernet port 5/1:

```
Router# configure terminal
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# dot1x system-auth-control
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show dot1x all

Dot1x Info for interface FastEthernet5/1
-----
AuthSM State      = FORCE UNAUTHORIZED
BendSM State      = IDLE
PortStatus        = UNAUTHORIZED
MaxReq            = 2
MultiHosts        = Disabled
Port Control      = Force Unauthorized
QuietPeriod       = 60 Seconds
Re-authentication = Disabled
ReAuthPeriod      = 3600 Seconds
ServerTimeout     = 30 Seconds
SuppTimeout       = 30 Seconds
TxPeriod          = 30 Seconds
```

Configuring Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by any of the following:

- Host name
- Host IP address

- Host name and specific UDP port numbers
- IP address and specific UDP port numbers

The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service (for example, authentication) the second host entry configured acts as the failover backup to the first one. The RADIUS host entries are tried in the order that they were configured.

To configure the RADIUS server parameters, perform this task:

	Command	Purpose
Step 1	Router(config)# ip radius source-interface <i>interface_name</i>	Specifies that the RADIUS packets have the IP address of the indicated interface.
	Router(config)# no ip radius source-interface	Prevents the RADIUS packets from having the IP address of the previously indicated interface.
Step 2	Router(config)# radius-server host { <i>hostname</i> <i>ip_address</i> }	Configures the RADIUS server host name or IP address on the switch. If you want to use multiple RADIUS servers, reenter this command.
	Router(config)# no radius-server host { <i>hostname</i> <i>ip_address</i> }	Deletes the specified RADIUS server.
Step 3	Router(config)# radius-server key <i>string</i>	Configures the authorization and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 4	Router(config)# end	Returns to privileged EXEC mode.

When you configure the RADIUS server parameters, note the following syntax information:

- For *hostname* or *ip_address*, specify the host name or IP address of the remote RADIUS server.
- Specify the **key string** on a separate command line.
- For **key string**, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.
- When you specify the **key string**, spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.
- You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands. For more information, refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, publication and the *Cisco IOS Security Command Reference*, Release 12.1, publication at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

**Note**

You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch. For more information, refer to the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on the switch:

```
Router# configure terminal
Router(config)# ip radius source-interface Vlan80
Router(config)# radius-server host 172.120.39.46
Router(config)# radius-server key rad123
Router(config)# end
```

Enabling Periodic Reauthentication

You can enable periodic 802.1X client reauthentication and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication attempts is 3600.

Automatic 802.1X client reauthentication is a global setting and cannot be set for clients connected to individual ports. To manually reauthenticate the client connected to a specific port, see the [“Manually Reauthenticating the Client Connected to a Port”](#) section on page 25-11.

To enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x reauthentication	Enables periodic reauthentication of the client, which is disabled by default.
	Router(config-if)# no dot1x reauthentication	Disables periodic reauthentication of the client.
Step 3	Router(config-if)# dot1x timeout re-authperiod <i>seconds</i>	Sets the number of seconds between reauthentication attempts. The range is 1 to 4294967295; the default is 3600 seconds. This command affects the behavior of the switch only if periodic reauthentication is enabled.
	Router(config-if)# no dot1x timeout re-authperiod	Returns to the default reauthorization period.
Step 4	Router(config-if)# end	Returns to privileged EXEC mode.
Step 5	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

```
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout re-authperiod 4000
```

Manually Reauthenticating the Client Connected to a Port



Note Reauthentication does not disturb the status of an already authorized port.

To manually reauthenticate the client connected to a port, perform this task:

	Command	Purpose
Step 1	Router# dot1x re-authenticate interface <i>type</i> ¹ <i>slot/port</i>	Manually reauthenticates the client connected to a port.
Step 2	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to manually reauthenticate the client connected to Fast Ethernet port 5/1:

```
Router# dot1x re-authenticate interface fastethernet 5/1
Starting reauthentication on FastEthernet 5/1
```

Initializing Authentication for the Client Connected to a Port



Note Initializing authentication disables any existing authentication before authenticating the client connected to the port.

To initialize the authentication for the client connected to a port, perform this task:

	Command	Purpose
Step 1	Router# dot1x initialize interface <i>type</i> ¹ <i>slot/port</i>	Initializes the authentication for the client connected to a port.
Step 2	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to initialize the authentication for the client connected to Fast Ethernet port 5/1:

```
Router# dot1x initialize interface fastethernet 5/1
Starting reauthentication on FastEthernet 5/1
```

Changing the Quiet Period

When the switch cannot authenticate the client, the switch remains idle for a set period of time, and then tries again. The idle time is determined by the quiet-period value. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering a smaller number than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout quiet-period <i>seconds</i>	Sets the number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds; the default is 60.
	Router(config-if)# no dot1x timeout quiet-period	Returns to the default quiet time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to set the quiet time on the switch to 30 seconds:

```
Router(config-if)# dot1x timeout quiet-period 30
```

Changing the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the switch with an EAP-response/identity frame. If the switch does not receive this response, it waits a set period of time (known as the retransmission time), and then retransmits the frame.



Note

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

To change the amount of time that the switch waits for client notification, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout tx-period <i>seconds</i>	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65535 seconds; the default is 30.
	Router(config-if)# dot1x timeout tx-period	Returns to the default retransmission time
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to set 60 as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request:

```
Router(config)# dot1x timeout tx-period 60
```

Setting the Switch-to-Client Retransmission Time for EAP-Request Frames

The client notifies the switch that it received the EAP-request frame. If the switch does not receive this notification, the switch waits a set period of time, and then retransmits the frame. You may set the amount of time that the switch waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the switch-to-client retransmission time for the EAP-request frames, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout supp-timeout <i>seconds</i>	Sets the switch-to-client retransmission time for the EAP-request frame.
	Router(config-if)# no dot1x timeout supp-timeout	Returns to the default retransmission time.
Step 3	Router# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to set the switch-to-client retransmission time for the EAP-request frame to 25 seconds:

```
Router(config-if)# dot1x timeout supp-timeout 25
```

Setting the Switch-to-Authentication-Server Retransmission Time for Layer 4 Packets

The authentication server notifies the switch each time it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the switch waits a set period of time and then retransmits the packet. You may set the amount of time that the switch waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the value for the retransmission of Layer 4 packets from the switch to the authentication server, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout server-timeout <i>seconds</i>	Sets the switch-to-authentication-server retransmission time for Layer 4 packets.
	Router(config-if)# no dot1x timeout server-timeout	Returns to the default retransmission time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to set the switch-to-authentication-server retransmission time for Layer 4 packets to 25 seconds:

```
Router(config-if)# dot1x timeout server-timeout 25
```

Setting the Switch-to-Client Frame Retransmission Number

In addition to changing the switch-to-client retransmission time, you can change the number of times that the switch sends an EAP-request/identity frame (assuming no response is received) to the client before restarting the authentication process.



Note

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

To set the switch-to-client frame retransmission number, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x max-req <i>count</i>	Sets the number of times that the switch sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
	Router(config-if)# no dot1x max-req	Returns to the default retransmission number.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to set 5 as the number of times that the switch sends an EAP-request/identity request before restarting the authentication process:

```
Router(config-if)# dot1x max-req 5
```

Enabling Multiple Hosts

You can attach multiple hosts to a single 802.1X-enabled port as shown in [Figure 25-3 on page 25-5](#). In this mode, only one of the attached hosts must be successfully authorized for all hosts to be granted network access. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), all attached clients are denied access to the network.

To allow multiple hosts (clients) on an 802.1X-authorized port that has the **dot1x port-control** interface configuration command set to **auto**, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x multi-hosts	Allows multiple hosts (clients) on an 802.1X-authorized port. Make sure that the dot1x port-control interface configuration command set is set to auto for the specified interface.
	Router(config-if)# no dot1x multi-hosts	Disables multiple hosts on the port.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x interface <i>type</i> ¹ <i>slot/port</i>	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable 802.1X on Fast Ethernet interface 5/1 and to allow multiple hosts:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x multi-hosts
```

Resetting the 802.1X Configuration to the Default Values

To reset the 802.1X configuration to the default values, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x default	Resets the configurable 802.1X parameters to the default values.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

Displaying 802.1X Status

To display global 802.1X administrative and operational status for the switch, use the **show dot1x** privileged EXEC command. To display the 802.1X administrative and operational status for a specific interface, use the **show dot1x interface** *interface-id* privileged EXEC command.

For detailed information about the fields in these displays, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.



Configuring Port Security

This chapter describes how to configure the port security feature. Release 12.1(13)E and later releases support the port security feature.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding Port Security, page 26-1](#)
- [Default Port Security Configuration, page 26-2](#)
- [Port Security Guidelines and Restrictions, page 26-2](#)
- [Configuring Port Security, page 26-2](#)
- [Displaying Port Security Settings, page 26-5](#)

Understanding Port Security

You can use the port security feature to restrict input to an interface by limiting and identifying MAC addresses of the workstations that are allowed to access the port. When you assign secure MAC addresses to a secure port, the port does not forward packets with source addresses outside the group of defined addresses. If you limit the number of secure MAC addresses to one and assign a single secure MAC address, the workstation attached to that port is assured the full bandwidth of the port.

If a port is configured as a secure port and the maximum number of secure MAC addresses is reached, when the MAC address of a workstation attempting to access the port is different from any of the identified secure MAC addresses, a security violation occurs. If a workstation with a secure MAC that is address configured or learned on one secure port attempts to access another secure port, a violation is flagged.

After you have set the maximum number of secure MAC addresses on a port, the secure addresses are included in an address table in one of these ways:

- You can configure all secure MAC addresses by using the **switchport port-security mac-address mac_address** interface configuration command.
- You can allow the port to dynamically configure secure MAC addresses with the MAC addresses of connected devices.
- You can configure a number of addresses and allow the rest to be dynamically configured.

**Note**

If the port shuts down, all dynamically learned addresses are removed.

After the maximum number of secure MAC addresses is configured, they are stored in an address table. To ensure that an attached device has the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

A security violation occurs if the maximum number of secure MAC addresses have been added to the address table and a workstation whose MAC address is not in the address table attempts to access the interface.

You can configure the interface for one of three violation modes: protect, restrict, or shutdown (see the [“Configuring Port Security”](#) section on page 26-2.)

Default Port Security Configuration

Table 26-1 shows the default port security configuration for an interface.

Table 26-1 Default Port Security Configuration

Feature	Default Setting
Port security	Disabled on a port
Maximum number of secure MAC addresses	1
Violation mode	Shutdown. The port shuts down when the maximum number of secure MAC addresses is exceeded, and an SNMP trap notification is sent.

Port Security Guidelines and Restrictions

Follow these guidelines when configuring port security:

- A secure port cannot be a trunk port.
- A secure port cannot be a destination port for Switch Port Analyzer (SPAN).
- A secure port cannot belong to an EtherChannel port-channel interface.
- A secure port cannot be an 802.1X port. If you try to enable 802.1X on a secure port, an error message appears, and 802.1X is not enabled. If you try to change an 802.1X-enabled port to a secure port, an error message appears, and the security settings are not changed.

Configuring Port Security

These sections describe how to configure port security:

- [Configuring Port Security on an Interface, page 26-3](#)
- [Configuring Port Security Aging, page 26-4](#)

Configuring Port Security on an Interface

To restrict traffic through a port by limiting and identifying MAC addresses of the stations allowed to access the port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>interface_id</i>	Enters interface configuration mode and enters the physical interface to configure, for example, gigabitethernet 3/1 .
Step 2	Router(config-if)# switchport mode access	Sets the interface mode as access; an interface in the default mode (dynamic desirable) cannot be configured as a secure port.
Step 3	Router(config-if)# switchport port-security	Enables port security on the interface.
Step 4	Router(config-if)# switchport port-security maximum <i>value</i>	(Optional) Sets the maximum number of secure MAC addresses for the interface. The range is 1 to 128; the default is 128.
Step 5	Router(config-if)# switchport port-security violation { protect restrict shutdown }	(Optional) Sets the violation mode and the action to be taken when a security violation is detected.
Step 6	Router(config-if)# switchport port-security mac-address <i>mac_address</i>	(Optional) Enters a secure MAC address for the interface. You can use this command to enter the maximum number of secure MAC addresses. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned.
Step 7	Router(config-if)# end	Returns to privileged EXEC mode.
Step 8	Router# show port-security interface <i>interface_id</i> Router# show port-security address	Verifies your entries.

When configuring port security, note the following syntax information about port security violation modes:

- **protect**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value.
- **restrict**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value and causes the SecurityViolation counter to increment.
- **shutdown**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.



Note

When port security is enabled, if an address learned or configured on one secure interface is seen on another secure interface in the same VLAN, port security puts the interface into the error-disabled state immediately.

To bring a secure port out of the error-disabled state, enter the **errdisable recovery cause** *pssecure_violation* global configuration command or you can manually reenable it by entering the **shutdown** and **no shut down** interface configuration commands.

To return the interface to the default condition (not a secure port), enter the **no switchport port-security interface** configuration command.

To return the interface to the default number of secure MAC addresses, enter the **no switchport port-security maximum value** command.

To delete a MAC address from the address table, enter the **no switchport port-security mac-address mac_address** command.

To return the violation mode to the default condition (shutdown mode), enter the **no switchport port-security violation {protocol | restrict}** command.

This example shows how to enable port security on Fast Ethernet port 12 and to set the maximum number of secure addresses to 5. The violation mode is the default, and no secure MAC addresses are configured.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 3/12
Router(config-if)# switchport mode access
Router(config-if)# switchport port-security
Router(config-if)# switchport port-security maximum 5
Router(config-if)# end
Router# show port-security interface fastethernet 3/12
Security Enabled:Yes, Port Status:SecureUp
Violation Mode:Shutdown
Max. Addrs:5, Current Addrs:0, Configure Addrs:0
```

This example shows how to configure a secure MAC address on Fast Ethernet port 12 and verify the configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport mode access
Router(config-if)# switchport port-security
Router(config-if)# switchport port-security mac-address 1000.2000.3000
Router(config-if)# end
Router# show port-security address
Secure Mac Address Table
-----
```

Vlan	Mac Address	Type	Ports
1	1000.2000.3000	SecureConfigured	Fa5/12

Configuring Port Security Aging

You can use port security aging to set the aging time for all secure addresses on a port.

Use this feature to remove and add PCs on a secure port without manually deleting the existing secure MAC addresses while still limiting the number of secure addresses on a port.

To configure port security aging, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>interface_id</i>	Enters interface configuration mode for the port on which you want to enable port security aging.
Step 2	Router(config-if)# switchport port-security aging time <i>aging_time</i>	Sets the aging time for the secure port. For <i>time</i> , specify the aging time for this port. All the secure addresses age out exactly after the time (minutes) specified lapses and are removed from the secure address list.
	Router(config-if)# no switchport port-security aging time	Disables aging.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show port security [interface <i>interface_id</i>] [address]	Verifies your entries.

When configuring port security aging, note the following:

- With all releases, you can enter the **no** keyword to disable aging.
- For Release 12.1(19)E and later releases, the valid aging-time range is from 1 to 1440 minutes.
- For releases earlier than Release 12.1(19)E, the valid aging-time range is from 0 to 1440 minutes. You can enter zero to disable aging.

This example shows how to set the aging time as 2 hours for the secure addresses on the Fast Ethernet interface 5/1:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport port-security aging time 120
```

This example shows how to set the aging time as 2 minutes:

```
Router(config-if)# switchport port-security aging time 2
```

You can verify the previous commands by entering the **show port-security interface** *interface_id* privileged EXEC command.

Displaying Port Security Settings

The **show interfaces** *interface_id* **switchport** privileged EXEC command displays the interface traffic suppression and control configuration. The **show interfaces counters** privileged EXEC commands display the count of discarded packets. The **show storm control** and **show port-security** privileged EXEC commands display those features.

To display traffic control information, enter one or more of these commands:

Command	Purpose
Router# show port-security [interface <i>interface_id</i>]	Displays port security settings for the switch or for the specified interface, including the maximum allowed number of secure MAC addresses for each interface, the number of secure MAC addresses on the interface, the number of security violations that have occurred, and the violation mode.
Router# show port-security [interface <i>interface_id</i>] address	Displays all secure MAC addresses configured on all switch interfaces or on a specified interface with aging information for each address.

This example displays output from the **show port-security** command when you do not enter an interface:

```
Router# show port-security
Secure Port      MaxSecureAddr  CurrentAddr  SecurityViolation  Security
Action
                (Count)        (Count)      (Count)
-----
      Fa5/1         11           11           0           Shutdown
      Fa5/5         15           5            0           Restrict
      Fa5/11        5            4            0           Protect
-----

Total Addresses in System: 21
Max Addresses limit in System: 128
```

This example displays output from the **show port-security** command for a specified interface:

```
Router# show port-security interface fastethernet 5/1
Port Security: Enabled
Port status: SecureUp
Violation mode: Shutdown
Maximum MAC Addresses: 11
Total MAC Addresses: 11
Configured MAC Addresses: 3
Aging time: 20 mins
Aging type: Inactivity
SecureStatic address aging: Enabled
Security Violation count: 0
```

This example displays output from the **show port-security address** privileged EXEC command:

```
Router# show port-security address
      Secure Mac Address Table
-----
Vlan   Mac Address      Type                Ports      Remaining Age
-----
      1   0001.0001.0001  SecureDynamic      Fa5/1      15 (I)
      1   0001.0001.0002  SecureDynamic      Fa5/1      15 (I)
      1   0001.0001.1111  SecureConfigured   Fa5/1      16 (I)
      1   0001.0001.1112  SecureConfigured   Fa5/1      -
      1   0001.0001.1113  SecureConfigured   Fa5/1      -
      1   0005.0005.0001  SecureConfigured   Fa5/5      23
      1   0005.0005.0002  SecureConfigured   Fa5/5      23
      1   0005.0005.0003  SecureConfigured   Fa5/5      23
      1   0011.0011.0001  SecureConfigured   Fa5/11     25 (I)
      1   0011.0011.0002  SecureConfigured   Fa5/11     25 (I)
-----

Total Addresses in System: 10
Max Addresses limit in System: 128
```



Configuring Layer 3 Protocol Filtering on Supervisor Engine 1



Note

Layer 3 protocol filtering is supported with Supervisor Engine 1. Layer 3 protocol filtering is not supported with Supervisor Engine 2.

This chapter describes how to configure Layer 3 protocol filtering on Layer 2 LAN ports on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How Layer 3 Protocol Filtering Works, page 27-1](#)
- [Configuring Layer 3 Protocol Filtering, page 27-2](#)

Understanding How Layer 3 Protocol Filtering Works

Layer 3 protocol filtering prevents specific Layer 3 protocol packets from being received or transmitted on a Layer 2 LAN port, which reduces the broadcast domain of specific protocols in a VLAN. For example, you can configure a Layer 2 LAN port in a VLAN to allow IP packets only, while another Layer 2 LAN port in the same VLAN allows both IP and Internetwork Packet Exchange (IPX) packets.

Layer 2 LAN trunk ports do not support protocol filtering. You can configure Layer 3 protocol filtering on a trunk, but the configuration is ignored while the port is a trunk.

Protocol filtering cannot be configured on Layer 3 interfaces—only nontrunk Layer 2 LAN ports support Layer 3 protocol filtering.

Layer 3 protocol filtering does not support the features available with standard and extended Cisco IOS ACLs.

Layer 2 protocols, such as Spanning Tree Protocol (STP) and Cisco Discovery Protocol (CDP), are not affected by Layer 3 protocol filtering. Layer 2 LAN ports that have port security enabled are members of all protocol groups.

You can configure a Layer 2 LAN port with any one of these modes for each protocol group: **on**, **off**, or **auto**. If the configuration is set to **on**, the port allows all traffic for that protocol. If the configuration is set to **off**, the port does not allow any traffic for that protocol.

If the configuration is set to **auto**, the Layer 2 LAN port initially does not allow any flood traffic to be transmitted from the port. After a packet is received on that port, the port will transmit traffic for that protocol group. Once in this state, the port reverts back to allowing flood traffic to be transmitted if no packets for that protocol have been received for 60 minutes. Layer 2 LAN ports are also removed from the protocol group when the supervisor engine detects that the link is down on the port.

If a host that supports both IP and IPX is connected to a Layer 2 LAN port configured as **auto** for IPX, but the host is transmitting only IP traffic, the port to which the host is connected will not transmit any flooded IPX traffic. However, if the host sends an IPX packet, the supervisor engine software detects the protocol traffic and the port begins transmitting flooded IPX traffic. If the host stops sending IPX traffic for more than 60 minutes, the port stops transmitting flooded IPX traffic.

By default, Layer 2 LAN ports are configured to **on** for all protocol groups. Typically, you should only configure a Layer 2 LAN port to **auto** for IP if an end station is directly connected to the port.

Protocol filters are configured according to groups of protocols, not specific protocols. There are four groups of protocols defined:

- IP
- IPX
- AppleTalk, DECnet, and Banyan VINES (“group”)
- Packets not belonging to any of these protocols (“other”)

Configuring Layer 3 Protocol Filtering

These sections describe how to configure Layer 3 protocol filtering on Ethernet-type VLANs and on any type of Layer 2 LAN port:

- [Enabling Layer 3 Protocol Filtering, page 27-2](#)
- [Configuring Layer 3 Protocol Filtering on a Layer 2 LAN Interface, page 27-3](#)
- [Verifying Layer 3 Protocol Filtering Configuration, page 27-3](#)



Note

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling Layer 3 Protocol Filtering

To enable Layer 3 protocol filtering globally, perform this task:

Command	Purpose
Router(config)# protocol-filter	Enables Layer 3 protocol filtering globally.
Router(config)# no protocol-filter	Disables Layer 3 protocol filtering globally.

This example shows how to enable Layer 3 protocol filtering globally:

```
Router# configure terminal
Router(config)# protocol-filtering
```

Configuring Layer 3 Protocol Filtering on a Layer 2 LAN Interface

To configure Layer 3 protocol filtering on a Layer 2 LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# switchport protocol {appletalk ip ipx group} {on off auto}	Configures Layer 3 protocol filtering on the LAN port.
	Router(config-if)# no switchport protocol {appletalk ip ipx group}	Clears Layer 3 protocol filtering configuration on the LAN port.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure the protocol membership of Fast Ethernet port 5/8 to allow IPX packets only, and verify the configuration:

```
Router(config)# interface fastethernet 5/8
Router(config-if)# switchport protocol appletalk off
Router(config-if)# switchport protocol ip off
Router(config-if)# switchport protocol ipx on
```

Verifying Layer 3 Protocol Filtering Configuration

To verify Layer 3 protocol filtering configuration, perform this task:

Command	Purpose
Router# show protocol-filtering interface {{type ¹ slot/port} {port-channel number}}	Verifies the interface filtering configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to verify the Layer 3 protocol filtering configuration of Fast Ethernet port 5/8:

```
Router# show protocol-filtering interface fastethernet 5/8
Interface      IP Mode      IPX Mode      Group Mode      Other Mode
-----
Fa5/8          OFF          ON            OFF            OFF
Router#
```



Note

The **show protocol filtering** command shows only ports that have at least one protocol set to the nondefault configuration.



Configuring Traffic Storm Control

This chapter describes how to configure the traffic storm control feature on the Catalyst 6500 series switches. Release 12.1(12c)E1 and later releases support traffic storm control. For earlier releases, refer to [Chapter 29, “Configuring Broadcast Suppression.”](#)



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.
 - The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX switching modules do not support traffic storm control.
-

This chapter consists of these sections:

- [Understanding Traffic Storm Control, page 28-1](#)
- [Default Traffic Storm Control Configuration, page 28-2](#)
- [Enabling Traffic Storm Control, page 28-2](#)

Understanding Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. The traffic storm control feature prevents LAN ports from being disrupted by a broadcast, multicast, or unicast traffic storm on physical interfaces.

Traffic storm control (also called traffic suppression) monitors incoming traffic levels over a 1-second traffic storm control interval and, during the interval, compares the traffic level with the traffic storm control level that you configure. The traffic storm control level is a percentage of the total available bandwidth of the port. Each port has a single traffic storm control level that is used for all types of traffic (broadcast, multicast, and unicast).



Note

- The switch supports multicast and unicast traffic storm control only on Gigabit Ethernet LAN ports.
 - The switch supports broadcast traffic storm control on all LAN ports.
 - Traffic storm control does not suppress spanning tree packets. Except for spanning tree packets, traffic storm control does not differentiate between control traffic and data traffic.
-

Traffic storm control monitors the level of each traffic type for which you enable traffic storm control in 1-second traffic storm control intervals. Within an interval, when the ingress traffic for which traffic storm control is enabled reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the traffic storm control interval ends.

The following are examples of traffic storm control behavior:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and the combined broadcast and multicast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.

Default Traffic Storm Control Configuration

Traffic storm control is disabled by default.

Enabling Traffic Storm Control

To enable traffic storm control, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects an interface to configure.
Step 2	Router(config-if)# storm-control broadcast level level[.level]	Enables broadcast traffic storm control on the interface, configures the traffic storm control level, and applies the traffic storm control level to all traffic storm control modes enabled on the interface.
	Router(config-if)# no storm-control broadcast level	Disables broadcast traffic storm control on the interface.
Step 3	Router(config-if)# storm-control multicast level level[.level]	Enables multicast traffic storm control on the interface, configures the traffic storm control level, and applies the traffic storm control level to all traffic storm control modes enabled on the interface.
	<p>Note The storm-control multicast command is supported only on Gigabit Ethernet interfaces.</p> Router(config-if)# no storm-control multicast level	Disables multicast traffic storm control on the interface.

	Command	Purpose
Step 4	Router(config-if)# storm-control unicast level <i>level[.level]</i>	Enables unicast traffic storm control on the interface, configures the traffic storm control level, and applies the traffic storm control level to all traffic storm control modes enabled on the interface.
	Note The storm-control unicast command is supported only on Gigabit Ethernet interfaces.	
	Router(config-if)# no storm-control unicast level	Disables unicast traffic storm control on the interface.
Step 5	Router(config-if)# end	Exits configuration mode.
Step 6	Router# show running-config interface	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When configuring the traffic storm control level, note the following:

- You can configure traffic storm control on an EtherChannel (a port channel interface).
- Do not configure traffic storm control on ports that are members of an EtherChannel. Configuring traffic storm control on ports that are configured as members of an EtherChannel puts the ports into a suspended state.
- Specify the level as a percentage of the total interface bandwidth:
 - The level can be from 0 to 100.
 - The optional fraction of a level can be from 0 to 99.
 - 100 percent means no traffic storm control.
 - 0.0 percent suppresses all traffic.

Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames making up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.

This example shows how to enable multicast traffic storm control on Gigabit Ethernet interface 3/16 and how to configure the traffic storm control level at 70.5 percent. This configuration applies the traffic storm control level to all traffic storm control modes enabled on Gigabit Ethernet interface 3/16:

```
Router# configure terminal
Router(config)# interface gigabitethernet 3/16
Router(config-if)# storm-control multicast level 70.5
Router(config-if)# end
```

Displaying Traffic Storm Control Settings

To display traffic storm control information, use the commands described in [Table 28-1](#).

Table 28-1 Commands for Displaying Traffic Storm Control Status and Configuration

Command	Purpose
Router# show interfaces [{type ¹ slot/port} {port-channel number}] switchport	Displays the administrative and operational status of all Layer 2 LAN ports or the specified Layer 2 LAN port.
Router# show interfaces [{type ¹ slot/port} {port-channel number}] counters broadcast Router# show interfaces [{type ¹ slot/port} {port-channel number}] counters multicast Router# show interfaces [{type ¹ slot/port} {port-channel number}] counters unicast	There is a single counter for all suppressed traffic. These commands all display the same discard count, which shows the total number of packets discarded for all three traffic storm control modes, on all interfaces or on the specified interface.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet



Note

The **show interfaces** [{interface_type slot/port} | {port-channel number}] **counters** command does not display the discard count. You must use one of the traffic-type keywords: **broadcast**, **multicast**, or **unicast**, which all display the same discard count.



Configuring Broadcast Suppression

This chapter describes how to configure broadcast suppression on the Catalyst 6500 series switches. Releases earlier than Release 12.1(12c)E1 support broadcast suppression. Use traffic storm control with Release 12.1(12c)E1 and later releases (see [Chapter 28, “Configuring Traffic Storm Control”](#)).



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How Broadcast Suppression Works, page 29-1](#)
- [Broadcast Suppression Configuration Guidelines and Restrictions, page 29-2](#)
- [Enabling Broadcast Suppression, page 29-3](#)

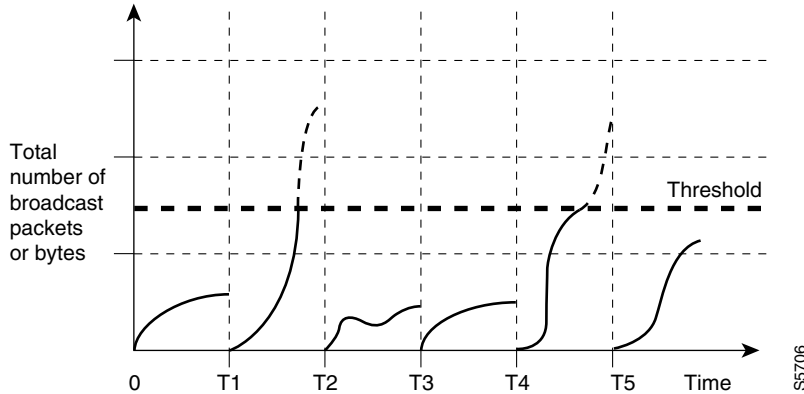
Understanding How Broadcast Suppression Works

Broadcast suppression prevents LAN interfaces from being disrupted by a broadcast storm. A broadcast storm occurs when broadcast or multicast packets flood the subnet, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can cause a broadcast storm.

Broadcast suppression uses filtering that measures broadcast activity in a subnet over a 1-second interval and compares the measurement with a predefined threshold. If the threshold is reached, further broadcast activity is suppressed for the duration the interval. Broadcast suppression is disabled by default.

[Figure 29-1](#) shows the broadcast traffic patterns on a LAN interface over a given interval. In this example, broadcast suppression occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 29-1 Broadcast Suppression



The broadcast suppression threshold numbers and the time interval combination make the broadcast suppression algorithm work with different levels of granularity. A higher threshold allows more broadcast packets to pass through.

Broadcast suppression on the Catalyst 6500 series switches is implemented in hardware. The suppression circuitry monitors packets passing from a LAN interface to the switching bus. Using the Individual/Group bit in the packet destination address, the broadcast suppression circuitry determines if the packet is unicast or broadcast, keeps track of the current count of broadcasts within the 1-second interval, and when a threshold is reached, filters out subsequent broadcast packets.

Because hardware broadcast suppression uses a bandwidth-based method to measure broadcast activity, the most significant implementation factor is setting the percentage of total available bandwidth that can be used by broadcast traffic. Because packets do not arrive at uniform intervals, the 1-second interval during which broadcast activity is measured can affect the behavior of broadcast suppression.

**Note**

Multicast traffic cannot be suppressed.

Broadcast Suppression Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring broadcast suppression:

- Broadcast suppression is supported on Layer 2 and Layer 3 LAN interfaces.
- Broadcast suppression is not supported on VLAN interfaces.
- You can specify broadcast suppression in hundredths of a percent.
- A threshold value of 0 suppresses all broadcast traffic.
- A threshold value of 100 percent does not suppress any broadcast traffic.
- The broadcast suppression configuration is cleared if you change the configuration of a LAN interface from Layer 3 to Layer 2 or from Layer 2 to Layer 3.

**Note**

With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling Broadcast Suppression

To enable broadcast suppression, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects an interface to configure.
Step 2	Router(config-if)# broadcast suppression threshold	Enables broadcast suppression.
	Router(config-if)# no broadcast suppression	Disables broadcast suppression.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show running-config interface	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When enabling broadcast suppression, you can specify the threshold in hundredths of a percent:

- Enter 0.00 to suppress all broadcasts.
- Enter 0.01 for 0.01% (1/100th percent).
- Enter 0.50 for 0.50% (one-half percent).
- Enter 1 or 1.00 for 1% (one percent).

The threshold range is 0.00–100.00.

This example shows how to enable one-quarter-percent broadcast suppression on interface FastEthernet 3/1 and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 3/1
Router(config-if)# broadcast suppression 0.25
Router(config-if)# end
Router# show running-config interface fastethernet 3/1 | include suppression
broadcast suppression 0.25
Router#
```




Configuring CDP

This chapter contains information about how to configure Cisco Discovery Protocol (CDP) on the Catalyst 6500 series switches, which supplements the information in these publications:

- The *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.1, “Cisco IOS System Management,” “Configuring Cisco Discovery Protocol (CDP)” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/fun_c/fcprt3/fcd301c.htm
- The *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.1, “Cisco IOS System Management Commands,” “CDP Commands” publication at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgr/fun_r/frprt3/frd3001b.htm

This chapter consists of these sections:

- [Understanding How CDP Works, page 30-1](#)
- [Configuring CDP, page 30-1](#)

Understanding How CDP Works

CDP is a protocol that runs over Layer 2 (the data link layer) on all Cisco routers, bridges, access servers, and switches. CDP allows network management applications to discover Cisco devices that are neighbors of already known devices, in particular, neighbors running lower-layer, transparent protocols. With CDP, network management applications can learn the device type and the SNMP agent address of neighboring devices. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all LAN and WAN media that support Subnetwork Access Protocol (SNAP).

Each CDP-configured device sends periodic messages to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain the time-to-live, or holdtime information, which indicates the length of time a receiving device should hold CDP information before discarding it.

Configuring CDP

These sections describe how to configure CDP:

- [Enabling CDP Globally, page 30-2](#)
- [Displaying the CDP Global Configuration, page 30-2](#)
- [Enabling CDP on a Port, page 30-2](#)

- [Displaying the CDP Interface Configuration, page 30-3](#)
- [Monitoring and Maintaining CDP, page 30-3](#)

**Note**

With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling CDP Globally

To enable CDP globally, perform this task:

Command	Purpose
Router(config)# cdp run	Enables CDP globally.
Router(config)# no cdp run	Disables CDP globally.

This example shows how to enable CDP globally:

```
Router(config)# cdp run
```

Displaying the CDP Global Configuration

To display the CDP configuration, perform this task:

Command	Purpose
Router# show cdp	Displays global CDP information.

This example shows how to display the CDP configuration:

```
Router# show cdp
Global CDP information:
  Sending CDP packets every 120 seconds
  Sending a holdtime value of 180 seconds
  Sending CDPv2 advertisements is enabled
Router#
```

For additional CDP show commands, see the “[Monitoring and Maintaining CDP](#)” section on page 30-3.

Enabling CDP on a Port

To enable CDP on a port, perform this task:

Command	Purpose
Step 1 Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects the port to configure.

Command	Purpose
Step 2 Router(config-if)# cdp enable	Enables CDP on the port.
Router(config-if)# no cdp enable	Disables CDP on the port.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable CDP on Fast Ethernet port 5/1:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# cdp enable
```

Displaying the CDP Interface Configuration

To display the CDP configuration for a port, perform this task:

Command	Purpose
Router# show cdp interface [{{type ¹ slot/port} {port-channel number}}]	Displays information about ports where CDP is enabled.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to display the CDP configuration of Fast Ethernet port 5/1:

```
Router# show cdp interface fastethernet 5/1
FastEthernet5/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 120 seconds
  Holdtime is 180 seconds
Router#
```

Monitoring and Maintaining CDP

To monitor and maintain CDP on your device, perform one or more of these tasks:

Command	Purpose
Router# clear cdp counters	Resets the traffic counters to zero.
Router# clear cdp table	Clears information about neighbors from the CDP table.
Router# show cdp	Displays global information such as frequency of transmissions and the holdtime for packets being transmitted.
Router# show cdp entry <i>entry_name</i> [protocol version]	Displays information about a specific neighbor. The display can be limited to protocol or version information.
Router# show cdp interface [<i>type¹ slot/port</i>]	Displays information about interfaces on which CDP is enabled.
Router# show cdp neighbors [<i>type¹ slot/port</i>] [detail]	Displays information about neighbors. The display can be limited to neighbors on a specific interface and expanded to provide more detailed information.

Command	Purpose
Router# show cdp traffic	Displays CDP counters, including the number of packets sent and received and checksum errors.
Router# show debugging	Displays information about the types of debugging that are enabled. Refer to the <i>Debug Command Reference</i> publication for more information about CDP debug commands.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to clear CDP counter configuration:

```
Router# clear cdp counters
```

This example shows how to display information about the neighboring equipment:

```
Router# show cdp neighbors
```

```
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
```

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
JAB023807H1	Fas 5/3	127	T S	WS-C2948	2/46
JAB023807H1	Fas 5/2	127	T S	WS-C2948	2/45
JAB023807H1	Fas 5/1	127	T S	WS-C2948	2/44
JAB023807H1	Gig 1/2	122	T S	WS-C2948	2/50
JAB023807H1	Gig 1/1	122	T S	WS-C2948	2/49
JAB03130104	Fas 5/8	167	T S	WS-C4003	2/47
JAB03130104	Fas 5/9	152	T S	WS-C4003	2/48



Configuring PFC QoS

This chapter describes how to configure quality of service (QoS) as implemented on the policy feature card (PFC) on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this publication, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter contains these sections:

- [Understanding How PFC QoS Works, page 31-1](#)
- [PFC QoS Default Configuration, page 31-25](#)
- [PFC QoS Configuration Guidelines and Restrictions, page 31-31](#)
- [Configuring PFC QoS, page 31-33](#)



Note

-
- With Release 12.1(13)E and later releases and with an MSFC2, you can configure Network-Based Application Recognition (NBAR) on Layer 3 interfaces instead of using PFC QoS.
 - All ingress and egress traffic on an interface that is configured with NBAR is processed in software on the MSFC2.
 - The PFC2 provides hardware support for input ACLs on ports where you configure NBAR.
 - When PFC QoS is enabled, the traffic through ports where you configure NBAR passes through the ingress and egress queues and drop thresholds. When PFC QoS is enabled, the MSFC2 sets egress CoS equal to egress IP precedence.
 - After passing through an ingress queue, all traffic is processed in software on the MSFC2 on interfaces where you configure NBAR.
 - To configure NBAR, refer to this publication:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm>

Understanding How PFC QoS Works

Typically, networks operate on a *best-effort* delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

QoS selects network traffic (both unicast and multicast), prioritizes it according to its relative importance, and uses congestion avoidance to provide priority-indexed treatment; QoS can also limit the bandwidth used by network traffic. QoS makes network performance more predictable and bandwidth utilization more effective.

**Note**

On the Catalyst 6500 series switches, queue architecture and QoS queueing features such as Weighted-Round Robin (WRR) and Weighted Random Early Detection (WRED) are implemented with a fixed configuration in Application Specific Integrated Circuits (ASICs). The queueing architecture cannot be reconfigured. For more information, see the “Receive Queues” section on page 31-13 and the “Transmit Queues” section on page 31-21.

These sections describe PFC QoS:

- [Hardware Supported by PFC QoS, page 31-2](#)
- [QoS Terminology, page 31-3](#)
- [PFC QoS Feature Flowcharts, page 31-6](#)
- [PFC QoS Feature Summary, page 31-11](#)
- [Ingress LAN Port Features, page 31-12](#)
- [PFC Marking and Policing, page 31-16](#)
- [LAN Egress Port Features, page 31-21](#)
- [PFC QoS Statistics Data Export, page 31-24](#)

Hardware Supported by PFC QoS

With Release 12.1(11a)E and later, PFC QoS supports both LAN ports and optical services module (OSM) ports:

- *LAN ports* are Ethernet ports on Ethernet switching modules, except for the 4-port Gigabit Ethernet WAN (GBIC) module (OSM-4GE-WAN). Except for the OSM-4GE-WAN module, OSMs have four Ethernet LAN ports in addition to WAN ports. With earlier releases, PFC QoS supports only LAN ports.
- *OSM ports* are the WAN ports on OSMs. The PFC provides ingress QoS for traffic from OSM ports. For more information, see the following sections:
 - [“Ingress OSM Port Features” section on page 31-11](#)
 - [“Egress OSM Port Features” section on page 31-12](#)
 - [“PFC Marking and Policing” section on page 31-16](#)
 - [“Attaching Policy Maps” section on page 31-21](#)
 - [“Configuring the Trust State of Ethernet LAN and OSM Ingress Ports” section on page 31-53](#)
- Refer to the following publication for information about additional OSM QoS features:
http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/osm_inst/index.htm

- The PFC does not provide QoS for FlexWAN module ports. Refer to the following publications for information about FlexWAN module QoS features:
 - *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm
 - *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm
 - *Class-Based Marking*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm>
 - *Traffic Policing*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtpoli.htm>
 - *Distributed Class-Based Weighted Fair Queueing and Distributed Weighted Random Early Detection*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtcbwred.htm>
 - *Distributed Low Latency Queueing*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtlqvip.htm>
 - *Configuring Burst Size in Low Latency Queueing*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtcfbst.htm>
 - *Distributed Traffic Shaping*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdts.htm>
 - MPLS QoS:
http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/osm_inst/mpls.htm

QoS Terminology

This section defines some QoS terminology:

- *Packets* carry traffic at Layer 3.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *Labels* are prioritization values carried in Layer 3 packets and Layer 2 frames:
 - Layer 2 class of service (CoS) values, which range between zero for low priority and seven for high priority:

Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p CoS value in the three least significant bits.

Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most significant bits, which are called the User Priority bits.

Other frame types cannot carry Layer 2 CoS values.



Note On LAN ports configured as Layer 2 ISL trunks, all traffic is in ISL frames. On LAN ports configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

- Layer 3 IP precedence values—The IP version 4 specification defines the three most significant bits of the 1-byte Type of Service (ToS) field as IP precedence. IP precedence values range between zero for low priority and seven for high priority.
- Layer 3 differentiated services code point (DSCP) values—The Internet Engineering Task Force (IETF) has defined the six most significant bits of the 1-byte IP ToS field as the DSCP. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63 (see the [“Configuring DSCP Value Maps”](#) section on page 31-66).



Note Layer 3 IP packets can carry either an IP precedence value or a DSCP value. PFC QoS supports the use of either value, since DSCP values are backwards compatible with IP precedence values (see [Table 31-1 on page 31-5](#)).

- *Classification* is the selection of traffic to be marked.
- *Marking*, according to RFC 2475, is the process of setting a Layer 3 DSCP value in a packet; in this publication, the definition of marking is extended to include setting Layer 2 CoS values.
- *Scheduling* is the assignment of Layer 2 frames to a queue. PFC QoS assigns frames to a queue based on Layer 2 CoS values.
- *Congestion avoidance* is the process by which PFC QoS reserves ingress and egress LAN port capacity for Layer 2 frames with high-priority Layer 2 CoS values. PFC QoS implements congestion avoidance with Layer 2 CoS value-based drop thresholds. A drop threshold is the percentage of queue buffer utilization above which frames with a specified Layer 2 CoS value is dropped, leaving the buffer available for frames with higher-priority Layer 2 CoS values.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing is done on the Policy Feature Card (PFC) or on the Policy Feature Card 2 (PFC2) and distributed forwarding cards (DFCs). Policing can mark or drop traffic.

Table 31-1 IP Precedence and DSCP Values

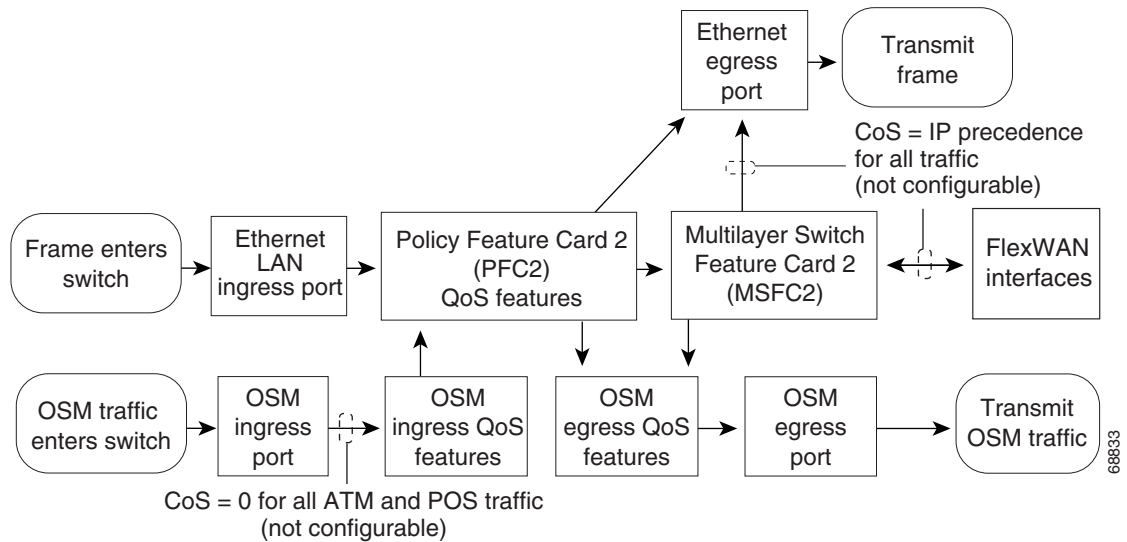
3-bit IP Precedence	6 MSb ¹ of ToS					6-bit DSCP	
	8	7	6	5	4		3
0	0	0	0	0	0	0	0
	0	0	0	0	0	1	1
	0	0	0	0	1	0	2
	0	0	0	0	1	1	3
	0	0	0	1	0	0	4
	0	0	0	1	0	1	5
	0	0	0	1	1	0	6
	0	0	0	1	1	1	7
1	0	0	1	0	0	0	8
	0	0	1	0	0	1	9
	0	0	1	0	1	0	10
	0	0	1	0	1	1	11
	0	0	1	1	0	0	12
	0	0	1	1	0	1	13
	0	0	1	1	1	0	14
	0	0	1	1	1	1	15
2	0	1	0	0	0	0	16
	0	1	0	0	0	1	17
	0	1	0	0	1	0	18
	0	1	0	0	1	1	19
	0	1	0	1	0	0	20
	0	1	0	1	0	1	21
	0	1	0	1	1	0	22
	0	1	0	1	1	1	23
3	0	1	1	0	0	0	24
	0	1	1	0	0	1	25
	0	1	1	0	1	0	26
	0	1	1	0	1	1	27
	0	1	1	1	0	0	28
	0	1	1	1	0	1	29
	0	1	1	1	1	0	30
	0	1	1	1	1	1	31
4	1	0	0	0	0	0	32
	1	0	0	0	0	1	33
	1	0	0	0	1	0	34
	1	0	0	0	1	1	35
	1	0	0	1	0	0	36
	1	0	0	1	0	1	37
	1	0	0	1	1	0	38
	1	0	0	1	1	1	39
5	1	0	1	0	0	0	40
	1	0	1	0	0	1	41
	1	0	1	0	1	0	42
	1	0	1	0	1	1	43
	1	0	1	1	0	0	44
	1	0	1	1	0	1	45
	1	0	1	1	1	0	46
	1	0	1	1	1	1	47
6	1	1	0	0	0	0	48
	1	1	0	0	0	1	49
	1	1	0	0	1	0	50
	1	1	0	0	1	1	51
	1	1	0	1	0	0	52
	1	1	0	1	0	1	53
	1	1	0	1	1	0	54
	1	1	0	1	1	1	55
7	1	1	1	0	0	0	56
	1	1	1	0	0	1	57
	1	1	1	0	1	0	58
	1	1	1	0	1	1	59
	1	1	1	1	0	0	60
	1	1	1	1	0	1	61
	1	1	1	1	1	0	62
	1	1	1	1	1	1	63

1. MSb = most significant bit

PFC QoS Feature Flowcharts

Figure 31-1 show how traffic flows through the components that support PFC QoS features.

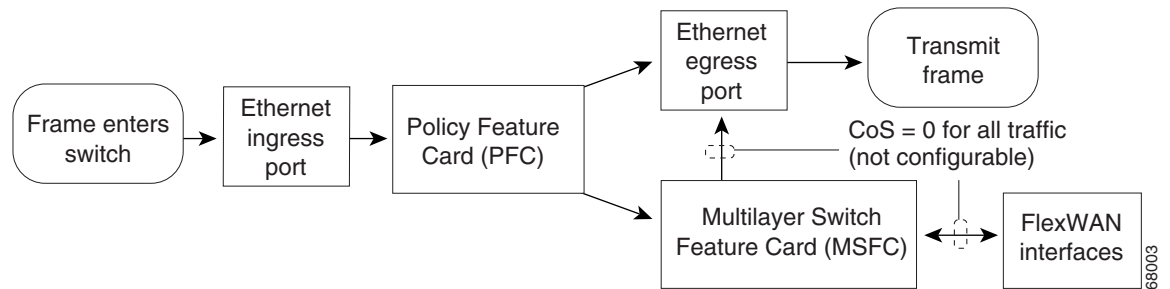
Figure 31-1 Traffic Flow Through PFC QoS Features with PFC2



Note

PFC QoS supports traffic from OSMs with Release 12.1(11a)E and later.

Figure 31-2 Traffic Flow Through PFC QoS Features with PFC



Note

- The PFC can provide Layer 3 switching for FlexWAN traffic but does not provide PFC QoS for FlexWAN traffic.
- PFC QoS does not change the ToS byte in FlexWAN ingress traffic.
- Traffic that is Layer 3-switched does not go through the MSFC and retains the Layer 2 CoS value assigned by the PFC.

Figure 31-3 through Figure 31-8 show how the PFC QoS features are implemented on the switch components.

Figure 31-3 Ingress LAN Port Layer 2 PFC QoS Features

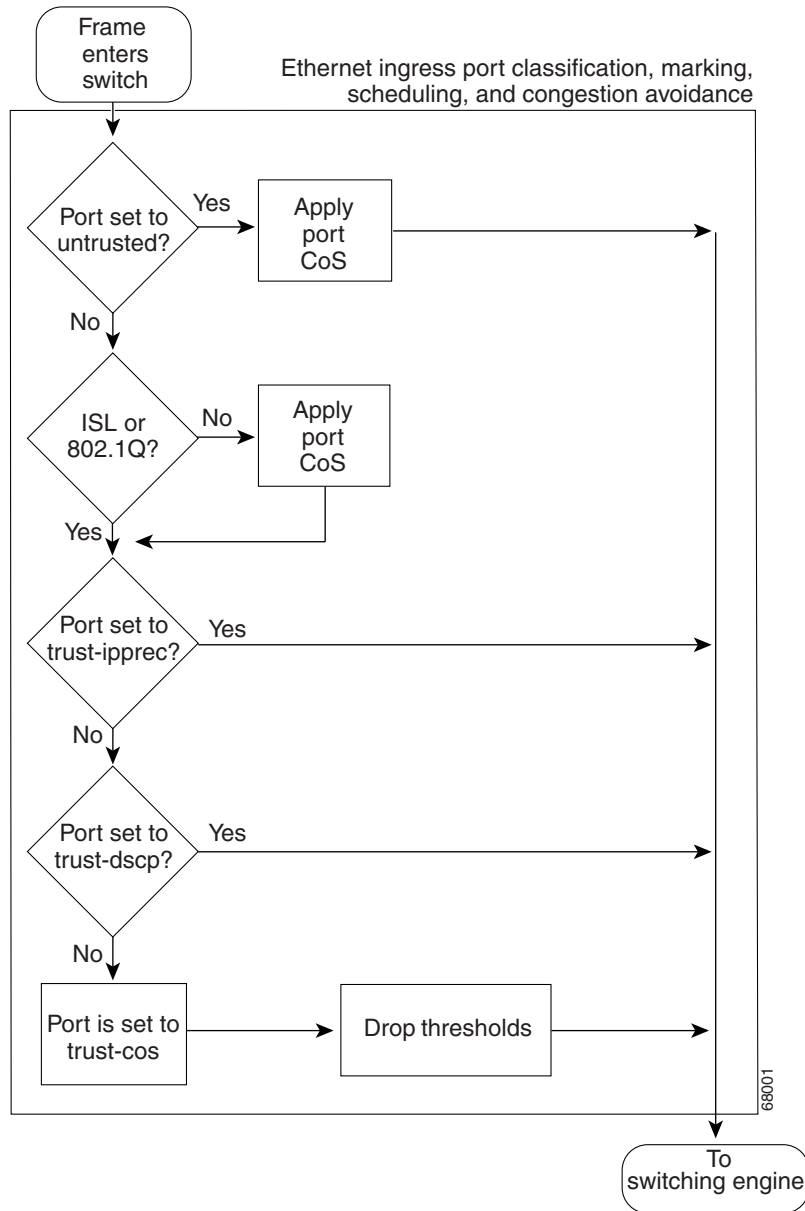


Figure 31-4 PFC Classification, Marking, and Policing

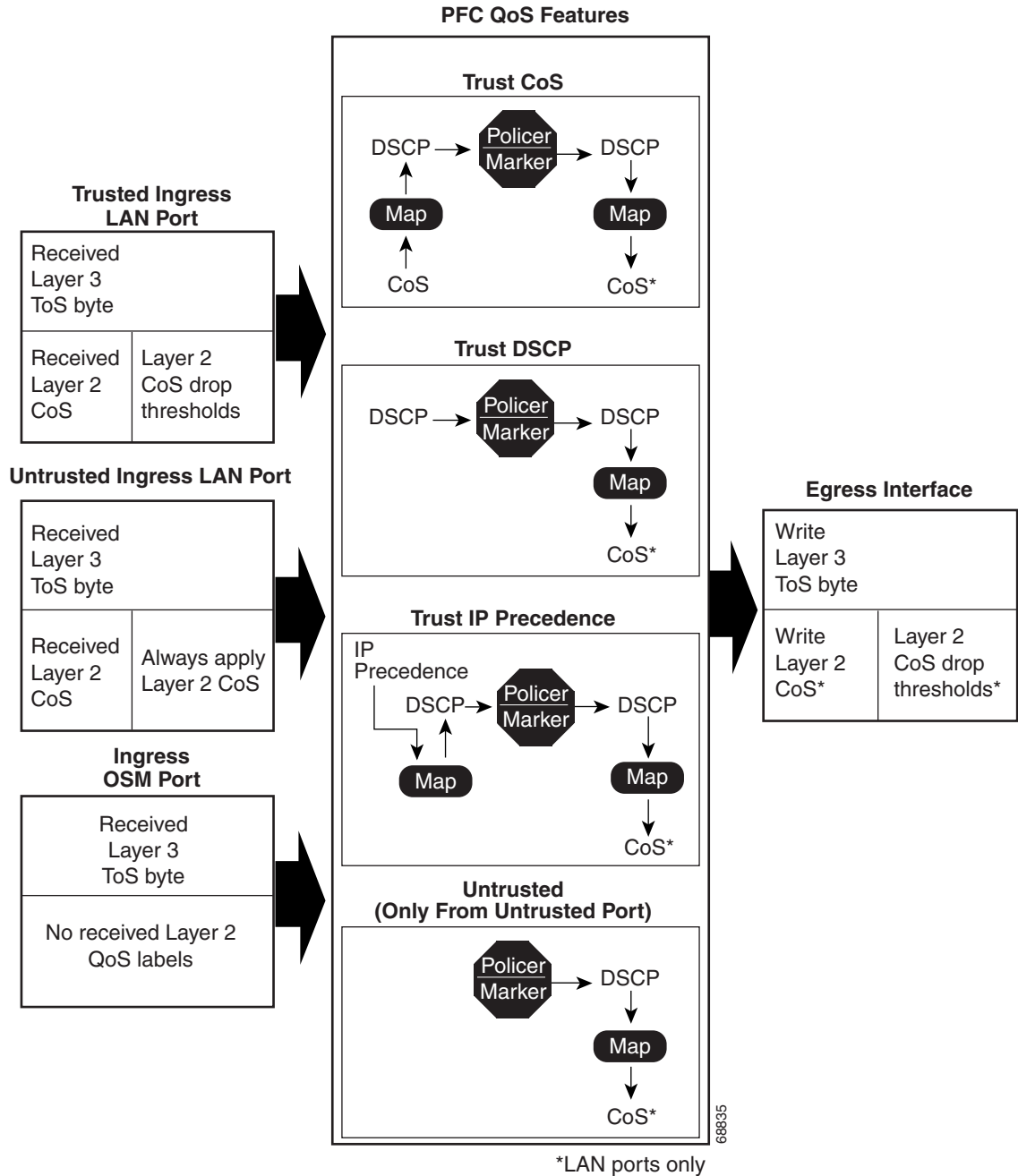


Figure 31-5 Marking with PFC2 and Multilayer Switch Feature Card 2

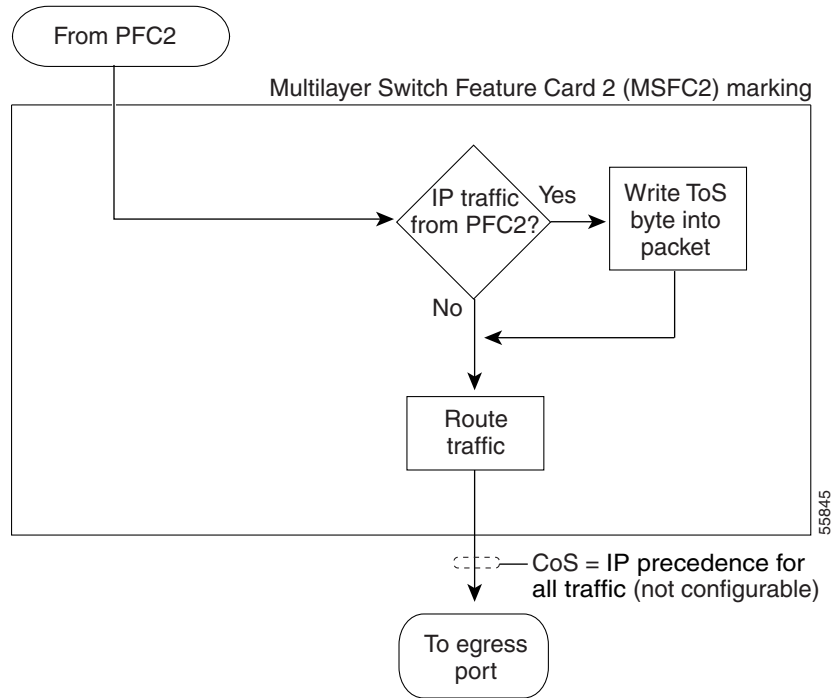


Figure 31-6 Marking with PFC1 and Multilayer Switch Feature Card 1 or 2

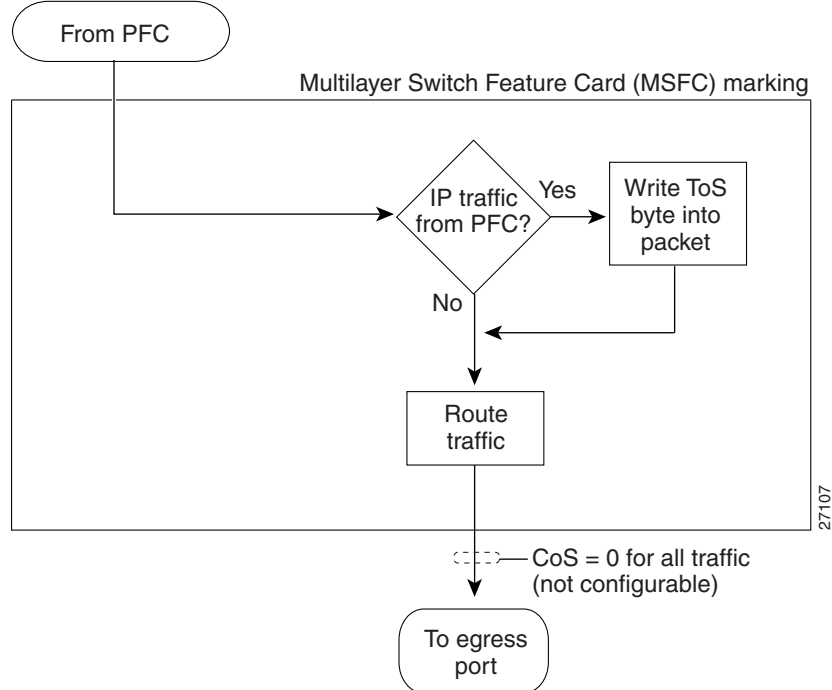


Figure 31-7 Egress WAN Port Marking

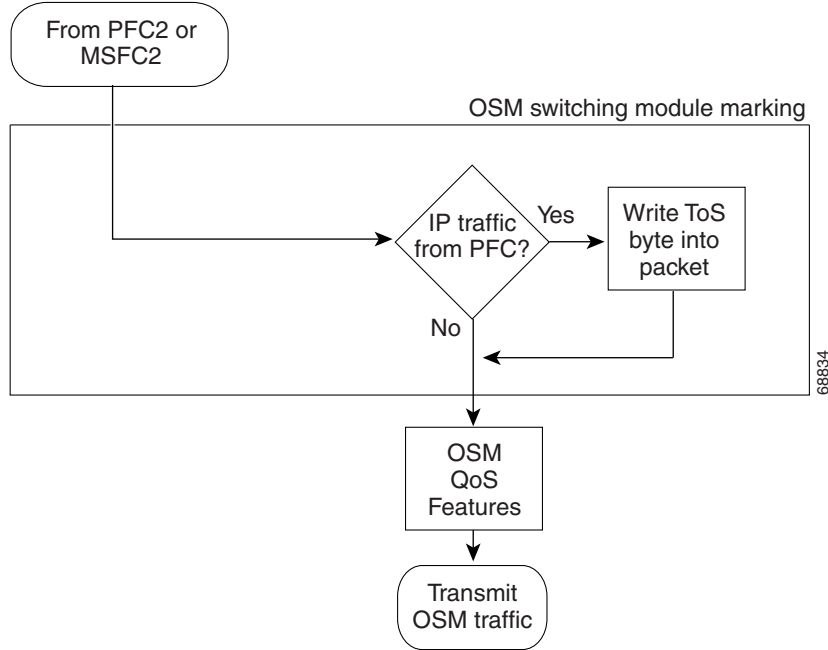
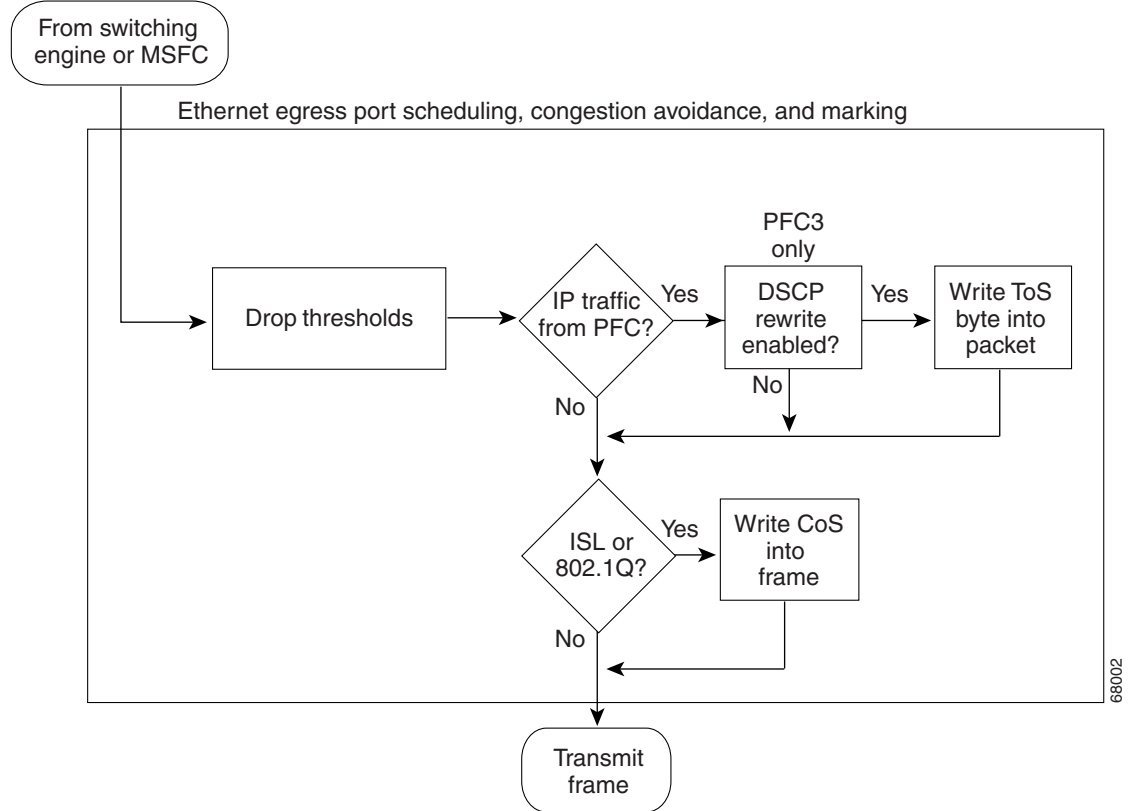


Figure 31-8 Egress LAN Port Scheduling, Congestion Avoidance, and Marking



PFC QoS Feature Summary

These sections summarize the PFC QoS features:

- [Ingress LAN Port Features, page 31-11](#)
- [Ingress OSM Port Features, page 31-11](#)
- [PFC QoS Features, page 31-11](#)
- [Egress LAN Port Features, page 31-12](#)
- [Egress OSM Port Features, page 31-12](#)
- [MSFC Features, page 31-12](#)

Ingress LAN Port Features

PFC QoS supports classification, marking, scheduling, and congestion avoidance using Layer 2 CoS values at ingress LAN ports. Classification, marking, scheduling, and congestion avoidance at ingress LAN ports do not use or set Layer 3 IP precedence or DSCP values. You can configure ingress LAN port trust states that can be used by the PFC to set Layer 3 IP precedence or DSCP values and the Layer 2 CoS value. See [Figure 31-3](#) and the “[Ingress LAN Port Features](#)” section on [page 31-12](#).

Ingress OSM Port Features

PFC QoS associates CoS zero with all traffic received through ingress OSM ports. You can configure ingress OSM port trust states that can be used by the PFC to set Layer 3 IP precedence or DSCP values and the Layer 2 CoS value. You can configure the trust state of each ingress OSM port as follows:

- Untrusted (default)
- Trust IP precedence
- Trust DSCP
- Trust CoS (CoS is always zero because the default port CoS is not configurable on OSM ports.)

PFC QoS Features

On the PFC, PFC QoS supports ingress classification, marking, and policing using policy maps. You can attach one policy map to an ingress port. Each policy map can contain multiple policy-map classes. You can configure a separate policy-map class for each type of traffic received through the ingress port. See the “[PFC Marking and Policing](#)” section on [page 31-16](#).



Note

- You can globally disable marking and policing with the **mls qos queueing-only** command (see the [Enabling Queueing-Only Mode, page 31-34](#)).
- You can disable marking and policing on a per-interface basis with the **no mls qos** interface command (see the “[Enabling or Disabling PFC Features on an Interface](#)” section on [page 31-51](#)).

Egress LAN Port Features

PFC QoS supports egress LAN port scheduling and congestion avoidance using Layer 2 CoS values. Egress LAN port marking sets Layer 2 CoS values and Layer 3 DSCP values. See the “[LAN Egress Port Features](#)” section on page 31-21.

Egress OSM Port Features

Ingress PFC QoS sets Layer 3 DSCP values that can be used by the OSM egress QoS features.

MSFC Features

PFC QoS marks IP traffic transmitted to the MSFC with rewritten Layer 3 DSCP values. With PFC2, CoS is equal to IP precedence in all traffic sent from the MSFC2 to egress ports; with PFC1, CoS is zero.



Note

Traffic that is Layer 3 switched does not go through the MFSC and retains the CoS value assigned by the PFC.

Ingress LAN Port Features

These sections describe ingress LAN port PFC QoS features:

- [Ingress LAN Port Trust States](#), page 31-12
- [Marking at Untrusted Ingress LAN Ports](#), page 31-13
- [Marking at Trusted Ingress LAN Ports](#), page 31-13
- [Ingress LAN Port Scheduling and Congestion Avoidance](#), page 31-13

Ingress LAN Port Trust States

The trust state of an ingress LAN port determines how the port marks, schedules, and classifies received Layer 2 frames, and whether or not congestion avoidance is implemented. You can configure the trust state of each ingress LAN port as follows:

- Untrusted (default)
- Trust IP precedence (not supported on **1q4t** LAN ports except Gigabit Ethernet)
- Trust DSCP (not supported on **1q4t** LAN ports except Gigabit Ethernet)
- Trust CoS (not supported on **1q4t** LAN ports except Gigabit Ethernet)

See the “[Configuring the Trust State of Ethernet LAN and OSM Ingress Ports](#)” section on page 31-53. PFC QoS implements ingress LAN port congestion avoidance only on LAN ports configured to trust CoS.



Note

Ingress LAN port marking, scheduling, and congestion avoidance use Layer 2 CoS values and does not use or set Layer 3 IP precedence or DSCP values.

Marking at Untrusted Ingress LAN Ports

PFC QoS marks all frames received through untrusted ingress LAN ports with the ingress port CoS value (the default is zero). PFC QoS does not implement ingress port congestion avoidance on untrusted ingress LAN ports.



Note

- To use the ingress port CoS value applied to untrusted traffic as the basis of egress DSCP, configure a trust-CoS policy map that matches the ingress traffic.
- The ingress port CoS value is configurable for each ingress LAN port (see the [“Configuring the Ingress LAN Port CoS Value”](#) section on page 31-54).

Marking at Trusted Ingress LAN Ports

When an ISL frame enters the Catalyst 6500 series switch through a trusted ingress LAN port, PFC QoS accepts the three least significant bits in the User field as a CoS value. When an 802.1Q frame enters the switch through a trusted ingress LAN port, PFC QoS accepts the User Priority bits as a CoS value. PFC QoS marks all traffic received in untagged frames with the ingress port CoS value.



Note

- PFC QoS uses the received CoS value in trusted tagged traffic as the basis of egress DSCP, unless there is a policy map that changes the trust state of the traffic.
- PFC QoS uses the ingress port CoS value applied to trusted untagged traffic as the basis of egress DSCP, unless there is a policy map that changes the trust state of the traffic.
- The ingress port CoS value is configurable for each ingress LAN port (see the [“Configuring the Ingress LAN Port CoS Value”](#) section on page 31-54).

Ingress LAN Port Scheduling and Congestion Avoidance

On ingress LAN ports configured to trust CoS, PFC QoS uses Layer 2 CoS-value based receive-queue drop thresholds to avoid congestion (see the [“Configuring the Trust State of Ethernet LAN and OSM Ingress Ports”](#) section on page 31-53).

Receive Queues

Enter the **show queueing interface** { **ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet** } *slot/port* | **include type** command to see the queue structure of a LAN port.

- **1q2t** indicates one standard queue with one configurable tail-drop threshold and one nonconfigurable tail-drop threshold.
- **1q4t** indicates one standard queue with four configurable tail-drop thresholds (usable only on Gigabit Ethernet ports).
- **1p1q4t** indicates one strict-priority queue and one standard queue with four configurable tail-drop thresholds.

- **1p1q0t** indicates one strict-priority queue and one standard queue with no configurable threshold (effectively a tail-drop threshold at 100 percent).
- **1p1q8t** indicates one strict-priority queue and one standard queue with eight thresholds, each configurable as either WRED-drop or tail-drop, and one non-configurable (100 percent) tail-drop threshold.

Strict-priority queues are queues that are serviced in preference to other queues. PFC QoS services traffic in a strict-priority queue before servicing the standard queue. When PFC QoS services the standard queue, after receiving a packet, it checks for traffic in the strict-priority queue. If PFC QoS detects traffic in the strict-priority queue, it suspends its service of the standard queue and completes service of all traffic in the strict-priority queue before returning to the standard queue.

Scheduling

PFC QoS schedules traffic through the receive queues based on Layer 2 CoS values. In the **1p1q4t**, **1p1q0t** and **1p1q8t** default configurations, PFC QoS assigns all traffic with CoS 5 to the strict-priority queue; PFC QoS assigns all other traffic to the standard queue. In the **1q4t** default configuration, PFC QoS assigns all traffic to the standard queue.

Congestion Avoidance

If an ingress LAN port is configured to trust CoS, PFC QoS implements Layer 2 CoS-value-based receive-queue drop thresholds to avoid congestion in received traffic.

1q2t ingress LAN ports have this default drop-threshold configuration:

- Frames with CoS 0, 1, 2, 3, or 4 go to tail-drop threshold 1, where the switch drops incoming frames when the standard receive-queue buffer is 80 percent full.
- Frames with CoS 5, 6, or 7 go to tail-drop threshold 2, where the switch drops incoming frames when the standard receive-queue buffer is 100 percent full.

1q4t ingress LAN ports have this default drop-threshold configuration:

- Using receive-queue tail-drop threshold 1, the switch drops incoming frames with CoS 0 or 1 when the receive-queue buffer is 50 percent or more full.
- Using receive-queue tail-drop threshold 2, the switch drops incoming frames with CoS 2 or 3 when the receive-queue buffer is 60 percent or more full.
- Using receive-queue tail-drop threshold 3, the switch drops incoming frames with CoS 4 or 5 when the receive-queue buffer is 80 percent or more full.
- Using receive-queue tail-drop threshold 4, the switch drops incoming frames with CoS 6 or 7 when the receive-queue buffer is 100 percent full.

1p1q4t ingress LAN ports have this default drop-threshold configuration:

- Frames with CoS 5 go to the strict-priority receive queue (queue 2), where the switch drops incoming frames only when the strict-priority receive-queue buffer is 100 percent full.
- Frames with CoS 0, 1, 2, 3, 4, 6, or 7 go to the standard receive queue.
 - Using standard receive-queue tail-drop threshold 1, the switch drops incoming frames with CoS 0 or 1 when the receive-queue buffer is 50 percent or more full.
 - Using standard receive-queue tail-drop threshold 2, the switch drops incoming frames with CoS 2 or 3 when the receive-queue buffer is 60 percent or more full.

- Using standard receive-queue tail-drop threshold 3, the switch drops incoming frames with CoS 4 when the receive-queue buffer is 80 percent or more full.
- Using standard receive-queue tail-drop threshold 4, the switch drops incoming frames with CoS 6 or 7 when the receive-queue buffer is 100 percent full.

1p1q0t ingress LAN ports have this default drop-threshold configuration:

- Frames with CoS 5 go to the strict-priority receive queue (queue 2), where the switch drops incoming frames only when the strict-priority receive-queue buffer is 100 percent full.
- Frames with CoS 0, 1, 2, 3, 4, 6, or 7 go to the standard receive queue. The switch drops incoming frames when the receive-queue buffer is 100 percent full.

1p1q8t ports have this default drop-threshold configuration:

- Frames with CoS 5 go to the strict-priority receive queue (queue 2), where the switch drops incoming frames only when the strict-priority receive-queue buffer is 100 percent full.
- Frames with CoS 0, 1, 2, 3, 4, 6, or 7 go to the standard receive queue, which uses WRED-drop thresholds:
 - Using standard receive-queue WRED-drop threshold 1 for incoming frames with CoS 0, the switch starts to drop frames when the receive-queue buffer is 40 percent full and drops all frames with CoS 0 when the receive-queue buffer is 70 percent or more full.
 - Using standard receive-queue WRED-drop threshold 2 for incoming frames with CoS 1, the switch starts to drop frames when the receive-queue buffer is 40 percent full and drops all frames with CoS 1 when the receive-queue buffer is 70 percent or more full.
 - Using standard receive-queue WRED-drop threshold 3 for incoming frames with CoS 2, the switch starts to drop frames when the receive-queue buffer is 50 percent full and drops all frames with CoS 2 when the receive-queue buffer is 80 percent or more full.
 - Using standard receive-queue WRED-drop threshold 4 for incoming frames with CoS 3, the switch starts to drop frames when the receive-queue buffer is 50 percent full and drops all frames with CoS 3 when the receive-queue buffer is 80 percent or more full.
 - Using standard receive-queue WRED-drop threshold 5 for incoming frames with CoS 4, the switch starts to drop frames when the receive-queue buffer is 60 percent full and drops all frames with CoS 4 when the receive-queue buffer is 90 percent or more full.
 - Using standard receive-queue WRED-drop threshold 6 for incoming frames with CoS 6, the switch starts to drop frames when the receive-queue buffer is 60 percent full and drops all frames with CoS 6 when the receive-queue buffer is 90 percent or more full.
 - Using standard receive-queue WRED-drop threshold 7 for incoming frames with CoS 7, the switch starts to drop frames when the receive-queue buffer is 70 percent full and drops all frames with CoS 7 when the receive-queue buffer is 100 percent or more full.



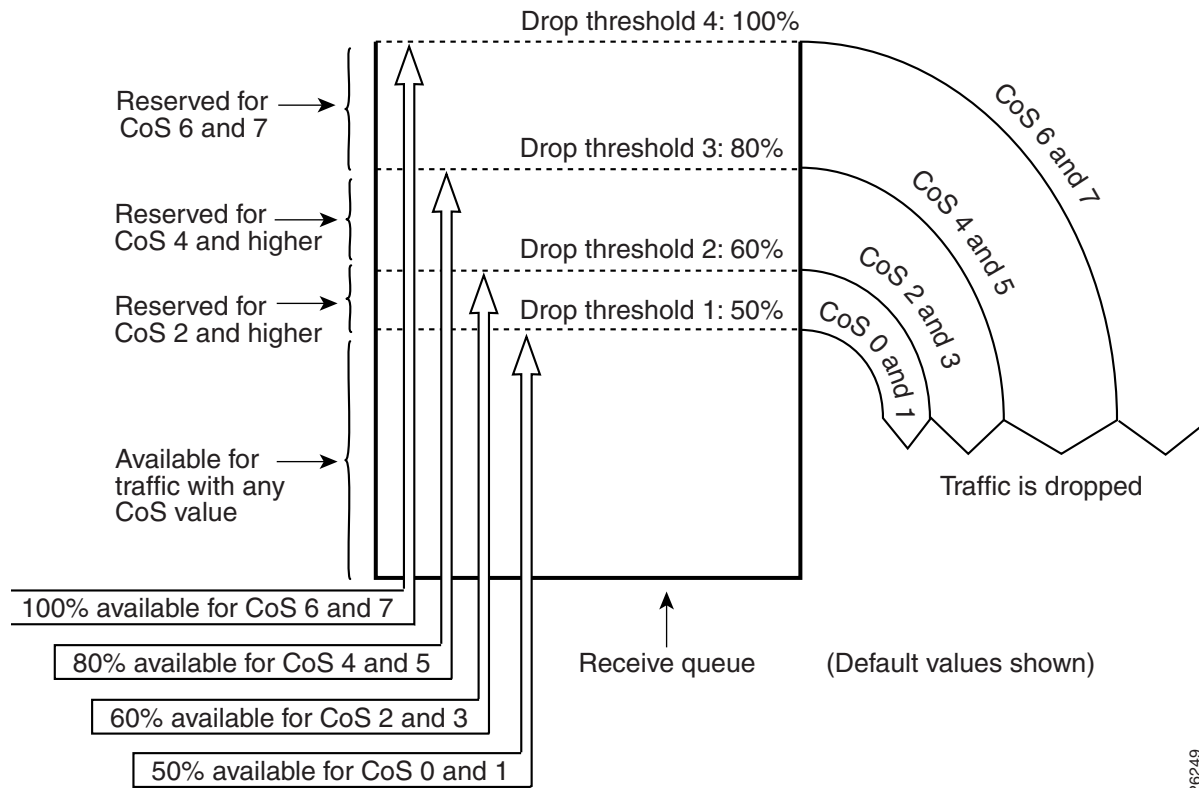
Note You can configure the standard receive queue to use both a tail-drop and a WRED-drop threshold by mapping a CoS value to the queue or to the queue and a threshold. The switch uses the tail-drop threshold for traffic carrying CoS values mapped only to the queue. The switch uses WRED-drop thresholds for traffic carrying CoS values mapped to the queue and a threshold. See the [“Configuring Standard-Queue Drop Threshold Percentages”](#) section on page 31-54.

**Note**

The explanations in this section use default values. You can configure many of the parameters (see the “[Configuring PFC QoS](#)” section on page 31-33). All LAN ports of the same type use the same drop-threshold configuration.

Figure 31-9 illustrates the drop thresholds for a **1q4t** ingress LAN port. Drop thresholds in other configurations function similarly.

Figure 31-9 Receive Queue Drop Thresholds



26249

PFC Marking and Policing

**Note**

- To mark untrusted traffic without policing in Release 12.1(12c)E1 and later releases, use the **set ip dscp** or **set ip precedence** policy map class commands (see the “[Configuring Policy Map Class Actions](#)” section on page 31-44).
- To mark untrusted traffic without policing in earlier releases, create a policer that only marks and does not police.

These sections describe PFC marking and policing:

- [Internal DSCP Values, page 31-17](#)
- [Policy Maps, page 31-18](#)

- [Policers, page 31-19](#)
- [Attaching Policy Maps, page 31-21](#)
- [Egress CoS and ToS Values, page 31-21](#)



Note Filtering for PFC QoS can use Layer 2, 3, and 4 values. Marking uses Layer 2 CoS values and Layer 3 IP precedence or DSCP values.

Internal DSCP Values

These sections describe the internal DSCP values:

- [Internal DSCP Sources, page 31-17](#)
- [Egress DSCP and CoS Sources, page 31-17](#)

Internal DSCP Sources

During processing, PFC QoS represents the priority of all traffic (including non-IP traffic) with an internal DSCP value. PFC QoS derives the internal DSCP value from the following:

- For trust-cos traffic, from received or ingress port Layer 2 CoS values



Note Traffic from an untrusted ingress LAN port has the ingress port CoS value and if traffic from an untrusted ingress Ethernet port matches a trust-cos policer, PFC QoS derives the internal DSCP value from the ingress port CoS value.

- For trust-ipprec traffic, from received IP precedence values
- For trust-dscp traffic, from received DSCP values
- For untrusted traffic, from ingress port CoS or configured DSCP values

The trust state of traffic is the trust state of the ingress LAN port unless set otherwise by a matching ACE.



Note A **trust-cos** policer cannot restore received CoS in traffic from untrusted ingress LAN ports. Traffic from untrusted ingress LAN ports always has the ingress port CoS value.

PFC QoS uses configurable mapping tables to derive the internal 6-bit DSCP value from CoS or IP precedence, which are 3-bit values (see the [“Mapping Received CoS Values to Internal DSCP Values”](#) section on page 31-66 or the [“Mapping Received IP Precedence Values to Internal DSCP Values”](#) section on page 31-67).

Egress DSCP and CoS Sources

For egress IP traffic, PFC QoS creates a ToS byte from the internal DSCP value and sends it to the egress port to be written into IP packets. For **trust-dscp** and **untrusted** IP traffic, the ToS byte includes the original 2 least-significant bits from the received ToS byte.



Note The internal DSCP value can mimic an IP precedence value (see [Table 31-1 on page 31-5](#)).

For all egress traffic, PFC QoS uses a configurable mapping table to derive a CoS value from the internal DSCP value associated with traffic (see the [“Mapping Internal DSCP Values to Egress CoS Values” section on page 31-67](#)). PFC QoS sends the CoS value to the egress LAN ports for use in scheduling and to be written into ISL and 802.1Q frames.

Policy Maps



Note

- You can globally disable marking and policing with the **mls qos queueing-only** command (see the [Enabling Queueing-Only Mode, page 31-34](#)).
- You can disable marking and policing on a per-interface basis with the **no mls qos** interface command (see the [“Enabling or Disabling PFC Features on an Interface” section on page 31-51](#)).

The PFC supports filtering, marking, and policing using policy maps (see the [“Configuring a Policy Map” section on page 31-42](#)). Each policy map can contain multiple policy-map classes. You can configure a separate policy-map class for each type of received traffic.

Policy-map classes specify filtering with the following:

- Cisco IOS access control lists (optional for IP, required for IPX and MAC-Layer filtering)
- Class-map **match** commands for Layer 3 IP precedence and DSCP values

Policy-map classes specify actions with the following:

- (Optional) Policy-map class **trust** commands. If specified, PFC QoS applies the policy-map class trust state to matched traffic. Policy-map class trust states supersede ingress LAN port trust states.



Note

If traffic matches a policy-map class that does not contain a **trust** command, the trust state remains as set on the ingress LAN port.

- (Optional) Aggregate and microflow policers, which can use bandwidth limits to either mark or drop both conforming and nonconforming traffic. See the [“PFC Marking and Policing” section on page 31-16](#).

The PFC uses the trust state (set by the ingress LAN port configuration or by a **trust** policy-map class command) to select the Layer 2 and Layer 3 PFC QoS labels that the egress port writes into the packets and frames before it is transmitted:

- Trust IP precedence—Sets the internal DSCP value to a mapped value based on received IP precedence (see the [“Mapping Received IP Precedence Values to Internal DSCP Values” section on page 31-67](#)).
- Trust DSCP—Sets the internal DSCP value to the received DSCP value.

Trust CoS—Sets the internal DSCP value to a mapped value based on received or port CoS. With trust CoS, note the following:

- Received CoS is overwritten with port CoS in traffic received through ports not configured to trust CoS.
- Received CoS is preserved only in traffic received through ports configured to trust CoS.
- Port CoS is applied to all traffic received in untagged frames, regardless of the port trust state.
- For information about mapping, see the [“Mapping Received CoS Values to Internal DSCP Values” section on page 31-66](#).

- Untrusted—Sets the internal DSCP value to a configured DSCP value.



Note With the default values, PFC QoS applies DSCP zero to traffic from ingress LAN ports configured as untrusted.

Policers



Note

Policing with the **conform-action transmit** keywords supersedes the ingress LAN port trust state of matched traffic with trust DSCP or with the trust state defined by a **trust** policy-map class command (see the “[Configuring the Policy Map Class Trust State](#)” section on page 31-45).

You can create policers that do the following:

- Mark traffic
- Limit bandwidth utilization and mark traffic

For more information, see the “[Creating Named Aggregate Policers](#)” section on page 31-35 and the “[Configuring Policy Map Class Actions](#)” section on page 31-44.

Policing rates are based on the Layer 3 packet size. You specify the bandwidth utilization limit as a committed information rate (CIR). With a PFC2, you can also specify a higher peak information rate (PIR). Packets that exceed a rate are “out of profile” or “nonconforming.”

In each policer, you specify if out-of-profile packets are to be dropped or to have a new DSCP value applied to them (applying a new DSCP value is called “markdown”). Because out-of-profile packets do not retain their original priority, they are not counted as part of the bandwidth consumed by in-profile packets.

With a PFC2, if you configure a PIR, the PIR out-of-profile action cannot be less severe than the CIR out-of-profile action. For example, if the CIR out-of-profile action is to mark down the traffic, then the PIR out-of-profile action cannot be to transmit the traffic.

For all policers, PFC QoS uses a configurable global table that maps the internal DSCP value to a marked-down DSCP value (see the “[Configuring DSCP Markdown Values](#)” section on page 31-68). When markdown occurs, PFC QoS gets the marked-down DSCP value from the table. You cannot specify marked-down DSCP values in individual policers.



Note

By default, the markdown table is configured so that no markdown occurs: the marked-down DSCP values are equal to the original DSCP values. To enable markdown, configure the table appropriately for your network.

You can create two kinds of policers: *aggregate* and *microflow*:

- PFC QoS applies the bandwidth limits specified in an aggregate policer cumulatively to all flows in matched traffic. You can create up to 1023 aggregate policers. You can create two types of aggregate policer: named and per port. Both types can be attached to more than one port:
 - You define per-interface aggregate policers in a policy map class with the **police** command. If you attach a per-interface aggregate policer to multiple ingress ports, it polices the matched traffic on each ingress port separately.
 - You create named aggregate policers with the **mls qos aggregate-policer** command. If you attach a named aggregate policer to multiple ingress ports, it polices the matched traffic from all the ingress ports to which it is attached.

**Note**

Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC2, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC2 and any non-DFC-equipped switching modules supported by the PFC2.

- PFC QoS applies the bandwidth limit specified in a microflow policer separately to each flow in matched traffic as follows:
 - You can create microflow policers with up to 63 different rate and burst parameter combinations.
 - You create microflow policers in a policy map class with the **police flow** command.
 - For IPX microflow policing, PFC QoS considers IPX traffic with the same source network, destination network, and destination node to be part of the same flow, including traffic with different source nodes or source sockets.
 - For MAC-Layer microflow policing, PFC QoS considers MAC-Layer traffic with the same protocol and the same source and destination MAC-Layer addresses to be part of the same flow, including traffic with different ethertypes.
 - By default, microflow policers only affect traffic routed by the MSFC. To enable microflow policing of other traffic, including traffic in bridge groups, enter the **mls qos bridged** command (see the “[Enabling Microflow Policing of Bridged Traffic](#)” section on page 31-50).

You can include both an aggregate policer and a microflow policer in each policy map class to police a flow based on both its own bandwidth utilization and on its bandwidth utilization combined with that of other flows.

**Note**

If traffic is both aggregate and microflow policed, then the aggregate and microflow policers must both be in the same policy-map class and each must use the same **conform-action** keyword option: **drop**, **set-dscp-transmit**, **set-prec-transmit**, or **transmit**.

For example, you could create a microflow policer with a bandwidth limit suitable for individuals in a group and you could create a named aggregate policer with bandwidth limits suitable for the group as a whole. You could include both policers in policy map classes that match the group’s traffic. The combination would affect individual flows separately and the group aggregately.

For policy map classes that include both an aggregate and a microflow policer, PFC QoS responds to an out-of-profile status from either policer and, as specified by the policer, applies a new DSCP value or drops the packet. If both policers return an out-of-profile status, then if either policer specifies that the packet is to be dropped, it is dropped; otherwise PFC QoS applies a marked-down DSCP value.

**Note**

To avoid inconsistent results, ensure that all traffic policed by the same aggregate policer has the same trust state.

Attaching Policy Maps

You can configure each ingress LAN port for either physical port-based PFC QoS (default) or VLAN-based PFC QoS (see the “[Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports](#)” section on page 31-52) and attach a policy map to the selected port (see the “[Attaching a Policy Map to an Interface](#)” section on page 31-49).

On ports configured for port-based PFC QoS, you can attach a policy map to the ingress LAN port as follows:

- On a nontrunk ingress LAN port configured for port-based PFC QoS, all traffic received through the port is classified, marked, and policed according to the policy map attached to the port.
- On a trunking ingress LAN port configured for port-based PFC QoS, traffic in *all VLANs* received through the port is classified, marked, and policed according to the policy map attached to the port.

On a nontrunk ingress LAN port configured for VLAN-based PFC QoS, traffic received through the port is classified, marked, and policed according to the policy map attached to the *port's* VLAN.

On a trunking ingress LAN port configured for VLAN-based PFC QoS, traffic received through the port is classified, marked, and policed according to the policy map attached to the *traffic's* VLAN.

You can attach policy maps to OSM ports.

Egress CoS and ToS Values

PFC QoS associates CoS and ToS values with traffic as specified by the trust state and policers in the policy map (see the “[Internal DSCP Values](#)” section on page 31-17). The associated CoS and ToS are used at the egress port (see the “[LAN Egress Port Features](#)” section on page 31-21).

LAN Egress Port Features

These sections describe how PFC QoS schedules traffic through the transmit queues based on CoS values and uses CoS-value-based transmit-queue drop thresholds to avoid congestion in traffic transmitted from egress LAN ports:

- [Transmit Queues](#), page 31-21
- [Scheduling and Congestion Avoidance](#), page 31-22
- [Marking](#), page 31-24

**Note**

Egress LAN port scheduling and congestion avoidance uses Layer 2 CoS values. Egress LAN port marking writes Layer 2 CoS values into trunk traffic and the Layer 3 ToS byte into all IP traffic.

Transmit Queues

Enter the **show queueing interface** { **ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet** } *slot/port* | **include type** command to see the queue structure of an egress LAN port.

The command displays one of the following:

- **2q2t** indicates two standard queues, each with two configurable tail-drop thresholds
- **1p2q2t** indicates one strict-priority queue and two standard queues, each with two configurable WRED-drop thresholds.

- **1p3q1t** indicates one strict-priority queue and three standard queues, each with one threshold configurable as either WRED-drop or tail-drop, and one nonconfigurable tail-drop threshold.
- **1p2q1t** indicates one strict-priority queue and two standard queues, each with one configurable WRED-drop threshold and one nonconfigurable tail-drop threshold.

All port types have a low-priority and a high-priority standard transmit queue. **1p3q1t** ports have a medium-priority standard transmit queue. **1p2q2t**, **1p3q1t** and **1p2q1t** ports have a strict-priority transmit queue in addition to the standard queues.

On **2q2t** ports, the default PFC QoS configuration allocates a minimum of 80 percent of the total transmit queue size to the low-priority standard queue and a minimum of 20 percent to the high-priority standard queue.

On **1p2q2t**, **1p3q1t**, and **1p2q1t** ports, the switch services traffic in the strict-priority queue before servicing the standard queues. When the switch is servicing a standard queue, after transmitting a packet, it checks for traffic in the strict-priority queue. If the switch detects traffic in the strict-priority queue, it suspends its service of the standard queue and completes service of all traffic in the strict-priority queue before returning to the standard queue.

On **1p2q2t** ports, the default PFC QoS configuration allocates a minimum of 70 percent of the total transmit queue size to the low-priority standard queue, a minimum of 15 percent to the high-priority standard queue, and a minimum of 15 percent to the strict-priority queue.

On **1p3q1t** ports, the transmit queue size is not configurable and is allocated equally among all queues.

On **1p2q1t** ports, the default PFC QoS configuration allocates a minimum of 50 percent of the total transmit queue size to the low-priority standard queue, a minimum of 30 percent to the high-priority standard queue, and a minimum of 20 percent to the strict-priority queue.

**Note**

Transmit-queue size is limited to the configured value (see the [“Setting the Receive-Queue Size Ratio on a 1p1q0t or 1p1q8t Ingress LAN Ports”](#) section on page 31-64), but any queue can use all available bandwidth (bandwidth is only available when there is no traffic in the other queues).

Scheduling and Congestion Avoidance

These sections describe scheduling and congestion avoidance:

- [2q2t Ports, page 31-23](#)
- [1p2q2t Ports, page 31-23](#)
- [1p3q1t Ports, page 31-23](#)
- [1p2q1t Ports, page 31-24](#)

**Note**

The explanations in these sections use default values. You can configure many of the parameters (for more information, see the [“Configuring PFC QoS”](#) section on page 31-33). All ports of the same type use the same drop-threshold configuration.

2q2t Ports

For **2q2t** ports, each transmit queue has two tail-drop thresholds that function as follows:

- Frames with CoS 0, 1, 2, or 3 go to the low-priority transmit queue (queue 1):
 - Using transmit queue 1, tail-drop threshold 1, the switch drops frames with CoS 0 or 1 when the low-priority transmit-queue buffer is 80 percent full.
 - Using transmit queue 1, tail-drop threshold 2, the switch drops frames with CoS 2 or 3 when the low-priority transmit-queue buffer is 100 percent full.
- Frames with CoS 4, 5, 6, or 7 go to the high-priority transmit queue (queue 2):
 - Using transmit queue 2, tail-drop threshold 1, the switch drops frames with CoS 4 or 5 when the high-priority transmit-queue buffer is 80 percent full.
 - Using transmit queue 2, tail-drop threshold 2, the switch drops frames with CoS 6 or 7 when the high-priority transmit-queue buffer is 100 percent full.

1p2q2t Ports

1p2q2t ports have a strict-priority queue and two standard transmit queues. The two standard transmit queues each have two WRED-drop thresholds.

- Frames with CoS 5 go to the strict-priority transmit queue (queue 3), where the switch drops frames only when the buffer is 100 percent full.
- Frames with CoS 0, 1, 2, or 3 go to the low-priority standard transmit queue (queue 1):
 - Using standard transmit queue 1, WRED-drop threshold 1, the switch drops frames with CoS 0 or 1 when the low-priority transmit-queue buffer is 80 percent full.
 - Using standard transmit queue 1, WRED-drop threshold 2, the switch drops frames with CoS 2 or 3 when the low-priority transmit-queue buffer is 100 percent full.
- Frames with CoS 4, 6, or 7 go to the high-priority standard transmit queue (queue 2):
 - Using standard transmit queue 2, WRED-drop threshold 1, the switch drops frames with CoS 4 when the high-priority transmit-queue buffer is 80 percent full.
 - Using standard transmit queue 2, WRED-drop threshold 2, the switch drops frames with CoS 6 or 7 when the high-priority transmit-queue buffer is 100 percent full.

1p3q1t Ports

1p3q1t ports have a strict-priority queue and three standard transmit queues. The standard transmit queues each have one WRED-drop threshold and one nonconfigurable tail-drop threshold.

- Frames with CoS 5 go to the strict-priority transmit queue (queue 4), where the switch drops frames only when the buffer is 100 percent full.
- Frames with CoS 0 and 1 go to the low-priority standard transmit queue (queue 1).
- Frames with CoS 2, 3, or 4 go to the medium-priority standard transmit queue (queue 2).
- Frames with CoS 6 or 7 go to the high-priority standard transmit queue (queue 3).



Note You can configure each standard transmit queue to use both a non-configurable 100 percent tail-drop threshold and a configurable WRED-drop threshold (see the [“Configuring Standard-Queue Drop Threshold Percentages”](#) section on page 31-54).

1p2q1t Ports

1p2q1t ports have a strict-priority queue and two standard transmit queues. The standard transmit queues each have one WRED-drop threshold and one nonconfigurable tail-drop threshold.

- Frames with CoS 5 go to the strict-priority transmit queue (queue 3), where the switch drops frames only when the buffer is 100 percent full.
- The standard transmit queues have WRED-drop thresholds:
 - Frames with CoS 0, 1, 2, or 3 go to the low-priority transmit queue (queue 1), where the switch starts to drop frames when the low-priority transmit-queue buffer is 70 percent full and drops all frames with CoS 0, 1, 2, or 3 when the buffer is 100 percent full.
 - Frames with CoS 4, 6, or 7 go to the high-priority transmit queue (queue 2), where the switch starts to drop frames when the high-priority transmit-queue buffer is 70 percent full and drops all frames with CoS 4, 6, or 7 when the buffer is 100 percent full.


Note

You can configure each standard transmit queue to use both the tail-drop and the WRED-drop threshold. See the [“Configuring Standard-Queue Drop Threshold Percentages” section on page 31-54](#).

Marking

When traffic is transmitted from the switch, PFC QoS writes the ToS byte into IP packets. On LAN ports, PFC QoS also writes the CoS value that was used for scheduling and congestion avoidance into ISL and 802.1Q frames (see the [“Egress CoS and ToS Values” section on page 31-21](#)).

PFC QoS Statistics Data Export


Note

Release 12.1(11b)E or later supports PFC QoS statistics data export.

The PFC QoS statistics data export feature generates per-LAN-port and per-aggregate policer utilization information and forwards this information in UDP packets to traffic monitoring, planning, or accounting applications. You can enable PFC QoS statistics data export on a per-LAN-port or on a per-aggregate policer basis. The statistics data generated per port consists of counts of the input and output packets and bytes. The aggregate policer statistics consist of counts of allowed packets and counts of packets exceeding the policed rate.

The PFC QoS statistics data collection occurs periodically at a fixed interval, but you can configure the interval at which the data is exported. PFC QoS statistics collection is enabled by default, and the data export feature is disabled by default for all ports and all aggregate policers configured on the Catalyst 6500 series switch.


Note

The PFC QoS statistics data export feature is completely separate from NetFlow Data Export and does not interact with it.

PFC QoS Default Configuration

Table 31-2 shows the PFC QoS default configuration.

Table 31-2 PFC QoS Default Configuration

Feature	Default Value
PFC QoS global enable state	Disabled Note With PFC QoS enabled and all other PFC QoS parameters at default values, PFC QoS sets Layer 3 DSCP to zero and Layer 2 CoS to zero in all traffic transmitted from the switch.
PFC QoS port enable state	Enabled when PFC QoS is globally enabled
Port CoS value	0
Microflow policing	Enabled
IntraVLAN microflow policing	Disabled
Port-based or VLAN-based PFC QoS	Port-based
CoS to DSCP map (DSCP set from CoS values)	CoS 0 = DSCP 0 CoS 1 = DSCP 8 CoS 2 = DSCP 16 CoS 3 = DSCP 24 CoS 4 = DSCP 32 CoS 5 = DSCP 40 CoS 6 = DSCP 48 CoS 7 = DSCP 56
IP precedence to DSCP map (DSCP set from IP precedence values)	IP precedence 0 = DSCP 0 IP precedence 1 = DSCP 8 IP precedence 2 = DSCP 16 IP precedence 3 = DSCP 24 IP precedence 4 = DSCP 32 IP precedence 5 = DSCP 40 IP precedence 6 = DSCP 48 IP precedence 7 = DSCP 56
DSCP to CoS map (CoS set from DSCP values)	DSCP 0–7 = CoS 0 DSCP 8–15 = CoS 1 DSCP 16–23 = CoS 2 DSCP 24–31 = CoS 3 DSCP 32–39 = CoS 4 DSCP 40–47 = CoS 5 DSCP 48–55 = CoS 6 DSCP 56–63 = CoS 7
Marked-down DSCP from DSCP map	Marked-down DSCP value equals original DSCP value (no markdown)
Policers	None
Policy maps	None

Table 31-2 PFC QoS Default Configuration (continued)

Feature	Default Value
With PFC QoS enabled	
Ingress LAN port trust state	Untrusted
2q2t transmit-queue size percentages	<ul style="list-style-type: none"> • Low priority: 80% • High priority: 20%
1p2q2t transmit-queue size percentages	<ul style="list-style-type: none"> • Low priority: 70% • High priority: 15% • Strict priority 15%
1p2q1t transmit-queue size percentages	<ul style="list-style-type: none"> • Low priority: 70% • High priority: 15% • Strict priority 15%
1p2q1t standard transmit-queue low:high priority bandwidth allocation ratio	100:255
2q2t , 1p2q2t , and 1p2q1t standard transmit-queue low:high priority bandwidth allocation ratio	5:255
1p3q1t standard transmit-queue low:medium:high-priority bandwidth allocation ratio	100:150:255
1q4t/2q2t receive and transmit-queue CoS value and drop-threshold mapping	<p>Receive queue 1/drop threshold 1 (50%) and transmit queue 1/drop threshold 1 (80%)—CoS 0 and 1</p> <p>Receive queue 1/drop threshold 2 (60%) and transmit queue 1/drop threshold 2 (100%)—CoS 2 and 3</p> <p>Receive queue 1/drop threshold 3 (80%) and transmit queue 2/drop threshold 1 (80%)—CoS 4 and 5</p> <p>Receive queue 1/drop threshold 4 (100%) and transmit queue 2/drop threshold 2 (100%)—CoS 6 and 7</p>
1q2t port receive-queue CoS value/drop-threshold mapping and threshold percentages	<p>Receive queue 1:</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 0, 1, 2, 3, and 4 – Tail-drop threshold: 80% • Threshold 2: <ul style="list-style-type: none"> – CoS 5, 6, and 7 – Tail-drop threshold: 100% (not configurable)

Table 31-2 PFC QoS Default Configuration (continued)

Feature	Default Value
1p1q4t port receive-queue CoS value and drop-threshold mapping and threshold percentages	Standard receive queue: <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 0 and 1 – Tail-drop: 50% • Threshold 2: <ul style="list-style-type: none"> – CoS 2 and 3 – Tail-drop: 60% • Threshold 3: <ul style="list-style-type: none"> – CoS 4 – Tail-drop: 80% • Threshold 4: <ul style="list-style-type: none"> – CoS 6 and 7 – Tail-drop: 100% Strict-priority receive queue: <ul style="list-style-type: none"> • CoS 5 • Tail-drop: 100% (nonconfigurable)
1p1q0t port receive-queue CoS value and drop-threshold mapping and threshold percentages	Standard receive queue 1: <ul style="list-style-type: none"> • CoS 0, 1, 2, 3, 4, 6, and 7 • Tail-drop: 100% (nonconfigurable) Strict-priority receive queue: <ul style="list-style-type: none"> • CoS 5 • Tail-drop: 100% (nonconfigurable)

Table 31-2 PFC QoS Default Configuration (continued)

Feature	Default Value
1p1q8t receive-queue port CoS value and drop-threshold mapping	Standard receive queue: <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 0 – Tail-drop: 70% – WRED-drop: 40% low, 70% high • Threshold 2: <ul style="list-style-type: none"> – CoS 1 – Tail-drop: 70% – WRED-drop: 40% low, 70% high • Threshold 3: <ul style="list-style-type: none"> – CoS 2 – Tail-drop: 80% – WRED-drop: 50% low, 80% high • Threshold 4: <ul style="list-style-type: none"> – CoS 3 – Tail-drop: 80% – WRED-drop: 50% low, 80% high • Threshold 5: <ul style="list-style-type: none"> – CoS 4 – Tail-drop: 90% – WRED-drop: 60% low, 90% high • Threshold 6: <ul style="list-style-type: none"> – CoS 6 – Tail-drop: 90% – WRED-drop: 60% low, 90% high • Threshold 6: <ul style="list-style-type: none"> – CoS 7 – Tail-drop: 100% – WRED-drop (enabled): 70% low, 100% high Strict-priority receive queue: <ul style="list-style-type: none"> • CoS 5 • Tail-drop: 100% (nonconfigurable)

Table 31-2 PFC QoS Default Configuration (continued)

Feature	Default Value
<p>1p2q2t port transmit-queue CoS value and drop-threshold mapping and threshold percentages</p>	<p>Standard transmit queue 1 (low priority):</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 0 and 1 – WRED-drop: 40% low, 70% high • Threshold 2: <ul style="list-style-type: none"> – CoS 2 and 3 – WRED-drop: 70% low, 100% high <p>Standard transmit queue 2 (high priority):</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 4 – WRED-drop: 40% low, 70% high • Threshold 2: <ul style="list-style-type: none"> – CoS 6 and 7 – WRED-drop: 70% low, 100% high <p>Strict-priority receive queue:</p> <ul style="list-style-type: none"> • CoS 5 • Tail-drop: 100% (nonconfigurable)
<p>1p7q8t transmit-queue CoS value and drop-threshold mapping</p>	<ul style="list-style-type: none"> •

Table 31-2 PFC QoS Default Configuration (continued)

Feature	Default Value
1p3q1t transmit-queue CoS value and drop-threshold mapping	<p>Standard transmit queue 1 (low priority):</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 0 and 1 – Tail drop (disabled): 100% – WRED-drop (enabled): 70% low, 100% high <p>Standard transmit queue 2 (medium priority) tail-drop threshold:</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 2, 3, and 4 – Tail drop (disabled): 100% – WRED-drop (enabled): 70% low, 100% high <p>Standard transmit queue 3 (high priority) tail-drop threshold:</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 6 and 7 – Tail drop (disabled): 100% – WRED-drop (enabled): 70% low, 100% high <p>Strict-priority receive queue:</p> <ul style="list-style-type: none"> • CoS 5 • Tail-drop: 100% (nonconfigurable)
1p2q1t transmit-queue port CoS value and drop-threshold mapping	<p>Standard transmit queue 1 (low priority):</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS 0, 1, 2, and 3 – WRED-drop: 70% low, 100% high <p>Standard transmit queue 2 (high priority) WRED-drop threshold:</p> <ul style="list-style-type: none"> • Threshold 1: <ul style="list-style-type: none"> – CoS CoS 4, 6, and 7 – WRED-drop: 70% low, 100% high <p>Strict-priority receive queue:</p> <ul style="list-style-type: none"> • CoS 5 • Tail-drop: 100% (nonconfigurable)
With PFC QoS disabled	
Ingress LAN port trust state	trust-dscp
Receive-queue drop-threshold percentages	All thresholds set to 100%
Transmit-queue drop-threshold percentages	All thresholds set to 100%
Transmit-queue bandwidth allocation ratio	255:1

Table 31-2 PFC QoS Default Configuration (continued)

Feature	Default Value
Transmit-queue size ratio	Low priority: 100% (other queues not used)
CoS value and drop threshold mapping	All CoS values mapped to the low-priority queue.

PFC QoS Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring PFC QoS:

Guidelines:

- With an MSFC2, Release 12.1(13)E and later releases support the **match protocol** class map command, which configures NBAR and sends all traffic on the Layer 3 interface, both ingress and egress, to be processed in software on the MSFC2. To configure NBAR, refer to this publication: <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm>
Earlier releases provide PFC QoS and Layer 3 switching in hardware, which prevents support of the **match protocol** class map command except for traffic being processed in software on the MSFC.
- With Release 12.1(12c)E1 and later releases, PFC QoS supports the **set ip dscp** and **set ip precedence** policy map class commands (see the “[Configuring Policy Map Class Actions](#)” section on page 31-44). With Release 12.1(12c)E1 and later releases, PFC QoS does not support the **set mpls experimental** or **set qos-group** policy map class commands. With earlier releases, PFC QoS does not support any **set** policy map class commands.
- With Release 12.1(11b)E1 and later releases, OSM QoS supports the **set mpls experimental** policy map class command. Refer to the following publication for information about OSM QoS: http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/osm_inst/index.htm
- PFC QoS has the following hardware granularity for CIR and PIR rate values:

CIR and PIR Rate Value Range	Granularity
32768 to 2097152 (2 Mbps)	32768 (32 Kb)
2097153 to 4194304 (4 Mbps)	65536 (64 Kb)
4194305 to 8388608 (8 Mbps)	131072 (128 Kb)
8388609 to 16777216 (16 Mbps)	262144 (256 Kb)
16777217 to 33554432 (32 Mbps)	524288 (512 Kb)
33554433 to 67108864 (64 Mbps)	1048576 (1 Mb)
67108865 to 134217728 (128 Mbps)	2097152 (2 Mb)
34217729 to 268435456 (256 Mbps)	4194304 (4 Mb)
268435457 to 536870912 (512 Mbps)	8388608 (8 Mb)
536870913 to 1073741824 (1 Gps)	16777216 (16 Mb)
1073741825 to 2147483648 (2 Gps)	33554432 (32 Mb)
2147483649 to 4294967296 (4 Gps)	67108864 (64 Mb)

Within each range, PFC QoS programs the PFC hardware with rate values that are multiples of the granularity values.

- PFC QoS has the following hardware granularity for CIR and PIR token bucket (burst) sizes:

CIR and PIR Token Bucket Size Range		Granularity	
1 to 32768	(32 KB)	1024	(1 KB)
32769 to 65536	(64 KB)	2048	(2 KB)
65537 to 131072	(128 KB)	4096	(4 KB)
131073 to 262144	(256 KB)	8196	(8 KB)
262145 to 524288	(512 KB)	16392	(16 KB)
524289 to 1048576	(1 MB)	32768	(32 KB)
1048577 to 2097152	(2 MB)	65536	(64 KB)
2097153 to 4194304	(4 MB)	131072	(128 KB)
4194305 to 8388608	(8 MB)	262144	(256 KB)
8388609 to 16777216	(16 MB)	524288	(512 KB)
16777217 to 33554432	(32 MB)	1048576	(1 MB)

Within each range, PFC QoS programs the PFC hardware with token bucket sizes that are multiples of the granularity values.

- For these commands, PFC QoS applies identical configuration to all LAN ports controlled by the same application-specific integrated circuit (ASIC) or group of ASICs (see the port group information for each module in the Release Notes):
 - rcv-queue queue-limit
 - wrp-queue queue-limit
 - wrp-queue bandwidth (except Gigabit Ethernet LAN ports)
 - priority-queue cos-map
 - rcv-queue cos-map
 - wrp-queue cos-map
 - wrp-queue threshold
 - rcv-queue threshold
 - wrp-queue random-detect
 - wrp-queue random-detect min-threshold
 - wrp-queue random-detect max-threshold

Restrictions

- PFC QoS filters only by access lists, dscp values, or IP precedence values.
- PFC QoS does not support the following commands and configurations:
 - **match cos**, **match any**, **match classmap**, **match destination-address**, **match input-interface**, **match mpls**, **match qos-group**, or **match source-address** class map commands
 - class maps that contain *multiple match* commands

- **output** service-policy keyword
- **class** *class_name* **destination-address**, **class** *class_name* **input-interface**, **class** *class_name* **protocol**, **class** *class_name* **qos-group**, or **class** *class_name* **source-address** policy map commands
- **bandwidth**, **priority**, **queue-limit**, or **random-detect** policy map class commands

Configuring PFC QoS

These sections describe how to configure PFC QoS on the Catalyst 6500 series switches:

- [Enabling PFC QoS Globally](#), page 31-33
- [Enabling Queueing-Only Mode](#), page 31-34
- [Creating Named Aggregate Policers](#), page 31-35
- [Configuring a PFC QoS Policy](#), page 31-37
- [Enabling or Disabling Microflow Policing](#), page 31-50
- [Enabling Microflow Policing of Bridged Traffic](#), page 31-50
- [Enabling or Disabling PFC Features on an Interface](#), page 31-51
- [Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports](#), page 31-52
- [Configuring the Trust State of Ethernet LAN and OSM Ingress Ports](#), page 31-53
- [Configuring the Ingress LAN Port CoS Value](#), page 31-54
- [Configuring Standard-Queue Drop Threshold Percentages](#), page 31-54
- [Mapping CoS Values to Drop Thresholds](#), page 31-59
- [Allocating Bandwidth Between LAN-Port Transmit Queues](#), page 31-64
- [Setting the Receive-Queue Size Ratio on a 1p1q0t or 1p1q8t Ingress LAN Ports](#), page 31-64
- [Setting the LAN-Port Transmit-Queue Size Ratio](#), page 31-65
- [Configuring DSCP Value Maps](#), page 31-66
- [Configuring PFC QoS Statistics Data Export](#), page 31-70



Note

- PFC QoS processes both unicast and multicast traffic.
- With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling PFC QoS Globally

To enable PFC QoS globally, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos	Enables PFC QoS globally on the switch.
	Router(config)# no mls qos	Disables PFC QoS globally on the switch.

	Command	Purpose
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos	Verifies the configuration.

This example shows how to enable PFC QoS globally:

```
Router# configure terminal
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
QoS is enabled globally
Microflow QoS is enabled globally

QoS global counters:
Total packets: 544393
IP shortcut packets: 1410
Packets dropped by policing: 0
IP packets with TOS changed by policing: 467
IP packets with COS changed by policing: 59998
Non-IP packets with COS changed by policing: 0
```

Enabling Queueing-Only Mode

To enable queueing-only mode on the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos queueing-only	Enables queueing-only mode on the switch.
	Router(config)# no mls qos queueing-only	Disables PFC QoS globally on the switch. Note You cannot disable queueing-only mode separately.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos	Verifies the configuration.

When you enable queueing-only mode, the switch does the following:

- Disables marking and policing globally
- Configures all ports to trust Layer 2 CoS



Note The switch applies the port CoS value to untagged ingress traffic and to traffic that is received through ports that cannot be configured to trust CoS.

This example shows how to enable queueing-only mode:

```
Router# configure terminal
Router(config)# mls qos queueing-only
Router(config)# end
Router#
```


Creating Named Aggregate Policers

To create a named aggregate policer (see the “Policers” section on page 31-19), perform this task:

Command	Purpose
<pre>Router(config)# mls qos aggregate-policer policer_name bits_per_second normal_burst_bytes [maximum_burst_bytes] [pir¹ peak_rate_bps] [[conform-action {drop set-dscp-transmit² dscp_value set-prec-transmit² ip_precedence_value transmit}] exceed-action {drop policed-dscp transmit}] violate-action¹ {drop policed-dscp transmit}]</pre>	Creates a named aggregate policer.
<pre>Router(config)# no mls qos aggregate-policer policer_name</pre>	Deletes a named aggregate policer.

1. Supported only with PFC2.

2. With PFC2, the `set-dscp-transmit` and `set-prec-transmit` keywords are only supported for IP traffic.



Note

With PFC2, aggregate policers can be applied to ingress interfaces on multiple modules, but aggregate policing works independently on each DFC-equipped switching module and independently on the PFC2, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC2 and any non-DFC-equipped switching modules supported by the PFC2.

When creating a named aggregate policer, note the following:

- Policing uses the Layer 3 packet size.
- See the “PFC QoS Configuration Guidelines and Restrictions” section on page 31-31 for information about rate and burst size granularity.
- The valid range of values for the CIR `bits_per_second` parameter is as follows:
 - Minimum—32 kilobits per second, entered as 32000
 - Maximum—4 gigabits per second, entered as 4000000000
- The `normal_burst_bytes` parameter sets the CIR token bucket size.
- The `maximum_burst_bytes` parameter sets the PIR token bucket size.
- When configuring the size of a token bucket, note the following:
 - The minimum token bucket size is 1 kilobyte, entered as 1000 (the `maximum_burst_bytes` parameter must be set larger than the `normal_burst_bytes` parameter)
 - The maximum token bucket size is approximately 32 megabytes, entered as 31250000
 - To sustain a specific rate, set the token bucket size to be at least the rate value divided by 4000, because tokens are removed from the bucket every 1/4000th of a second (0.25 ms).
 - Because the token bucket must be large enough to hold at least one frame, set the parameter larger than the maximum Layer 3 packet size of the traffic being policed.
 - For TCP traffic, configure the token bucket size as a multiple of the TCP window size, with a minimum value at least twice as large as the maximum Layer 3 packet size of the traffic being policed.

The *maximum_burst_bytes* parameter is supported with PFC2. The *maximum_burst_bytes* parameter is not supported with PFC, but can be entered with a value equal to the *normal_burst_bytes* parameter.

- The valid range of values for the **pir** *bits_per_second* parameter is as follows:
 - Minimum—32 kilobits per second, entered as 32000 (the value cannot be smaller than the CIR *bits_per_second* parameters)
 - Maximum—4 gigabits per second, entered as 4000000000

The **pir** *bits_per_second* parameter is supported with the PFC2. The **pir** *bits_per_second* parameter is not supported with the PFC1 but can be entered with the PFC1 if the value is equal to the CIR *bits_per_second* parameter.

- (Optional) You can specify a conform action for matched in-profile traffic as follows:
 - The default conform action is **transmit**, which sets the policy map class trust state to *trust DSCP* unless the policy map class contains a **trust** command (see the “Policy Maps” section on page 31-18 and the “Configuring Policy Map Class Actions” section on page 31-44).
 - To set PFC QoS labels in untrusted traffic, enter the **set-dscp-transmit** keyword to mark matched untrusted traffic with a new DSCP value or enter the **set-prec-transmit** keyword to mark matched untrusted traffic with a new IP precedence value (with the PFC2, the **set-dscp-transmit** and **set-prec-transmit** keywords are only supported for IP traffic). PFC QoS sets egress ToS and CoS from the configured value.
 - Enter the **drop** keyword to drop all matched traffic.



Note When you configure **drop** as the conform action, PFC QoS configures **drop** as the exceed action and the violate action.

- (Optional) For traffic that exceeds the CIR, you can specify an exceed action as follows:
 - The default exceed action is **drop**, except with a *maximum_burst_bytes* parameter (**drop** is not supported with a *maximum_burst_bytes* parameter).



Note When the exceed action is **drop**, PFC QoS ignores any configured violate action.

- Enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map (see the “Configuring DSCP Markdown Values” section on page 31-68).



Note When you create a policer that does not use the **pir** keyword and the *maximum_burst_bytes* parameter is equal to the *normal_burst_bytes* parameter (which is the case if you do not enter the *maximum_burst_bytes* parameter), the **exceed-action policed-dscp-transmit** keywords cause PFC QoS to mark traffic down as defined by the **policed-dscp max-burst** markdown map.

- (Optional) For traffic that exceeds the PIR, you can specify a violate action as follows:
 - To mark traffic without policing, enter the **transmit** keyword to transmit all matched out-of-profile traffic.
 - The default violate action is equal to the exceed action.

- Enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map (see the “[Configuring DSCP Markdown Values](#)” section on page 31-68).
- For marking without policing, enter the **transmit** keyword to transmit all matched out-of-profile traffic.

The **violate-action** keyword is not supported with the PFC1, but the keyword can be entered with a PFC1 if the parameters match the **exceed-action** parameters.

This example shows how to create a named aggregate policer with a 1-Mbps rate limit and a 10-MB burst size that transmits conforming traffic and marks down out-of-profile traffic:

```
Router(config)# mls qos aggregate-policer aggr-1 1000000 10000000 conform-action transmit
exceed-action policed-dscp-transmit
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos aggregate-policer aggr-1
ag1 1000000 1000000 conform-action transmit exceed-action policed-dscp-transmit AgId=0
[pol4]
Router#
```

The output displays the following:

- The **AgId** parameter displays the hardware policer ID.
- The policy maps that use the policer are listed in the square brackets ([]).

Configuring a PFC QoS Policy

These sections describe PFC QoS policy configuration:

- [PFC QoS Policy Configuration Overview](#), page 31-37
- [Configuring MAC-Layer Named Access Lists \(Optional\)](#), page 31-39
- [Configuring a Class Map \(Optional\)](#), page 31-40
- [Verifying Class Map Configuration](#), page 31-41
- [Configuring a Policy Map](#), page 31-42
- [Verifying Policy Map Configuration](#), page 31-48
- [Attaching a Policy Map to an Interface](#), page 31-49



Note

PFC QoS policies process both unicast and multicast traffic.

PFC QoS Policy Configuration Overview



Note

To mark traffic without limiting bandwidth utilization, create a policer that uses the **transmit** keywords for both conforming and nonconforming traffic.

These commands configure traffic classes and the policies to be applied to those traffic classes and attach the policies to ports:

- **access-list** (Optional for IP traffic. You can filter IP traffic with **class-map** commands.):
 - PFC QoS supports these access list types:

Protocol	Numbered Access Lists?	Extended Access Lists?	Named Access Lists?
IP	Yes: 1 to 99 1300 to 1999	Yes: 100 to 199 2000 to 2699	Yes
IPX ¹	Yes: 800 to 899	Yes: 900 to 999	Yes
MAC Layer ¹	No	No	Yes ²

1. Supported with Release 12.1(1)E and later.

2. Supported with Release 12.1(1)E and later; see the “[Configuring MAC-Layer Named Access Lists \(Optional\)](#)” section on page 31-39.

- In Release 12.1(19)E and later releases, PFC QoS supports time-based Cisco IOS ACLs.
 - In Release 12.1(1)E and later releases, PFC QoS supports IPX access lists that contain a *source-network* parameter and the optional *destination-network* and *destination-node* parameters. PFC QoS does not support IPX access control lists that contain other parameters (for example, *source-node*, *protocol*, *source-socket*, *destination-socket*, or *service-type*).
 - Except for MAC-Layer named access lists (see the “[Configuring MAC-Layer Named Access Lists \(Optional\)](#)” section on page 31-39), refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, “Traffic Filtering and Firewalls,” “Access Control Lists: Overview and Guidelines,” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/index.htm
 - See [Chapter 23, “Configuring Network Security,”](#) for additional information about ACLs on the Catalyst 6500 series switches.
- **class-map** (optional)—Enter the **class-map** command to define one or more traffic classes by specifying the criteria by which traffic is classified (see the “[Configuring a Class Map \(Optional\)](#)” section on page 31-40).



Note You can also create class-maps during policy map creation with the **policy-map class** command (see the “[Creating a Policy Map Class and Configuring Filtering](#)” section on page 31-43).

- **policy-map**—Enter the **policy-map** command to define the following:
 - New class maps
 - Policy map class trust mode
 - Aggregate policing and marking
 - Microflow policing and marking
- **service-policy**—Enter the **service-policy** command to attach a policy map to an interface.

Configuring MAC-Layer Named Access Lists (Optional)

In Release 12.1(1)E and later releases, you can configure named access lists that filter DECnet, AppleTalk, VINES, or XNS traffic based on Layer 2 addresses.

To configure a MAC-Layer named access list, perform this task:

	Command	Purpose
Step 1	Router(config)# mac access-list extended <i>list_name</i>	Configures a MAC-Layer named access list.
	Router(config)# no mac access-list extended <i>list_name</i>	Deletes a MAC-Layer named access list.
Step 2	Router(config-ext-macl)# {permit deny} { <i>src_mac_mask</i> any } { <i>dest_mac_mask</i> any } [aarp amber appletalk diagnostic decnet-iv dec-spanning dsm etype-6000 etype-8042 lat lavc-sca mop-console mop-dump msdos mumps netbios vines-ip vines-echo xns]	Configures an access control entry (ACE) in a MAC-Layer named access list.
	Router(config-ext-macl)# no {permit deny} { <i>src_mac_mask</i> any } { <i>dest_mac_mask</i> any } [aarp amber appletalk diagnostic decnet-iv dec-spanning dsm etype-6000 etype-8042 lat lavc-sca mop-console mop-dump msdos mumps netbios vines-ip vines-echo xns]	Deletes an ACE from a MAC-Layer named access list.

When configuring an entry in a MAC-Layer access list, note the following:

- You can enter MAC addresses as three 4-byte values in dotted hexadecimal format. For example, 0030.9629.9f84.
- You can enter MAC address masks as three 4-byte values in dotted hexadecimal format. Use 1 bits as wildcards. For example, to match an address exactly, use 0000.0000.0000 (can be entered as 0.0.0).
- Entries without a protocol parameter match any protocol.
- Access lists entries are scanned in the order you enter them. The first matching entry is used. To improve performance, place the most commonly used entries near the beginning of the access list.
- An implicit **deny any any** entry exists at the end of an access list unless you include an explicit **permit any any** entry at the end of the list.
- All new entries to an existing list are placed at the end of the list. You cannot add entries to the middle of a list.

This list shows the ethertype values matched by the protocol keywords:

- 0x0600—xns-idp—Xerox XNS IDP
- 0x0BAD—vines-ip—Banyan VINES IP
- 0x0baf—vines-echo—Banyan VINES Echo
- 0x6000—etype-6000—DEC unassigned, experimental
- 0x6001—mop-dump—DEC Maintenance Operation Protocol (MOP) Dump/Load Assistance
- 0x6002—mop-console—DEC MOP Remote Console
- 0x6003—decnet-iv—DEC DECnet Phase IV Route
- 0x6004—lat—DEC Local Area Transport (LAT)

- 0x6005—diagnostic—DEC DECnet Diagnostics
- 0x6007—lavc-sca—DEC Local-Area VAX Cluster (LAVC), SCA
- 0x6008—amber—DEC AMBER
- 0x6009—mumps—DEC MUMPS
- 0x8038—dec-spanning—DEC LANBridge Management
- 0x8039—dsm—DEC DSM/DDP
- 0x8040—netbios—DEC PATHWORKS DECnet NETBIOS Emulation
- 0x8041—msdos—DEC Local Area System Transport
- 0x8042—etype-8042—DEC unassigned
- 0x809B—appletalk—Kinetics EtherTalk (AppleTalk over Ethernet)
- 0x80F3—arp—Kinetics AppleTalk Address Resolution Protocol (AARP)

This example shows how to create a MAC-Layer access list named **mac_layer** that denies **dec-phase-iv** traffic with source address 0000.4700.0001 and destination address 0000.4700.0009, but permits all other traffic:

```
Router(config)# mac access-list extended mac_layer
Router(config-ext-macl)# deny 0000.4700.0001 0.0.0 0000.4700.0009 0.0.0 dec-phase-iv
Router(config-ext-macl)# permit any any
```

Configuring a Class Map (Optional)

These sections describe class map configuration:

- [Creating a Class Map, page 31-40](#)
- [Configuring Filtering in a Class Map, page 31-40](#)



Note

You can also create class maps during policy map creation with the **policy-map class** command (see the “[Creating a Policy Map Class and Configuring Filtering](#)” section on page 31-43).

Creating a Class Map

To create a class map, perform this task:

Command	Purpose
Router(config)# class-map <i>class_name</i>	Creates a class map.
Router(config)# no class-map <i>class_name</i>	Deletes a class map.

Configuring Filtering in a Class Map



Note

Except for MAC-Layer ACLs (see the “[Configuring MAC-Layer Named Access Lists \(Optional\)](#)” section on page 31-39), access lists are not documented in this publication. See the reference under **access-list** in the “[Configuring a PFC QoS Policy](#)” section on page 31-37.

To configure filtering in a class map, perform one of these tasks:

Command	Purpose
Router(config-cmap)# match access-group name acl_index_or_name	(Optional) Configures the class map to filter using an ACL.
Router(config-cmap)# no match access-group name acl_index_or_name	Clears the ACL configuration from the class map.
Router (config-cmap)# match ip precedence ipp_value1 [ipp_value2 [ipp_valueN]]	(Optional—for IP traffic only) Configures the class map to filter on up to eight IP precedence values.
Router (config-cmap)# no match ip precedence ipp_value1 [ipp_value2 [ipp_valueN]]	Clears configured IP precedence values from the class map.
Router (config-cmap)# match ip dscp dscp_value1 [dscp_value2 [dscp_valueN]]	(Optional—for IP traffic only) Configures the class map to filter on up to eight DSCP values.
Router (config-cmap)# no match ip dscp dscp_value1 [dscp_value2 [dscp_valueN]]	Clears configured DSCP values from the class map.



Note

- With an MSFC2, Release 12.1(13)E and later releases support the **match protocol** class map command, which configures NBAR and sends all traffic on the Layer 3 interface, both ingress and egress, to be processed in software on the MSFC2. To configure NBAR, refer to this publication: <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm> Earlier releases provide PFC QoS and Layer 3 switching in hardware, which prevents support of the **match protocol** class map command except for traffic being processed in software on the MSFC.
- PFC QoS supports class maps that contain a single **match** command.
- PFC QoS does not support the **match cos**, **match any**, **match classmap**, **match destination-address**, **match input-interface**, **match mpls**, **match qos-group**, and **match source-address** class map commands.
- Catalyst 6500 series switches do not detect the use of unsupported commands until you attach a policy map to an interface (see the “Attaching a Policy Map to an Interface” section on page 31-49).

Verifying Class Map Configuration

To verify class map configuration, perform this task:

	Command	Purpose
Step 1	Router (config-cmap)# end	Exits configuration mode.
Step 2	Router# show class-map class_name	Verifies the configuration.

This example shows how to create a class map named **ipp5** and how to configure filtering to match traffic with IP precedence 5:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# class-map ipp5
Router(config-cmap)# match ip precedence 5
Router(config-cmap)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show class-map ipp5
Class Map match-all ipp5 (id 1)
  Match ip precedence 5

Router#
```

Configuring a Policy Map

You can attach only one policy map to an interface. Policy maps can contain one or more policy map classes, each with different policy map commands.

Configure a separate policy map class in the policy map for each type of traffic that an interface receives. Put all commands for each type of traffic in the same policy map class. PFC QoS does not attempt to apply commands from more than one policy map class to matched traffic.

These sections describe policy map configuration:

- [Creating a Policy Map, page 31-42](#)
- [Creating a Policy Map Class and Configuring Filtering, page 31-43](#)
- [Configuring Policy Map Class Actions, page 31-44](#)

Creating a Policy Map

To create a policy map, perform this task:

Command	Purpose
Router(config)# policy-map <i>policy_name</i>	Creates a policy map.
Router(config)# no policy-map <i>policy_name</i>	Deletes the policy map.

Creating a Policy Map Class and Configuring Filtering



Note

- With an MSFC2, Release 12.1(13)E and later releases support the **class** *class_name* **protocol** policy map command, which configures NBAR and sends all traffic on the Layer 3 interface, both ingress and egress, to be processed in software on the MSFC2. To configure NBAR, refer to this publication: <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm>
Earlier releases provide PFC QoS and Layer 3 switching in hardware, which prevents support of the **class** *class_name* **protocol** policy map command except for traffic being processed in software on the MSFC.
- PFC QoS does not support the **class** *class_name* **destination-address**, **class** *class_name* **input-interface**, **class** *class_name* **qos-group**, and **class** *class_name* **source-address** policy map commands.
- PFC QoS does not detect the use of unsupported commands until you attach a policy map to an interface (see the “Attaching a Policy Map to an Interface” section on page 31-49).

Policy maps can contain one or more policy map classes. Enter one of these **class** commands to create a policy map class and configure filtering in it.

To create a policy map class and configure it to filter with an already defined class map, perform this task:

Command	Purpose
Router(config-pmap)# class <i>class_name</i>	Creates a policy map class and configures it to filter with a class map (see the “Creating a Class Map” section on page 31-40). Note PFC QoS supports class maps that contain a single match command.
Router(config-pmap)# no class <i>class_name</i>	Clears use of the class map.

To create a policy map class and a class map simultaneously, perform this task:

Command	Purpose
Router(config-pmap)# class <i>class_name</i> { access-group <i>acl_index_or_name</i> dscp <i>dscp_1</i> [<i>dscp_2</i> [<i>dscp_N</i>]] precedence <i>ipp_1</i> [<i>ipp_2</i> [<i>ipp_N</i>]]}	Creates a policy map class and creates a class map and configures the policy map class to filter with the class map. Note This command creates a class map that can be used in other policy maps.
Router(config-pmap)# no class <i>class_name</i>	Clears use of the class map (does not delete the class map).



Note

- Put all trust-state and policing commands for each type of traffic in the same policy map class.
- PFC QoS does not attempt to apply commands from more than one policy map class to traffic.

Configuring Policy Map Class Actions

When configuring policy map class actions, note the following:

- For hardware-switched traffic, PFC QoS does not support the **bandwidth**, **priority**, **queue-limit**, or **random-detect** policy map class commands. You can configure these commands because they can be used for software-switched traffic.
- With Release 12.1(12c)E1 and later releases, PFC QoS does not support the **set mpls** or **set qos-group** policy map class commands. With earlier releases, PFC QoS does not support any **set** policy map class commands.
- With Release 12.1(12c)E1 and later releases, PFC QoS supports the **set ip dscp** and **set ip precedence** policy map class commands (see the “[Configuring Policy Map Class Marking](#)” section on page 31-44).
- With Release 12.1(12c)E1 and later releases, you cannot do all three of the following in a policy map class:
 - Mark traffic with the **set ip dscp** or **set ip precedence** commands
 - Configure the trust state
 - Configure policing

In a policy map class, you can either mark untrusted traffic with the **set ip dscp** or **set ip precedence** commands or do one or both of the following:

- Configure the trust state
- Configure policing



Note When configure policing, you can mark traffic with policing keywords (see the “[Configuring Policy Map Class Policing](#)” section on page 31-45).

These sections describe policy map class action configuration:

- [Configuring Policy Map Class Marking](#), page 31-44
- [Configuring the Policy Map Class Trust State](#), page 31-45
- [Configuring Policy Map Class Policing](#), page 31-45

Configuring Policy Map Class Marking

With Release 12.1(12c)E1 and later releases, PFC QoS supports policy map class marking for untrusted traffic with the **set ip dscp** and **set ip precedence** policy map class commands.

To configure policy map class marking for untrusted traffic, perform this task:

Command	Purpose
Router(config-pmap-c)# set ip { dscp <i>dscp_value</i> precedence <i>ip_precedence_value</i> }	Configures the policy map class to mark matched untrusted traffic with the configured DSCP or IP precedence value.
Router(config-pmap-c)# no set ip { dscp <i>dscp_value</i> precedence <i>ip_precedence_value</i> }	Clears the marking configuration.

Configuring the Policy Map Class Trust State

To configure the policy map class trust state, perform this task:

Command	Purpose
Router(config-pmap-c)# trust { <i>cos</i> <i>dscp</i> <i>ip-precedence</i> }	Configures the policy map class trust state, which selects the value that PFC QoS uses as the source of the internal DSCP value (see the “ Internal DSCP Values ” section on page 31-17).
Router(config-pmap-c)# no trust	Reverts to the default policy-map class trust state (untrusted).

When configuring the policy map class trust state, note the following:

- Enter the **no trust** command to use the trust state configured on the ingress port (this is the default).
- With the **cos** keyword, PFC QoS sets the internal DSCP value from received or ingress port CoS (see the “[Mapping Received CoS Values to Internal DSCP Values](#)” section on page 31-66).
- With the **dscp** keyword, PFC QoS uses received DSCP.
- With the **ip-precedence** keyword, PFC QoS sets DSCP from received IP precedence (see the “[Mapping Received IP Precedence Values to Internal DSCP Values](#)” section on page 31-67).

Configuring Policy Map Class Policing

When you configure policy map class policing, note the following:

- PFC QoS does not support the **set-qos-transmit** policer keyword.
- PFC QoS does not support the **set-dscp-transmit** or **set-prec-transmit** keywords as arguments to the **exceed-action** keyword.
- PFC QoS does not detect the use of unsupported keywords until you attach a policy map to an interface (see the “[Attaching a Policy Map to an Interface](#)” section on page 31-49).

These sections describe configuration of policy map class policing:

- [Using a Named Aggregate Policer, page 31-45](#)
- [Configuring a Per-Interface Policer, page 31-46](#)



Note

Policing with the **conform-action transmit** keywords sets the port trust state of matched traffic to trust DSCP or to the trust state configured by a **trust** command in the policy map class.

Using a Named Aggregate Policer

To use a named aggregate policer (see the “[Creating Named Aggregate Policers](#)” section on page 31-35), perform this task:

Command	Purpose
Router(config-pmap-c)# police aggregate <i>aggregate_name</i>	Configures the policy map class to use a previously defined named aggregate policer.
Router(config-pmap-c)# no police aggregate <i>aggregate_name</i>	Clears use of the named aggregate policer.

Configuring a Per-Interface Policer

To configure a per-interface policer (see the “Policers” section on page 31-19), perform this task:

Command	Purpose
<pre>Router(config-pmap-c)# police [flow] bits_per_second normal_burst_bytes [maximum_burst_bytes] [pir¹ peak_rate_bps] [[[conform-action {drop set-dscp-transmit² dscp_value set-prec-transmit² ip_precedence_value transmit}] exceed-action {drop policed-dscp transmit}] violate-action¹ {drop policed-dscp transmit}]</pre> <pre>Router(config-pmap-c)# no police [flow] bits_per_second normal_burst_bytes [maximum_burst_bytes] [pir peak_rate_bps] [[[conform-action {drop set-dscp-transmit dscp_value set-prec-transmit ip_precedence_value transmit}] exceed-action {drop policed-dscp transmit}] violate-action {drop policed-dscp transmit}]</pre>	<p>Creates a per-interface policer and configures the policy map class to use it.</p> <p>Deletes the per-interface policer from the policy map class.</p>

1. Supported only with PFC2. Not supported in microflow policers (the **flow** keyword configures a microflow policer).
2. With PFC2, the **set-dscp-transmit** and **set-prec-transmit** keywords are only supported for IP traffic.

When configuring a per-interface policer, note the following:

- Policing uses the Layer 3 packet size.
- See the “PFC QoS Configuration Guidelines and Restrictions” section on page 31-31 for information about rate and burst size granularity.
- You can enter the **flow** keyword to define a microflow policer. During microflow policing, the following occurs:
 - PFC QoS considers IPX traffic with same source network, destination network, and destination node to be part of the same flow, including traffic with different source nodes or sockets.
 - PFC QoS considers MAC-Layer traffic with the same protocol and the same source and destination MAC-Layer addresses to be part of the same flow, including traffic with different ethertypes.
 - Microflow policers do not support the *maximum_burst_bytes* parameter, the **pir bits_per_second** keyword and parameter, or the **violate-action** keyword.
- The valid range of values for the CIR *bits_per_second* parameter is as follows:
 - Minimum—32 kilobits per second, entered as 32000
 - Maximum—4 gigabits per second, entered as 4000000000
- The *normal_burst_bytes* parameter sets the CIR token bucket size.
- The *maximum_burst_bytes* parameter sets the PIR token bucket size (not supported with the **flow** keyword)
- When configuring the size of a token bucket, note the following:
 - The minimum token bucket size is 1 kilobyte, entered as 1000 (the *maximum_burst_bytes* parameter must be set larger than the *normal_burst_bytes* parameter)
 - The maximum token bucket size is approximately 32 megabytes, entered as 31250000
 - To sustain a specific rate, set the token bucket size to be at least the rate value divided by 4000, because tokens are removed from the bucket every 1/4000th of a second (0.25 ms).

- Because the token bucket must be large enough to hold at least one frame, set the parameter larger than the maximum Layer 3 packet size of the traffic being policed.
- For TCP traffic, configure the token bucket size as a multiple of the TCP window size, with a minimum value at least twice as large as the maximum Layer 3 packet size of the traffic being policed.

The *maximum_burst_bytes* parameter is supported with the PFC2. The *maximum_burst_bytes* parameter is not supported with the PFC1, but the keyword can be entered with a value equal to the *normal_burst_bytes* parameter.

- (Not supported with the **flow** keyword.) The valid range of values for the **pir bits_per_second** parameter is as follows:
 - Minimum—32 kilobits per second, entered as 32000 (the value cannot be smaller than the CIR *bits_per_second* parameters)
 - Maximum—4 gigabits per second, entered as 4000000000

The **pir bits_per_second** parameter is supported with the PFC2. The **pir bits_per_second** parameter is not supported with the PFC1, but can be entered with the PFC1 if the value is equal to the CIR *bits_per_second* parameter.

- (Optional) You can specify a conform action for matched in-profile traffic as follows:
 - The default conform action is **transmit**, which sets the policy map class trust state to *trust DSCP* unless the policy map class contains a **trust** command (see the “Policy Maps” section on page 31-18 and the “Configuring Policy Map Class Actions” section on page 31-44).
 - To set PFC QoS labels in untrusted traffic, you can enter the **set-dscp-transmit** keyword to mark matched untrusted traffic with a new DSCP value or enter the **set-prec-transmit** keyword to mark matched untrusted traffic with a new IP precedence value (with the PFC2, the **set-dscp-transmit** and **set-prec-transmit** keywords are only supported for IP traffic). PFC QoS sets egress ToS and CoS from the configured value.
 - You can enter the **drop** keyword to drop all matched traffic.
 - Ensure that aggregate and microflow policers that are applied to the same traffic each specify the same conform-action behavior.
- (Optional) For traffic that exceeds the CIR, you can specify an exceed action as follows:
 - For marking without policing, you can enter the **transmit** keyword to transmit all matched out-of-profile traffic.
 - The default exceed action is **drop**, except with a *maximum_burst_bytes* parameter (**drop** is not supported with a *maximum_burst_bytes* parameter).



Note When the exceed action is **drop**, PFC QoS ignores any configured violate action.

- You can enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map (see the “Configuring DSCP Markdown Values” section on page 31-68).



Note When you create a policer that does not use the **pir** keyword and the *maximum_burst_bytes* parameter is equal to the *normal_burst_bytes* parameter (which is the case if you do not enter the *maximum_burst_bytes* parameter), the **exceed-action policed-dscp-transmit** keywords cause PFC QoS to mark traffic down as defined by the **policed-dscp max-burst** markdown map.

- (Optional—Not supported with the **flow** keyword) For traffic that exceeds the PIR, you can specify a violate action as follows:
 - For marking without policing, you can enter the **transmit** keyword to transmit all matched out-of-profile traffic.
 - The default violate action is equal to the exceed action.
 - You can enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map (see the “[Configuring DSCP Markdown Values](#)” section on page 31-68).

The **violate-action** keyword is not supported with the PFC1, but the keyword can be entered with the PFC1 if the parameters match the **exceed-action** parameters.



Note

Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC2, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC2 and any non-DFC-equipped switching modules supported by the PFC2.

This example shows how to create a policy map named **max-pol-ipp5** that uses the class-map named **ipp5**, which is configured to trust received IP precedence values and is configured with a maximum-capacity aggregate policer and with a microflow policer:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# policy-map max-pol-ipp5
Router(config-pmap)# class ipp5
Router(config-pmap-c)# trust ip-precedence
Router(config-pmap-c)# police 200000000 200000 conform-action set-prec-transmit 6
exceed-action policed-dscp-transmit
Router(config-pmap-c)# police flow 1000000 10000 conform-action set-prec-transmit 6
exceed-action policed-dscp-transmit
Router(config-pmap-c)# end
```

Verifying Policy Map Configuration

To verify policy map configuration, perform this task:

	Command	Purpose
Step 1	Router(config-pmap-c)# end	Exits policy map class configuration mode. Note Enter additional class commands to create additional classes in the policy map.
Step 2	Router# show policy-map <i>policy_name</i>	Verifies the configuration.

This example shows how to verify the configuration:

```
Router# show policy-map max-pol-ipp5
Policy Map max-pol-ipp5
  class ipp5

  class ipp5
    police flow 10000000 10000 conform-action set-prec-transmit 6 exceed-action
    policed-dscp-transmit
    trust precedence
    police 2000000000 2000000 2000000 conform-action set-prec-transmit 6 exceed-action
    policed-dscp-transmit

Router#
```

Attaching a Policy Map to an Interface



Note PFC QoS does not support the **output** service-policy keyword.

To attach a policy map to an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# service-policy input policy_map_name	Attaches a policy map to the input direction of the interface.
	Router(config-if)# no service-policy input policy_map_name	Removes the policy map from the interface.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show policy-map interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, ge-wan, pos, or atm



Note Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC2, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC2 and any non-DFC-equipped switching modules supported by the PFC2.

This example shows how to attach the policy map named **pmap1** to Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# service-policy input pmap1
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show policy-map interface fastethernet 5/36
FastEthernet5/36
  service-policy input: pmap1
    class-map: cmap1 (match-all)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 5
    class cmap1
      police 8000 8000 conform-action transmit exceed-action drop
      class-map: cmap2 (match-any)
        0 packets, 0 bytes
        5 minute rate 0 bps
        match: ip precedence 2
          0 packets, 0 bytes
          5 minute rate 0 bps
      class cmap2
        police 8000 10000 conform-action transmit exceed-action drop
Router#
```

Enabling or Disabling Microflow Policing

To enable or disable microflow policing (see the “Policers” section on page 31-19), perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos flow-policing	Enables microflow policing.
	Router(config)# no mls qos flow-policing	Disables microflow policing.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos	Verifies the configuration.

This example shows how to disable microflow policing:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# no mls qos flow-policing
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos | include Microflow
Microflow QoS is disabled globally
Router#
```

Enabling Microflow Policing of Bridged Traffic



Note

To apply microflow policing to multicast traffic, you must enter the **mls qos bridged** command on the Layer 3 multicast ingress interfaces.

By default, microflow policers affect only routed traffic. To enable microflow policing of bridged traffic on specified VLANs, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port}}	Selects the interface to configure.
Step 2	Router(config-if)# mls qos bridged	Enables microflow policing of bridged traffic, including bridge groups, on the VLAN.
	Router(config-if)# no mls qos bridged	Disables microflow policing of bridged traffic.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show mls qos	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable microflow policing of bridged traffic on VLANs 3 through 5:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface range vlan 3 - 5
Router(config-if)# mls qos bridged
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos | begin Bridged QoS
Bridged QoS is enabled on the following interfaces:
    V13 V14 V15
<...output truncated...>
Router#
```

Enabling or Disabling PFC Features on an Interface

You can enable or disable the PFC QoS features implemented on the PFC for traffic from an interface (see the “PFC Marking and Policing” section on page 31-16). Disabling the PFC QoS features on an interface leaves the configuration intact. The **mls qos** interface command reenables any previously configured PFC QoS features. The **mls qos** interface command does not affect the port queuing configuration.

To enable or disable PFC features for traffic from an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# mls qos	Enables PFC QoS on the interface.
	Router(config-if)# no mls qos	Disables PFC QoS on the interface.
Step 3	Router(config-if)# end	Exits configuration interface.
Step 4	Router# show mls qos	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, ge-wan, pos, or atm

This example shows how to disable PFC QoS on the VLAN 5 interface:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface vlan 5
Router(config-if)# no mls qos
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos | begin QoS is disabled
QoS is disabled on the following interfaces:
V15
<...Output Truncated...>
Router#
```

Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports



Note

With DFCs installed, Supervisor Engine 2 does not support VLAN-based PFC QoS.

By default, PFC QoS uses policy maps attached to LAN ports. For ports configured as Layer 2 LAN ports with the **switchport** keyword, you can configure PFC QoS to use policy maps attached to a VLAN (see the “Attaching Policy Maps” section on page 31-21). Ports not configured with the **switchport** keyword are not associated with a VLAN.

To enable VLAN-based PFC QoS on a Layer 2 LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# mls qos vlan-based Router(config-if)# no mls qos vlan-based	Enables VLAN-based PFC QoS on a Layer 2 LAN port. Disables VLAN-based PFC QoS.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show mls qos	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable VLAN-based PFC QoS on Fast Ethernet port 5/42:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/42
Router(config-if)# mls qos vlan-based
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show mls qos | begin QoS is vlan-based
QoS is vlan-based on the following interfaces:
Fa5/42
<...Output Truncated...>
```

**Note**

Configuring a Layer 2 LAN port for VLAN-based PFC QoS preserves the policy map port configuration. The **no mls qos vlan-based** port command reenables any previously configured port commands.

Configuring the Trust State of Ethernet LAN and OSM Ingress Ports

By default, all ingress ports are untrusted. You can configure the ingress port trust state on all Ethernet LAN ports except non-Gigabit Ethernet **1q4t/2q2t** ports (see the “[Ingress LAN Port Features](#)” section on page 31-12). You can configure the ingress port trust state on OSM ports (see the “[Ingress OSM Port Features](#)” section on page 31-11).

To configure the trust state of an ingress port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# mls qos trust [dscp ip-precedence cos ²]	Configures the trust state of the port.
	Router(config-if)# no mls qos trust	Reverts to the default trust state (untrusted).
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show mls qos	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, ge-wan, pos, or atm.
2. Not supported for pos or atm interface types.

When configuring the trust state of an ingress port, note the following:

- With no other keywords, the **mls qos trust** command is equivalent to **mls qos trust dscp**.
- The **mls qos trust cos** command enables receive-queue drop thresholds. To avoid dropping traffic because of inconsistent CoS values, configure ports with the **mls qos trust cos** command only when the received traffic is ISL or 802.1Q frames carrying CoS values that you know to be consistent with network policy.
- Use the **no mls qos trust** command to set the port state to untrusted.

This example shows how to configure Gigabit Ethernet port 1/1 with the **trust cos** keywords:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# mls qos trust cos
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | include trust
Trust state: trust COS
Router#
```

Configuring the Ingress LAN Port CoS Value



Note

Whether or not PFC QoS uses the CoS value applied with the **mls qos cos** command depends on the trust state of the port and the trust state of the traffic received through the port. The **mls qos cos** command does not configure the trust state of the port or the trust state of the traffic received through the port.

To use the CoS value applied with the **mls qos cos** command as the basis of internal DSCP (see the “[Internal DSCP Values](#)” section on page 31-17):

- On a port that receives only untagged ingress traffic, configure the ingress port as trusted or configure a trust-CoS policy map that matches the ingress traffic.
- On a port that receives tagged ingress traffic, configure a trust-CoS policy map that matches the ingress traffic.

You can configure the CoS value that PFC QoS assigns to untagged frames from ingress LAN ports configured as trusted and to all frames from ingress LAN ports configured as untrusted.

To configure the CoS value for an ingress LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects the interface to configure.
Step 2	Router(config-if)# mls qos cos default_cos Router(config-if)# no mls qos cos default_cos	Configures the ingress LAN port CoS value. Reverts to the default port CoS value.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface {ethernet fastethernet gigabitethernet} slot/port	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure the CoS 5 as the default on Fast Ethernet port 5/24 and verify the configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/24
Router(config-if)# mls qos cos 5
Router(config-if)# end
Router# show queueing interface fastethernet 5/24 | include Default COS
Default COS is 5
Router#
```

Configuring Standard-Queue Drop Threshold Percentages

These sections describe configuring standard-queue drop threshold percentages:

- [Configuring a Tail-Drop Receive Queue, page 31-55](#)
- [Configuring a WRED-Drop Transmit Queue, page 31-56](#)
- [Configuring a WRED-Drop and Tail-Drop Transmit Queue, page 31-57](#)
- [Configuring 1q4t/2q2t Tail-Drop Threshold Percentages, page 31-58](#)

**Note**

- Enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command to see the queue structure of a port (see the “Receive Queues” section on page 31-13 and the “Transmit Queues” section on page 31-21).
- **1p1q0t** ports have no configurable thresholds.
- **1p3q1t** (transmit), **1p2q1t** (transmit), and **1p1q8t** (receive) ports also have nonconfigurable tail-drop thresholds (see the “Mapping CoS Values to Standard Transmit-Queue Thresholds” section on page 31-61).

When configuring thresholds, note the following:

- Queue number 1 is the lowest-priority standard queue.
- Higher-numbered queues are higher priority standard queues.

When you configure multiple-threshold standard queues, note the following:

- The first percentage that you enter sets the lowest-priority threshold.
- The second percentage that you enter sets the next highest-priority threshold.
- The last percentage that you enter sets the highest-priority threshold.
- The percentages range from 1 to 100. A value of 10 indicates a threshold when the buffer is 10-percent full.
- Always set highest-numbered threshold to 100 percent.

When configuring the WRED-drop thresholds, note the following:

- Each WRED-drop threshold has a low-WRED and a high-WRED value.
- Low-WRED and high-WRED values are a percentage of the queue capacity (the range is from 1 to 100).
- The low-WRED value is the traffic level under which no traffic is dropped. The low-WRED value must be lower than the high-WRED value.
- The high-WRED value is the traffic level above which all traffic is dropped.
- Traffic in the queue between the low- and high-WRED values has an increasing chance of being dropped as the queue fills.

Configuring a Tail-Drop Receive Queue

These port types have only tail-drop thresholds in their receive-queues:

- **1p1q4t**
- **1q2t**

To configure the drop thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {fastethernet gigabitethernet} slot/port	Selects the interface to configure.
Step 2	Router(config-if)# rcv-queue threshold queue_id thr1% thr2% thr3% thr4% {thr5% thr6% thr7% thr8%}	Configures the receive-queue tail-drop threshold percentages.
	Router(config-if)# no rcv-queue threshold [queue_id]	Reverts to the default receive-queue tail-drop threshold percentages.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface {fastethernet gigabitethernet} slot/port	Verifies the configuration.

This example shows how to configure the receive-queue drop thresholds for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# rcv-queue threshold 1 60 75 85 100
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | begin Receive queues
Receive queues [type = 1p1q4t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          Standard      4
      2          Priority       1

Trust state: trust COS

  queue tail-drop-thresholds
  -----
      1      60[1] 75[2] 85[3] 100[4]
<...Output Truncated...>
Router#
```

Configuring a WRED-Drop Transmit Queue

These port types have only WRED-drop thresholds in their transmit queues:

- **1p2q2t**
- **1p2q1t**

To configure the drop thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface type ¹ slot/port	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue random-detect min-threshold queue_id thr1% [thr2%]	Configures the low WRED-drop thresholds.
	Router(config-if)# no wrr-queue random-detect min-threshold [queue_id]	Reverts to the default low WRED-drop thresholds.

	Command	Purpose
Step 3	Router(config-if)# wrr-queue random-detect max-threshold queue_id thr1% [thr2%]	Configures the high WRED-drop thresholds.
	Router(config-if)# no wrr-queue random-detect max-threshold [queue_id]	Reverts to the default high WRED-drop thresholds.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show queueing interface type ¹ slot/port	Verifies the configuration.

1. type = fastethernet, gigabitethernet, or tengigabitethernet

Configuring a WRED-Drop and Tail-Drop Transmit Queue

1p3q1t ports have both WRED-drop and tail-drop thresholds in their transmit queues.

To configure the drop thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface type ¹ slot/port	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue threshold queue_id thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]	Configures the tail-drop thresholds.
	Router(config-if)# no wrr-queue threshold [queue_id]	Reverts to the default tail-drop thresholds.
Step 3	Router(config-if)# wrr-queue random-detect min-threshold queue_id thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]	Configures the low WRED-drop thresholds.
	Router(config-if)# no wrr-queue random-detect min-threshold [queue_id]	Reverts to the default low WRED-drop thresholds.
Step 4	Router(config-if)# wrr-queue random-detect max-threshold queue_id thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]	Configures the high WRED-drop thresholds.
	Router(config-if)# no wrr-queue random-detect max-threshold [queue_id]	Reverts to the default high WRED-drop thresholds.
Step 5	Router(config-if)# wrr-queue random-detect queue_id	Enables WRED-drop thresholds .
	Router(config-if)# no wrr-queue random-detect [queue_id]	Enables tail-drop thresholds.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show queueing interface type ¹ slot/port	Verifies the configuration.

1. type = fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure the low-priority transmit queue high-WRED-drop thresholds for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# wrr-queue random-detect max-threshold 1 70 70
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | begin Transmit queues
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          WRR low          2
      2          WRR high          2
      3          Priority          1

  queue random-detect-max-thresholds
  -----
      1      40[1] 70[2]
      2      40[1] 70[2]
<...Output Truncated...>
Router#
```

Configuring 1q4t/2q2t Tail-Drop Threshold Percentages

On **1q4t/2q2t** ports, the receive- and transmit-queue drop thresholds have this relationship:

- Receive queue 1 (standard) threshold 1 = transmit queue 1 (standard low priority) threshold 1
- Receive queue 1 (standard) threshold 2 = transmit queue 1 (standard low priority) threshold 2
- Receive queue 1 (standard) threshold 3 = transmit queue 2 (standard high priority) threshold 1
- Receive queue 1 (standard) threshold 4 = transmit queue 2 (standard high priority) threshold 2

To configure tail-drop threshold percentages for the standard receive and transmit queues on **1q4t/2q2t** LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# interface { ethernet fastethernet gigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue threshold <i>queue_id</i> <i>thr1% thr2%</i> Router(config-if)# no wrr-queue threshold [<i>queue_id</i>]	Configures the receive- and transmit-queue tail-drop thresholds. Reverts to the default receive- and transmit-queue tail-drop thresholds.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface { ethernet fastethernet gigabitethernet } <i>slot/port</i>	Verifies the configuration.

When configuring the receive- and transmit-queue tail-drop thresholds, note the following:

- You must use the transmit queue and threshold numbers.
- The *queue_id* is 1 for the standard low-priority queue and 2 for the standard high-priority queue.
- The percentages range from 1 to 100. A value of 10 indicates a threshold when the buffer is 10-percent full.
- Always set threshold 2 to 100 percent.
- Ethernet and Fast Ethernet **1q4t** ports do not support receive-queue tail-drop thresholds.

This example shows how to configure receive queue 1/threshold 1 and transmit queue 1/threshold 1 for Gigabit Ethernet port 2/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 2/1
Router(config-if)# wrr-queue threshold 1 60 100
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 2/1
  Transmit queues [type = 2q2t]:

<...Output Truncated...>

queue tail-drop-thresholds
-----
  1      60[1] 100[2]
  2      40[1] 100[2]

<...Output Truncated...>

  Receive queues [type = 1q4t]:

<...Output Truncated...>

queue tail-drop-thresholds
-----
  1      60[1] 100[2] 40[3] 100[4]
<...Output Truncated...>
Router#
```

Mapping CoS Values to Drop Thresholds

These sections describe mapping CoS values to drop thresholds:



Note

Enter the **show queueing interface {ethernet | fastethernet | gigabitethernet | tengigabitethernet} slot/port | include type** command to see the queue structure of a port.

These sections describe how to map CoS values:

- [Mapping CoS Values to Standard Receive-Queue Thresholds](#), page 31-60
- [Mapping CoS Values to Standard Transmit-Queue Thresholds](#), page 31-61
- [Mapping CoS Values to Strict-Priority Queues](#), page 31-61
- [Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports](#), page 31-62

When CoS values to thresholds, note the following:

- Queue number 1 is the lowest-priority standard queue.
- Higher-numbered queues are higher priority standard queues.
- You can map up to 8 CoS values to a threshold.

- Threshold 0 is a nonconfigurable 100-percent tail-drop threshold on these port types:
 - **1p1q0t** (receive)
 - **1p1q8t** (receive)
 - **1p3q1t** (transmit)
 - **1p2q1t** (transmit)
- The standard queue thresholds can be configured as either tail-drop or WRED-drop thresholds on these port types:
 - **1p1q8t** (receive)
 - **1p3q1t** (transmit)

See the “[Configuring Standard-Queue Drop Threshold Percentages](#)” section on page 31-54 for more information about configuring thresholds as either tail-drop or WRED-drop.

Mapping CoS Values to Standard Receive-Queue Thresholds

To map CoS values to the standard receive-queue thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface { fastethernet gigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# rcv-queue cos-map <i>queue_# threshold_# cos1 [cos2 [cos3 [cos4 [cos5 [cos6 [cos7 [cos8]]]]]]]</i> Router(config-if)# no rcv-queue cos-map	Maps CoS values to the standard receive queue thresholds. Reverts to the default mapping.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface { fastethernet gigabitethernet } <i>slot/port</i>	Verifies the configuration.

This example shows how to map the CoS values 0 and 1 to threshold 1 in the standard receive queue for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# rcv-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1
<...Output Truncated...>
  queue thresh cos-map
  -----
  1      1      0 1
  1      2      2 3
  1      3      4 5
  1      4      6 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Standard Transmit-Queue Thresholds

To map CoS values to standard transmit-queue thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {fastethernet gigabitethernet} slot/port	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue cos-map transmit_queue_# threshold_# cos1 [cos2 [cos3 [cos4 [cos5 [cos6 [cos7 [cos8]]]]]]]	Maps CoS values to a standard transmit-queue threshold.
	Router(config-if)# no wrr-queue cos-map	Reverts to the default mapping.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface {fastethernet gigabitethernet} slot/port	Verifies the configuration.

This example shows how to map the CoS values 0 and 1 to standard transmit queue 1/threshold 1 for Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# wrr-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 5/36 | begin queue thresh cos-map
queue thresh cos-map
-----
1      1      0 1
1      2      2 3
2      1      4 5
2      2      6 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Strict-Priority Queues

To map CoS values to the receive and transmit strict-priority queues, perform this task:

	Command	Purpose
Step 1	Router(config)# interface type ¹ slot/port	Selects the interface to configure.
Step 2	Router(config-if)# priority-queue cos-map queue_# cos1 [cos2 [cos3 [cos4 [cos5 [cos6 [cos7 [cos8]]]]]]]	Maps CoS values to the receive and transmit strict-priority queues.
	Router(config-if)# no priority-queue cos-map	Reverts to the default mapping.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface type ¹ slot/port	Verifies the configuration.

1. type = fastethernet, gigabitethernet, or tengigabitethernet

When mapping CoS values to the strict-priority queues, note the following:

- The queue number is always 1.
- You can enter up to 8 CoS values to map to the queue.

This example shows how to map CoS value 7 to the strict-priority queues on Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# priority-queue cos-map 1 7
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1
<...Output Truncated...>
Transmit queues [type = 1p2q2t]:
<...Output Truncated...>
  queue thresh cos-map
-----
  1      1      0 1
  1      2      2 3
  2      1      4
  2      2      6
  3      1      5 7

Receive queues [type = 1p1q4t]:
<...Output Truncated...>
  queue thresh cos-map
-----
  1      1      0 1
  1      2      2 3
  1      3      4
  1      4      6
  2      1      5 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports



Note

Enter the `show queueing interface { ethernet | fastethernet | gigabitethernet | tengigabitethernet } slot/port include type` command to see the queue structure of a port.

On **1q4t/2q2t** LAN ports, the receive- and transmit-queue tail-drop thresholds have this relationship:

- Receive queue 1 (standard) threshold 1 = transmit queue 1 (standard low priority) threshold 1
- Receive queue 1 (standard) threshold 2 = transmit queue 1 (standard low priority) threshold 2
- Receive queue 1 (standard) threshold 3 = transmit queue 2 (standard high priority) threshold 1
- Receive queue 1 (standard) threshold 4 = transmit queue 2 (standard high priority) threshold 2

To map CoS values to tail-drop thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue cos-map <i>transmit_queue_# threshold_# cos1 [cos2 [cos3</i> <i>[cos4 [cos5 [cos6 [cos7 [cos8]]]]]]]</i>	Maps CoS values to a tail-drop threshold.
Step 3	Router(config-if)# no wrr-queue cos-map	Reverts to the default mapping.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When mapping CoS values to a tail-drop threshold, note the following:

- Use the transmit queue and threshold numbers.
- Queue 1 is the low-priority standard transmit queue.
- Queue 2 is the high-priority standard transmit queue.
- There are two thresholds in each queue.
- Enter up to 8 CoS values to map to the threshold.

This example shows how to map the CoS values 0 and 1 to standard transmit queue 1/threshold 1 for Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# wrr-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 5/36 | begin queue thresh cos-map
queue thresh cos-map
-----
1      1      0 1
1      2      2 3
2      1      4 5
2      2      6 7
<...Output Truncated...>
Router#
```

Allocating Bandwidth Between LAN-Port Transmit Queues

The switch transmits frames from one standard queue at a time using a WRR algorithm. WRR uses the ratio between queue weight values to decide how much to transmit from one queue before switching to the other. The more the ratio favors a queue, the more transmit bandwidth is allocated to it.

To allocate bandwidth for an egress LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue bandwidth <i>low_priority_queue_weight</i> [<i>medium_priority_queue_weight</i>] <i>high_priority_queue_weight</i>	Allocates bandwidth between standard transmit queues. The valid values for weight range from 1 to 255.
	Router(config-if)# no wrr-queue bandwidth	
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to allocate a 3-to-1 bandwidth ratio for Gigabit Ethernet port 1/2:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/2
Router(config-if)# wrr-queue bandwidth 3 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/2 | include bandwidth
WRR bandwidth ratios: 3[queue 1] 1[queue 2]
Router#
```

Setting the Receive-Queue Size Ratio on a 1p1q0t or 1p1q8t Ingress LAN Ports

To set the size ratio between the strict-priority and standard receive queues on a 1p1q0t or 1p1q8t ingress LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# interface { fastethernet tengigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# rcv-queue queue-limit <i>standard_queue_weight</i> <i>strict_priority_queue_weight</i>	Sets the size ratio between the strict-priority and standard receive queues.
	Router(config-if)# no rcv-queue queue-limit	
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface { fastethernet tengigabitethernet } <i>slot/port</i>	Verifies the configuration.

When setting the receive-queue size ratio, note the following:

- The **rcv-queue queue-limit** command configures ports on a per-ASIC basis.
- Estimate the mix of strict priority-to-standard traffic on your network (for example, 80 percent standard traffic and 20 percent strict-priority traffic).
- Use the estimated percentages as queue weights.
- Valid values are from 1 to 100 percent, except on **1p1q8t** ingress LAN ports, where valid values for the strict priority queue are from 3 to 100 percent.

This example shows how to set the receive-queue size ratio for Fast Ethernet port 2/2:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 2/2
Router(config-if)# rcv-queue queue-limit 75 15
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 2/2 | include queue-limit
queue-limit ratios:      75[queue 1] 15[queue 2]
Router#
```

Setting the LAN-Port Transmit-Queue Size Ratio

To set the transmit-queue size ratio on an egress LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue queue-limit <i>low_priority_queue_weight</i> <i>[medium_priority_queue_weight]</i> <i>high_priority_queue_weight</i> Router(config-if)# no wrr-queue queue-limit	Sets the transmit-queue size ratio between transmit queues. Reverts to the default transmit-queue size ratio.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

When setting the transmit-queue size ratio between transmit queues, note the following:

- Estimate the mix of low priority-to-high priority traffic on your network (for example, 80 percent low-priority traffic and 20 percent high-priority traffic).
- On LAN ports that have an egress strict priority queue, PFC QoS sets the egress strict-priority queue size equal to the high-priority queue size.
- Use the estimated percentages as queue weights.
- Valid values are from 1 to 100 percent, except on **1p2q1t** egress LAN ports, where valid values for the high priority queue are from 5 to 100 percent.

This example shows how to set the transmit-queue size ratio for Gigabit Ethernet port 1/2:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/2
Router(config-if)# wrr-queue queue-limit 75 15
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/2 | include queue-limit
queue-limit ratios: 75[queue 1] 25[queue 2]
Router#
```

Configuring DSCP Value Maps

These sections describe how DSCP values are mapped to other values:

- [Mapping Received CoS Values to Internal DSCP Values, page 31-66](#)
- [Mapping Received IP Precedence Values to Internal DSCP Values, page 31-67](#)
- [Mapping Internal DSCP Values to Egress CoS Values, page 31-67](#)
- [Configuring DSCP Markdown Values, page 31-68](#)

Mapping Received CoS Values to Internal DSCP Values

To configure the mapping of received CoS values to the DSCP value that PFC QoS uses internally on the PFC (see the “[Internal DSCP Values](#)” section on page 31-17), perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos map cos-dscp dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8 Router(config)# no mls qos map cos-dscp	Configures the received CoS to internal DSCP map. You must enter 8 DSCP values to which PFC QoS maps CoS values 0 through 7. Reverts to the default map.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos maps	Verifies the configuration.

This example shows how to configure the received CoS to internal DSCP map:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls qos map cos-dscp 0 1 2 3 4 5 6 7
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos maps | begin Cos-dscp map
Cos-dscp map:
cos: 0 1 2 3 4 5 6 7
-----
dscp: 0 1 2 3 4 5 6 7
<...Output Truncated...>
Router#
```


Mapping Received IP Precedence Values to Internal DSCP Values

To configure the mapping of received IP precedence values to the DSCP value that PFC QoS uses internally on the PFC (see the “[Internal DSCP Values](#)” section on page 31-17), perform this task:

	Command	Purpose
Step 1	<pre>Router(config)# mls qos map ip-prec-dscp dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8 Router(config)# no mls qos map ip-prec-dscp</pre>	<p>Configures the received IP precedence to internal DSCP map. You must enter 8 internal DSCP values to which PFC QoS maps received IP precedence values 0 through 7.</p> <p>Reverts to the default map.</p>
Step 2	<pre>Router(config)# end</pre>	Exits configuration mode.
Step 3	<pre>Router# show mls qos maps</pre>	Verifies the configuration.

This example shows how to configure the received IP precedence to internal DSCP map:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls qos map ip-prec-dscp 0 1 2 3 4 5 6 7
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos maps | begin IpPrecedence-dscp map
IpPrecedence-dscp map:
  ipprec:  0  1  2  3  4  5  6  7
  -----
  dscp:    0  1  2  3  4  5  6  7
<...Output Truncated...>
Router#
```

Mapping Internal DSCP Values to Egress CoS Values

To configure the mapping of the DSCP value that PFC QoS uses internally on the PFC to the CoS value used for egress LAN port scheduling and congestion avoidance (see the “[Internal DSCP Values](#)” section on page 31-17 and the “[LAN Egress Port Features](#)” section on page 31-21), perform this task:

	Command	Purpose
Step 1	<pre>Router(config)# mls qos map dscp-cos dscp1 [dscp2 [dscp3 [dscp4 [dscp5 [dscp6 [dscp7 [dscp8]]]]]]] to cos_value Router(config)# no mls qos map dscp-cos</pre>	<p>Configures the internal DSCP to egress CoS map.</p> <p>Reverts to the default map.</p>
Step 2	<pre>Router(config)# end</pre>	Exits configuration mode.
Step 3	<pre>Router# show mls qos maps</pre>	Verifies the configuration.

When configuring the internal DSCP to egress CoS map, note the following:

- You can enter up to 8 DSCP values that PFC QoS maps to a CoS value.
- You can enter multiple commands to map additional DSCP values to a CoS value.
- You can enter a separate command for each CoS value.

This example shows how to configure internal DSCP values 0, 8, 16, 24, 32, 40, 48, and 54 to be mapped to egress CoS value 0:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls qos map dscp-cos 0 8 16 24 32 40 48 54 to 0
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos map | begin Dscp-cos map
Dscp-cos map: (dscp= d1d2)
  d1 : d2 0  1  2  3  4  5  6  7  8  9
-----
  0 :   00 00 00 00 00 00 00 00 00 00 01
  1 :   01 01 01 01 01 01 00 02 02 02 02
  2 :   02 02 02 02 00 03 03 03 03 03 03
  3 :   03 03 00 04 04 04 04 04 04 04 04
  4 :   00 05 05 05 05 05 05 05 00 06 06
  5 :   06 06 06 06 00 06 07 07 07 07 07
  6 :   07 07 07 07
<...Output Truncated...>
Router#
```



Note

In the **Dscp-cos** display, the CoS values are shown in the body of the matrix; the first digit of the DSCP value is in the column labeled **d1** and the second digit is in the top row. In the example shown, DSCP values 41 through 47 all map to CoS 05.

Configuring DSCP Markdown Values

To configure the mapping of DSCP markdown values used by policers (see the “[Policers](#)” section on page 31-19), perform this task:

	Command	Purpose
Step 1	<pre>Router(config)# mls qos map policed-dscp {normal-burst max-burst} dscp1 [dscp2 [dscp3 [dscp4 [dscp5 [dscp6 [dscp7 [dscp8]]]]]]] to markdown_dscp Router(config)# no mls qos map policed-dscp {normal-burst max-burst}</pre>	<p>Configures a DSCP markdown map.</p> <p>Reverts to the default map.</p>
Step 2	<pre>Router(config)# end</pre>	Exits configuration mode.
Step 3	<pre>Router# show mls qos maps</pre>	Verifies the configuration.

When configuring a DSCP markdown map, note the following:

- You can enter the **normal-burst** keyword to configure the markdown map used by the **exceed-action policed-dscp-transmit** keywords.
- You can enter the **max-burst** keyword to configure the markdown map used by the **violate-action policed-dscp-transmit** keywords.

**Note**

When you create a policer that does not use the **pir** keyword, and the *maximum_burst_bytes* parameter is equal to the *normal_burst_bytes* parameter (which occurs if you do not enter the *maximum_burst_bytes* parameter), the **exceed-action policed-dscp-transmit** keywords cause PFC QoS to mark traffic down as defined by the **policed-dscp max-burst** markdown map.

- To avoid out-of-sequence packets, configure the markdown maps so that conforming and nonconforming traffic uses the same queue.
- You can enter up to 8 DSCP values that map to a marked-down DSCP value.
- You can enter multiple commands to map additional DSCP values to a marked-down DSCP value.
- You can enter a separate command for each marked-down DSCP value.

**Note**

Configure marked-down DSCP values that map to CoS values consistent with the markdown penalty (see the “[Mapping Internal DSCP Values to Egress CoS Values](#)” section on page 31-67).

This example shows how to map DSCP 1 to marked-down DSCP value 0:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls qos map policed-dscp normal-burst 1 to 0
Router(config)# end
```

This example shows how to verify the configuration:

```
Router# show mls qos map
Normal Burst Policed-dscp map:                                     (dscp= d1d2)
d1 : d2 0  1  2  3  4  5  6  7  8  9
-----
 0 :    00 01 02 03 04 05 06 07 08 09
 1 :    10 11 12 13 14 15 16 17 18 19
 2 :    20 21 22 23 24 25 26 27 28 29
 3 :    30 31 32 33 34 35 36 37 38 39
 4 :    40 41 42 43 44 45 46 47 48 49
 5 :    50 51 52 53 54 55 56 57 58 59
 6 :    60 61 62 63

Maximum Burst Policed-dscp map:                                   (dscp= d1d2)
d1 : d2 0  1  2  3  4  5  6  7  8  9
-----
 0 :    00 01 02 03 04 05 06 07 08 09
 1 :    10 11 12 13 14 15 16 17 18 19
 2 :    20 21 22 23 24 25 26 27 28 29
 3 :    30 31 32 33 34 35 36 37 38 39
 4 :    40 41 42 43 44 45 46 47 48 49
 5 :    50 51 52 53 54 55 56 57 58 59
 6 :    60 61 62 63
<...Output Truncated...>
Router#
```

**Note**

In the **Policed-dscp** displays, the marked-down DSCP values are shown in the body of the matrix; the first digit of the original DSCP value is in the column labeled **d1** and the second digit is in the top row. In the example shown, DSCP 41 maps to DSCP 41.

Configuring PFC QoS Statistics Data Export



Note

Release 12.1(11b)E and later releases support PFC QoS statistics data export.

These sections describe how to configure PFC QoS statistics data export:

- [Enabling PFC QoS Statistics Data Export Globally, page 31-70](#)
- [Enabling PFC QoS Statistics Data Export for a Port, page 31-71](#)
- [Enabling PFC QoS Statistics Data Export for a Named Aggregate Policer, page 31-72](#)
- [Enabling PFC QoS Statistics Data Export for a Class Map, page 31-73](#)
- [Setting the PFC QoS Statistics Data Export Time Interval, page 31-74](#)
- [Configuring PFC QoS Statistics Data Export Destination Host and UDP Port, page 31-75](#)
- [Setting the PFC QoS Statistics Data Export Field Delimiter, page 31-77](#)

Enabling PFC QoS Statistics Data Export Globally

To enable PFC QoS statistics data export globally, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos statistics-export	Enables PFC QoS statistics data export globally.
	Router(config)# no mls qos statistics-export	Disables PFC QoS statistics data export globally.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos statistics-export info	Verifies the configuration.

This example shows how to enable PFC QoS statistics data export globally and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export
Router(config)# end
% Warning: Export destination not set.
% Use 'mls qos statistics-export destination' command to configure the export destination
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
-----
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured
Router#
```



Note

You must enable PFC QoS statistics data export globally for other PFC QoS statistics data export configuration to take effect.

Enabling PFC QoS Statistics Data Export for a Port

To enable PFC QoS statistics data export for a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# mls qos statistics-export	Enables PFC QoS statistics data export for the port.
	Router(config-if)# no mls qos statistics-export	Disables PFC QoS statistics data export for the port.
Step 3	Router(config)# end	Exits configuration mode.
Step 4	Router# show mls qos statistics-export info	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable PFC QoS statistics data export on FastEthernet port 5/24 and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 5/24
Router(config-if)# mls qos statistics-export
Router(config-if)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
-----
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured

QoS Statistics Data Export is enabled on following ports:
-----
FastEthernet5/24
Router#
```

When enabled on a port, PFC QoS statistics data export contains the following fields, separated by the delimiter character:

- Export type (“1” for a port)
- Slot/port
- Number of ingress packets
- Number of ingress bytes
- Number of egress packets
- Number of egress bytes
- Time stamp

Enabling PFC QoS Statistics Data Export for a Named Aggregate Policer

To enable PFC QoS statistics data export for a named aggregate policer, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos statistics-export aggregate-policer <i>aggregate_policer_name</i>	Enables PFC QoS statistics data export for a named aggregate policer.
	Router(config)# no mls qos statistics-export aggregate-policer <i>aggregate_policer_name</i>	Disables PFC QoS statistics data export for a named aggregate policer.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos statistics-export info	Verifies the configuration.

This example shows how to enable PFC QoS statistics data export for an aggregate policer named `aggr1M` and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export aggregate-policer aggr1M
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
-----
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured

QoS Statistics Data Export is enabled on following ports:
-----
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----
aggr1M
Router#
```

When enabled for a named aggregate policer, PFC QoS statistics data export contains the following fields, separated by the delimiter character:

- Export type (“3” for an aggregate policer)
- Aggregate policer name
- Direction (“in”)
- PFC or DFC slot number
- Number of in-profile packets
- Number of packets that exceed the CIR
- Number of packets that exceed the PIR
- Time stamp

Enabling PFC QoS Statistics Data Export for a Class Map

To enable PFC QoS statistics data export for a class map, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos statistics-export class-map <i>classmap_name</i>	Enables PFC QoS statistics data export for a class map.
	Router(config)# no mls qos statistics-export class-map <i>classmap_name</i>	Disables PFC QoS statistics data export for a class map.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos statistics-export info	Verifies the configuration.

This example shows how to enable PFC QoS statistics data export for a class map named class3 and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export class-map class3
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
-----
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured

QoS Statistics Data Export is enabled on following ports:
-----
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
-----
class3
Router#
```

When enabled for a class map, PFC QoS statistics data export contains the following fields, separated by the delimiter character:

- For data from a physical port:
 - Export type (“4” for a classmap and port)
 - Class map name
 - Direction (“in”)
 - Slot/port
 - Number of in-profile packets
 - Number of packets that exceed the CIR
 - Number of packets that exceed the PIR
 - Time stamp

- For data from a VLAN interface:
 - Export type (“5” for a class map and VLAN)
 - Class map name
 - Direction (“in”)
 - PFC or DFC slot number
 - VLAN ID
 - Number of in-profile packets
 - Number of packets that exceed the CIR
 - Number of packets that exceed the PIR
 - Time stamp
- For data from a port channel interface:
 - Export type (“6” for a class map and port channel)
 - Class map name
 - Direction (“in”)
 - PFC or DFC slot number
 - Port channel ID
 - Number of in-profile packets
 - Number of packets that exceed the CIR
 - Number of packets that exceed the PIR
 - Time stamp

Setting the PFC QoS Statistics Data Export Time Interval

To set the time interval for the PFC QoS statistics data export, perform this task:

	Command	Purpose
Step 1	<code>Router(config)# mls qos statistics-export interval interval_in_seconds</code>	Sets the time interval for the PFC QoS statistics data export. Note The interval needs to be short enough to avoid counter wraparound with the activity in your configuration, but because exporting PFC QoS statistic creates a significant load on the switch, be careful when decreasing the interval.
	<code>Router(config)# no mls qos statistics-export interval interval_in_seconds</code>	Reverts to the default time interval for the PFC QoS statistics data export.
Step 2	<code>Router(config)# end</code>	Exits configuration mode.
Step 3	<code>Router# show mls qos statistics-export info</code>	Verifies the configuration.

This example shows how to set the PFC QoS statistics data export interval and verify the configuration:

```
Router(config)# mls qos statistics-export interval 250
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
-----
Export Status : enabled
Export Interval : 250 seconds
Export Delimiter : |
Export Destination : Not configured

QoS Statistics Data Export is enabled on following ports:
-----
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
-----
class3
Router#
```

Configuring PFC QoS Statistics Data Export Destination Host and UDP Port

To configure the PFC QoS statistics data export destination host and UDP port number, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos statistics-export destination {host_name host_ip_address} {port port_number syslog [facility facility_name] [severity severity_value]}	Configures the PFC QoS statistics data export destination host and UDP port number.
	Router(config)# no mls qos statistics-export destination	Clears configured values.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos statistics-export info	Verifies the configuration.



Note

When the PFC QoS data export destination is a syslog server, the exported data is prefaced with a syslog header.

Table 31-3 lists the supported PFC QoS data export facility and severity parameter values.

Table 31-3 Supported PFC QoS Data Export Facility Parameter Values

Name	Definition	Name	Definition
kern	kernel messages	cron	cron/at subsystem
user	random user-level messages	local0	reserved for local use
mail	mail system	local1	reserved for local use
daemon	system daemons	local2	reserved for local use

Table 31-3 Supported PFC QoS Data Export Facility Parameter Values (continued)

Name	Definition	Name	Definition
auth	security/authentication messages	local3	reserved for local use
syslog	internal syslogd messages	local4	reserved for local use
lpr	line printer subsystem	local5	reserved for local use
news	netnews subsystem	local6	reserved for local use
uucp	uucp subsystem	local7	reserved for local use

Table 31-4 lists the supported PFC QoS data export severity parameter values.

Table 31-4 Supported PFC QoS Data Export Severity Parameter Values

Severity Parameter		
Name	Number	Definition
emerg	0	system is unusable
alert	1	action must be taken immediately
crit	2	critical conditions
err	3	error conditions
warning	4	warning conditions
notice	5	normal but significant condition
info	6	informational
debug	7	debug-level messages

This example shows how to configure 172.20.52.3 as the destination host and syslog as the UDP port number and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export destination 172.20.52.3 syslog
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
-----
Export Status : enabled
Export Interval : 250 seconds
Export Delimiter : |
Export Destination : 172.20.52.3, UDP port 514 Facility local6, Severity debug

QoS Statistics Data Export is enabled on following ports:
-----
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
-----
class3
```

Setting the PFC QoS Statistics Data Export Field Delimiter

To set the PFC QoS statistics data export field delimiter, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos statistics-export delimiter <i>delimiter_character</i>	Sets the PFC QoS statistics data export field delimiter.
	Router(config)# no mls qos statistics-export delimiter	Reverts to the default PFC QoS statistics data export field delimiter
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos statistics-export info	Verifies the configuration.

This example shows how to set the PFC QoS statistics data export field delimiter and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export delimiter ,
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
-----
Export Status : enabled
Export Interval : 250 seconds
Export Delimiter : ,
Export Destination : 172.20.52.3, UDP port 514 Facility local6, Severity debug

QoS Statistics Data Export is enabled on following ports:
-----
FastEthernet5/24

QoS Statistics Data Export is enabled on following shared aggregate policers:
-----
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
-----
class3
```




Configuring UDLD

This chapter describes how to configure the UniDirectional Link Detection (UDLD) protocol in Release 12.1(2)E and later releases on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How UDLD Works, page 32-1](#)
- [Default UDLD Configuration, page 32-3](#)
- [Configuring UDLD, page 32-3](#)

Understanding How UDLD Works

These sections describe how UDLD works:

- [UDLD Overview, page 32-1](#)
- [UDLD Aggressive Mode, page 32-2](#)

UDLD Overview

The UDLD protocol allows devices connected through fiber-optic or copper (for example, Category 5 cabling) Ethernet cables connected to LAN ports to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a unidirectional link is detected, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems, including spanning tree topology loops.

UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, then UDLD at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

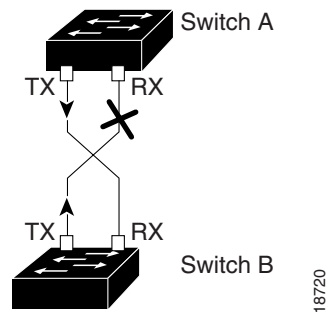
The Catalyst 6500 series switch periodically transmits UDLD packets to neighbor devices on LAN ports with UDLD enabled. If the packets are echoed back within a specific time frame and they are lacking a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links.

**Note**

By default, UDLD is locally disabled on copper LAN ports to avoid sending unnecessary control traffic on this type of media since it is often used for access ports.

Figure 32-1 shows an example of a unidirectional link condition. Switch B successfully receives traffic from Switch A on the port. However, Switch A does not receive traffic from Switch B on the same port. UDLD detects the problem and disables the port.

Figure 32-1 Unidirectional Link



UDLD Aggressive Mode

Release 12.1(3a)E and later releases support UDLD aggressive mode. UDLD aggressive mode is disabled by default. Configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. With UDLD aggressive mode enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD packets, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port is disabled.

To prevent spanning tree loops, nonaggressive UDLD with the default interval of 15 seconds is fast enough to shut down a unidirectional link before a blocking port transitions to the forwarding state (with default spanning tree parameters).

When you enable UDLD aggressive mode, you receive additional benefits in the following situations:

- One side of a link has a port stuck (both Tx and Rx)
- One side of a link remains up while the other side of the link has gone down

In these cases, UDLD aggressive mode disables one of the ports on the link, which prevents traffic from being discarded.

Default UDLD Configuration

Table 32-1 shows the default UDLD configuration.

Table 32-1 UDLD Default Configuration

Feature	Default Value
UDLD global enable state	Globally disabled
UDLD aggressive mode	Disabled
UDLD per-port enable state for fiber-optic media	Enabled on all Ethernet fiber-optic LAN ports
UDLD per-port enable state for twisted-pair (copper) media	Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports

Configuring UDLD

These sections describe how to configure UDLD:

- [Enabling UDLD Globally, page 32-3](#)
- [Enabling UDLD on Individual LAN Interfaces, page 32-4](#)
- [Disabling UDLD on Fiber-Optic LAN Interfaces, page 32-5](#)
- [Configuring the UDLD Probe Message Interval, page 32-5](#)
- [Resetting Disabled LAN Interfaces, page 32-6](#)



Note

With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Enabling UDLD Globally

To enable UDLD globally on all fiber-optic LAN ports, perform this task:

Command	Purpose
Router(config)# udld { enable aggressive }	Enables UDLD globally on fiber-optic LAN ports. Note This command only configures fiber-optic LAN ports. Individual LAN port configuration overrides the setting of this command.
Router(config)# no udld { enable aggressive }	Disables UDLD globally on fiber-optic LAN ports.

Enabling UDLD on Individual LAN Interfaces

With Release 12.1(13)E and later releases, to enable UDLD on individual LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# udld port [aggressive] Router(config-if)# no udld port [aggressive]	Enables UDLD on a specific LAN port. Enter the aggressive keyword to enable aggressive mode. On a fiber-optic LAN port, this command overrides the udld enable global configuration command setting. Disables UDLD on a nonfiber-optic LAN port. Note On fiber-optic LAN ports, the no udld port command reverts the LAN port configuration to the udld enable global configuration command setting.
Step 3	Router# show udld <i>type</i> ¹ <i>slot/number</i>	Verifies the configuration.
	1. <i>type</i> = ethernet, fastethernet, gigabitethernet, or tengigabitethernet	

With earlier releases, to enable UDLD on individual LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# udld enable [aggressive] Router(config-if)# no udld enable [aggressive]	Enables UDLD on a specific LAN port. Enter the aggressive keyword to enable aggressive mode. On a fiber-optic LAN port, this command overrides the udld enable global configuration command setting. Disables UDLD on a nonfiber-optic LAN port. Note On fiber-optic LAN ports, the no udld enable command reverts the LAN port configuration to the udld enable global configuration command setting.
Step 3	Router# show udld <i>type</i> ¹ <i>slot/number</i>	Verifies the configuration.
	1. <i>type</i> = ethernet, fastethernet, gigabitethernet, or tengigabitethernet	

Disabling UDLD on Fiber-Optic LAN Interfaces

With Release 12.1(13)E and later releases, to disable UDLD on individual fiber-optic LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# udld port disable Router(config-if)# no udld port disable	Disables UDLD on a fiber-optic LAN port. Reverts to the udld enable global configuration command setting. Note This command is only supported on fiber-optic LAN ports.
Step 3	Router# show udld <i>type</i> ¹ <i>slot/number</i>	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

With earlier releases, to disable UDLD on individual fiber-optic LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# udld disable Router(config-if)# no udld disable	Disables UDLD on a fiber-optic LAN port. Reverts to the udld enable global configuration command setting. Note This command is only supported on fiber-optic LAN ports.
Step 3	Router# show udld <i>type</i> ¹ <i>slot/number</i>	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

Configuring the UDLD Probe Message Interval

To configure the time between UDLD probe messages on ports that are in advertisement mode and are currently determined to be bidirectional, perform this task:

	Command	Purpose
Step 1	Router(config)# udld message time <i>interval</i> Router(config)# no udld message	Configures the time between UDLD probe messages on ports that are in advertisement mode and are currently determined to be bidirectional; valid values are from 7 to 90 seconds. Returns to the default value (60 seconds).
Step 2	Router# show udld <i>type</i> ¹ <i>slot/number</i>	Verifies the configuration.

Resetting Disabled LAN Interfaces

To reset all LAN ports that have been shut down by UDLD, perform this task:

Command	Purpose
Router# <code>udld reset</code>	Resets all LAN ports that have been shut down by UDLD.



Configuring NDE

This chapter describes how to configure NetFlow Data Export (NDE) on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication and the Release 12.1 publications at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

This chapter consists of these sections:

- [Understanding How NDE Works, page 33-1](#)
- [Default NDE Configuration, page 33-7](#)
- [Configuring NDE, page 33-8](#)



Note

- NDE does not support bridged traffic or Internetwork Packet Exchange (IPX) traffic.
 - NDE does not support IP multicast traffic. You can display NetFlow statistics for IP multicast traffic with the **show mls ip multicast** command.
-

Understanding How NDE Works

These sections describe how NDE works:

- [NDE Overview, page 33-2](#)
- [NDE from the MSFC, page 33-2](#)
- [NDE from the PFC, page 33-2](#)



Note

In this chapter, the term “PFC” refers to either a PFC2 or a PFC1, except when specifically differentiated, and the term “MSFC” refers to either an MSFC2 or an MSFC1, except when specifically differentiated.

NDE Overview

NDE makes routed-traffic statistics available for analysis by an external data collector. You can use NDE to analyze all IP unicast traffic that is Layer 3-switched on the PFC and all IP unicast traffic that is routed in software on the MSFC.

The Supervisor Engine 2 stores NetFlow statistics in the NetFlow table. The NDE configuration has no effect on Layer 3 switching in hardware by the PFC2. If the NetFlow table has more than 32K entries, there is an increased probability that there will be insufficient room to store statistics. On the Supervisor Engine 2, no statistics are available for flows that are switched when the NetFlow table is full.

On the Supervisor Engine 1, NetFlow statistics are derived from the MLS cache, which is used primarily for Layer 3 switching by the PFC. If you change the configuration to modify NDE, the new configuration applies to PFC Layer 3 switching. For more information about Layer 3 switching by the PFC on Supervisor Engine 1, see [Chapter 19, “Configuring IP Unicast Layer 3 Switching on Supervisor Engine 1.”](#) On the Supervisor Engine 1, when the MLS cache is full, the PFC sends flows to be switched by the MSFC, and NetFlow statistics are available from the MSFC for flows that are routed by the MSFC.

NDE from the MSFC

The NetFlow cache on the MSFC captures statistics for routed flows.

NDE on the Catalyst 6500 series switches can use NDE version 1, 5, or 6 to export the statistics captured on the MSFC for routed traffic. For more information, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_c/xcprt3/xcdnfov.htm

NDE from the PFC

These sections describe NDE from the PFC:

- [Flow Masks, page 33-2](#)
- [NDE Versions, page 33-3](#)
- [MLS Cache Entries, page 33-6](#)
- [Sampled NetFlow, page 33-6](#)

Flow Masks

The PFC uses a flow mask to create flow entries. The following flow masks exist:

- destination—The least-specific flow mask. The PFC maintains one entry for each destination IP address. All flows to a given destination IP address use this entry.
- destination-source—A more-specific flow mask. The PFC maintains one entry for each source and destination IP address pair. All flows between same source and destination IP addresses use this entry.
- destination-source-interface—A more-specific flow mask. Adds the source VLAN SNMP ifIndex to the information in the destination-source flow mask. The destination-source-interface flow mask is supported on Supervisor Engine 2 with Release 12.1(13)E and later releases.

- full—A more-specific flow mask. The PFC creates and maintains a separate cache entry for each IP flow. A full entry includes the source IP address, destination IP address, protocol, and protocol-specific Layer 4 port information.
- full-interface—The most-specific flow mask. Adds the source VLAN SNMP ifIndex to the information in the full flow mask. The full-interface flow mask is supported on Supervisor Engine 2 with Release 12.1(13)E and later releases.

The PFC uses only one flow mask for all Layer 3-switched traffic. If you change the flow mask configuration, the entire MLS cache is purged.

NDE Versions

NDE on the PFC supports the following NDE versions to export the statistics captured on the PFC for Layer 3-switched traffic:

- Supervisor Engine 1 and PFC—NDE version 7
- Supervisor Engine 2 and PFC2
 - NDE version 5 with Release 12.1(13)E and later releases
 - NDE version 7 with all releases

Depending on the current flow mask, some fields in the flow records might not have values. When the PFC exports cached entries, unsupported fields are filled with a zero (0).

The following tables list the supported NDE fields:

- [Table 33-1](#)—Version 5 header format
- [Table 33-2](#)—Version 5 flow record format
- [Table 33-3](#)—Version 7 header format
- [Table 33-4](#)—Version 7 flow record format

Table 33-1 NDE Version 5 Header Format

Bytes	Content	Description
0–1	version	Netflow export format version number
2–3	count	Number of flows exported in this packet (1–30)
4–7	SysUptime	Current time in milliseconds since router booted
8–11	unix_secs	Current seconds since 0000 UTC 1970
12–15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16–19	flow_sequence	Sequence counter of total flows seen
20–21	engine_type	Type of flow switching engine
21–23	engine_id	Slot number of the flow switching engine

Table 33-2 NDE Version 5 Flow Record Format

Bytes	Content	Description	Flow masks:				
			Destination	Destination Source	Destination Source Interface ¹	Full	Full Interface ¹
0–3	srcaddr	Source IP address		X	X	X	X
4–7	dstaddr	Destination IP address	X	X	X	X	X
8–11	nexthop	Next hop router's IP address	A ²	A	A	A	A
12–13	input	Ingress interface SNMP ifIndex			X		X
14–15	output	Egress interface SNMP ifIndex	A ²	A	A	A	A
16–19	dPkts	Packets in the flow	X	X	X	X	X
20–23	dOctets	Octets (bytes) in the flow	X	X	X	X	X
24–27	first	SysUptime at start of the flow	X	X	X	X	X
28–31	last	SysUptime at the time the last packet of the flow was received	X	X	X	X	X
32–33	srcport	Layer 4 source port number or equivalent				X	X
34–35	dstport	Layer 4 destination port number or equivalent				X	X
36	pad1	Unused (zero) byte					
37	tcp_flags	Cumulative OR of TCP flags					
38	prot	Layer 4 protocol (for example, 6=TCP, 17=UDP)				X	X
39	tos	IP type-of-service byte					
40–41	src_as	Autonomous system number of the source, either origin or peer		A	A	A	A
42–43	dst_as	Autonomous system number of the destination, either origin or peer	A	A	A	A	A
44–45	src_mask	Source address prefix mask bits					
46–47	dst_mask	Destination address prefix mask bits					
48	pad2	Pad 2					

1. Supported in Release 12.1(13)E and later releases.

2. With the destination flowmask, the "Next hop router's IP address" field and the "Output interface's SNMP ifIndex" field might not contain information that is accurate for all flows.

Table 33-3 NDE Version 7 Header Format

Bytes	Content	Description
0–1	version	Netflow export format version number
2–3	count	Number of flows exported in this packet (1–30)
4–7	SysUptime	Current time in milliseconds since router booted
8–11	unix_secs	Current seconds since 0000 UTC 1970
12–15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16–19	flow_sequence	Sequence counter of total flows seen
20–24	reserved	Unused (zero) bytes

Table 33-4 NDE Version 7 Flow Record Format

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the “Populating Additional NDE Fields” section on page 33-10)				
			Destination	Destination Source	Destination Source Interface ¹	Full	Full Interface ¹
0–3	srcaddr	Source IP address		X	X	X	X
4–7	dstaddr	Destination IP address	X	X	X	X	X
8–11	nexthop	Next hop router’s IP address	X ²	X	X	X	X
12–13	input	Ingress interface SNMP ifIndex			X		X
14–15	output	Egress interface SNMP ifIndex	X ²	X	X	X	X
16–19	dPkts	Packets in the flow	X	X	X	X	X
20–23	dOctets	Octets (bytes) in the flow	X	X	X	X	X
24–27	First	SysUptime at start of the flow	X	X	X	X	X
28–31	Last	SysUptime at the time the last packet of the flow was received	X	X	X	X	X
32–33	srcport	Layer 4 source port number or equivalent				X	X
34–35	dstport	Layer 4 destination port number or equivalent				X	X
36	flags	flow mask in use	X	X	X	X	X
37	tcp_flags	Cumulative OR of TCP flags					
38	prot	Layer 4 protocol (for example, 6=TCP, 17=UDP)				X	X
39	tos	IP type-of-service byte					
40–41	src_as	Autonomous system number of the source, either origin or peer		A	A	A	A

Table 33-4 NDE Version 7 Flow Record Format (continued)

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the “Populating Additional NDE Fields” section on page 33-10)				
			Destination	Destination Source	Destination Source Interface ¹	Full	Full Interface ¹
42–43	dst_as	Autonomous system number of the destination, either origin or peer	A	A	A	A	A
44	src_mask	Source address prefix mask bits					
45	dst_mask	Destination address prefix mask bits					
46–47	pad2	Pad 2					
48–51	MLS RP	IP address of MLS router	X	X	X	X	X

1. Supported in Release 12.1(13)E and later releases.

2. With the destination flowmask, the “Next hop router’s IP address” field and the “Output interface’s SNMP ifIndex” field might not contain information that is accurate for all flows.

MLS Cache Entries

NDE captures statistics for Layer 3-switched flows in the MLS cache on the PFC.

NDE maintains traffic statistics for each active flow in the MLS cache and increments the statistics when packets within each flow are switched. Periodically, NDE exports summarized traffic statistics for all expired flows, which the external data collector receives and processes.

Exported NetFlow data contains statistics for the flow entries in the MLS cache that have expired since the last export. Flow entries in the MLS cache expire and are flushed from the MLS cache when one of the following conditions occurs:

- The transport protocol indicates that the connection is completed.
- Traffic inactivity exceeds 15 seconds.

For flows that remain continuously active, flow entries in the MLS cache expire every 32 minutes to ensure periodic reporting of active flows.

NetFlow data export packets go to the external data collector either when the number of recently expired flows reaches a predetermined maximum, or every second, whichever occurs first.

By default, all expired flows are exported unless filtered. With a filter configured, NDE only exports expired and purged flows that match the filter criteria. NDE flow filters are stored in NVRAM and are not cleared when NDE is disabled. See the “Configuring NDE Flow Filters” section on page 33-15 for NDE filter configuration procedures.

Sampled NetFlow

Sampled NetFlow exports data for a subset of the Layer 3-switched IP packets instead of for all packets in a flow. Sampled NetFlow substantially decreases the Supervisor Engine 2 CPU utilization. Release 12.1(13)E and later releases support sampled NetFlow on the Supervisor Engine 2.

With the full-interface or destination-source-interface flow masks, you can enable or disable sampled NetFlow on each LAN port. With all other flow masks, sampled Netflow is enabled or disabled globally.

You can configure sampled NetFlow to use time-based sampling or packet-based sampling.

Table 33-5 lists the time-based sampling rates and export intervals.

Table 33-5 Time-Based Sampling Rates, Sampling Times, and Export Intervals

Sampling Rate	Sampling Time (Milliseconds)	Export Interval (Milliseconds)
1 in 64	64	4096
1 in 128	32	4096
1 in 256	16	4096
1 in 512	8	4096
1 in 1024	4	4096
1 in 2048	4	8192
1 in 4096	4	16384
1 in 8192	4	32768

As examples, if you configure 64 as the rate, then every 4096 milliseconds the sampled NetFlow feature uses traffic from the first 64 milliseconds of a flow; if the rate is 2048, then every 8192 milliseconds, the sampled NetFlow feature uses traffic from the first 4 milliseconds of a flow. With time-based sampled NetFlow, the export interval is not configurable.

Packet-based sampled NetFlow uses this formula to sample a flow: the number of times sampled is approximately the length divided by the rate (*packets_in_flow/sampling_rate*). For example, if the flow is 32,768 packets long and the sampling rate is 1024, the flow is sampled approximately 32 times ($32,768/1,024$). With packet-based sampled NetFlow, the export interval is configurable.

Default NDE Configuration

Table 33-6 shows the default NDE configuration.

Table 33-6 Default NetFlow Data Export Configuration

Feature	Default Value
NDE	Disabled
NDE source addresses	None
NDE data collector address and UDP port	None
NDE filters	None
Sampled NetFlow	Disabled
Populating additional NDE fields	Disabled

Configuring NDE

These sections describe how to configure NDE:

- [Configuring NDE on the PFC, page 33-8](#)
- [Configuring NDE on the MSFC, page 33-13](#)
- [Displaying the NDE Address and Port Configuration, page 33-14](#)
- [Configuring NDE Flow Filters, page 33-15](#)
- [Displaying the NDE Configuration, page 33-17](#)



Note

- You must enable NetFlow on the MSFC Layer 3 interfaces to support NDE on the PFC and on the MSFC.
- You must configure NDE on the MSFC to support NDE on the PFC.
- With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Configuring NDE on the PFC

These sections describe how to configure NDE on the PFC:

- [Enabling NDE From the PFC, page 33-8](#)
- [Setting the Minimum IP MLS Flow Mask, page 33-9](#)
- [Populating Additional NDE Fields, page 33-10](#)
- [Configuring the MLS Aging Time, page 33-10](#)
- [Configuring Sampled NetFlow, page 33-11](#)

Enabling NDE From the PFC

NDE from the PFC uses the source configured for the MSFC. To enable NDE from the PFC, perform this task:

Command	Purpose
Router(config)# mls nde sender [version {5 7}]	Enables NDE from the PFC. Note NDE version 5 is supported on Supervisor Engine 2 with Release 12.1(13)E and later releases.
Router(config)# no mls nde sender	Disables NDE from the PFC.



Note

With Supervisor Engine 1 and PFC, if NDE is enabled and you disable Multilayer Switching (MLS), you lose the statistics for existing cache entries. They are not exported when MLS shuts down.

This example shows how to enable NDE from the PFC:

```
Router(config)# mls nde sender
```

Setting the Minimum IP MLS Flow Mask

You can set the minimum granularity of the flow mask for the MLS cache on the PFC. The actual flow mask used will have at least the granularity specified by this command. For information on how the different flow masks work, see the “Flow Masks” section on page 33-2.

If you configure TCP intercept, IOS Server Load Balancing (ISLB), Context-Based Access Control (CBAC), reflexive ACLs, or Web Cache Communication Protocol (WCCP), the flow mask changes to full.



Caution

Changing the flow mask purges all existing shortcuts in the MLS cache, which on a Supervisor Engine 1 affects the number of active shortcuts. Be careful when using this command on a Supervisor Engine 1. With a Supervisor Engine 2, NDE configuration has no effect on Layer 3 switching in hardware by the PFC2.

To set the minimum IP MLS flow mask, perform this task:

Command	Purpose
Router(config)# mls flow ip { destination destination-source interface-destination-source full interface-full }	Sets the minimum IP MLS flow mask for the protocol.
Router(config)# no mls flow ip	Reverts to the default IP MLS flow mask.



Note

Release 12.1(13)E and later releases support the **interface-destination-source** and **interface-full** keywords.

This example shows how to set the minimum IP MLS flow mask:

```
Router(config)# mls flow ip destination
```

To display the IP MLS flow mask configuration, perform this task:

Command	Purpose
Router# show mls netflow flowmask	With Release 12.1(8a)E and later releases, displays the flow mask configuration.
Router# show mls flowmask	With releases earlier than Release 12.1(8a)E, displays the flow mask configuration.

This example shows how to display the MLS flow mask configuration:

```
Router# show mls netflow flowmask
current ip flowmask for unicast: destination address
current ipx flowmask for unicast: destination address
Router#
```

Populating Additional NDE Fields

With Release 12.1(13)E and later releases, you can configure NDE to populate the following additional fields in the NDE packets:

- IP address of the next hop router
- Egress interface SNMP ifIndex
- Source autonomous system number
- Destination autonomous system number

Not all of the additional fields are populated with all flow masks. See the “[NDE Versions](#)” section on [page 33-3](#) for additional information.

To populate the additional fields in NDE packets, perform this task:

Command	Purpose
Router(config)# mls nde interface	Populates additional fields in NDE packets.
Router(config)# no mls nde interface	Disables population of the additional fields.

This example shows how to populate the additional fields in NDE packets:

```
Router(config)# mls nde interface
```

Configuring the MLS Aging Time

The MLS aging time applies to all MLS cache entries. The aging-time value is applied directly to destination mode aging. The MLS aging time value is divided by two to obtain the source-to-destination mode aging time and divided by eight to obtain the full-flow aging time. The default MLS aging time value is 256 seconds.

You can configure the normal aging time in the range of 32 to 4092 seconds in 8-second increments. Any aging-time value that is not a multiple of 8 seconds is adjusted to the closest multiple of 8 seconds. For example, a value of 65 is adjusted to 64 and a value of 127 is adjusted to 128.

Other events might cause MLS entries to be purged, such as routing changes or a change in link state (PFC link is down).



Note

If the number of MLS entries exceeds 32K, only adjacency statistics might be available for some flows.

To keep the MLS cache size below 32K entries, enable the following parameters when using the **mls aging** command:

- **normal**—Configures the wait before aging out and deleting shortcut entries in the Layer 3 table.
- **fast aging**—Configures an efficient process to age out entries created for flows that only switch a few packets and then are never used again. The **fast aging** parameter uses the **time** keyword value to check if at least the **threshold** keyword value of packets have been switched for each flow. If a flow has not switched the threshold number of packets during the time interval, then the entry in the Layer 3 table is aged out.
- **long**—Configures entries for deletion that have been up for the specified value even if the Layer 3 entry is in use. Long aging is used to prevent counter wraparound, which can cause inaccurate statistics.

A typical cache entry that is removed is the entry for flows to and from a Domain Name Server (DNS) or TFTP server. This entry might not be used again after it is created. The PFC saves space in the MLS cache for other data when it detects and ages out these entries.

If you need to enable MLS fast aging time, initially set the value to 128 seconds. If the size of the MLS cache continues to grow over 32K entries, decrease the setting until the cache size stays below 32K. If the cache continues to grow over 32K entries, decrease the normal MLS aging time.

To configure the MLS aging time, perform this task:

Command	Purpose
Router(config)# mls aging { fast [threshold {1-128} time {1-128}] long 64-900 normal 32-4092}	Configures the MLS aging time for an MLS cache entry.
Router(config)# no mls aging { fast long normal }	Reverts to the default MLS aging time.

This example displays how to configure the MLS aging time:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls aging fast threshold 64 time 30
```

To display the MLS aging-time configuration, perform this task:

Command	Purpose
Router# show mls aging	Displays the MLS aging-time configuration.

This example shows how to display the MLS aging-time configuration:

```
Router# show mls aging
          enable timeout  packet threshold
          -----
normal aging false      300      N/A
fast aging   false      32       100
long aging   false      900      N/A

Router#
```

Configuring Sampled NetFlow

These sections describe how to configure sampled NetFlow on the PFC:

- [Configuring Sampled NetFlow Globally, page 33-12](#)
- [Configuring Sampled NetFlow on a Layer 3 Interface, page 33-12](#)



Note

- Release 12.1(13)E and later releases support sampled NetFlow on the PFC.
- NDE on the MSFC does not support sampled NetFlow.
- With the full-interface or destination-source-interface flow masks, you can enable or disable sampled NetFlow on individual Layer 3 interfaces. With all other flow masks, sampled NetFlow is enabled or disabled globally.

Configuring Sampled NetFlow Globally

To configure sampled NetFlow globally, perform this task:

	Command	Purpose
Step 1	Router(config)# mls sampling { time-based <i>rate</i> packet-based <i>rate</i> [<i>interval</i>]}	Enables sampled NetFlow and configures the rate. For packet-based sampling, optionally configures the export interval.
	Router(config)# no mls sampling	Clears the sampled NetFlow configuration.
Step 2	Router(config)# end	Exits configuration mode.

When you configure sampled NetFlow globally, note the following:

- The valid values for *rate* are 64, 128, 256, 512, 1024, 2048, 4096, and 8192.
- The valid values for the packet-based export *interval* are from 4000 through 16,000.

See the “[Sampled NetFlow](#)” section on page 33-6 for more information.

Configuring Sampled NetFlow on a Layer 3 Interface



Note

- With the full-interface or destination-source-interface flow masks, you can enable or disable sampled NetFlow on individual Layer 3 interfaces. With all other flow masks, sampled NetFlow is enabled or disabled globally.
- The Layer 3 interface must be configured with an IP address.

To configure sampled NetFlow on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface { vlan <i>vlan_ID</i> <i>type</i> ¹ <i>slot/port</i> }	Selects an interface to configure.
Step 2	Router(config-if)# mls netflow sampling	Enables sampled NetFlow on the interface.
	Router(config-if)# no mls netflow sampling	Disables sampled NetFlow on the interface.
Step 3	Router(config)# end	Exits configuration mode.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable sampled NetFlow on Fast Ethernet port 5/12:

```
Router# configure terminal
Router(config)# interface fastethernet 5/12
Router(config-if)# mls netflow sampling
Router(config)# end
Router#
```

Configuring NDE on the MSFC

This section supplements the NetFlow procedures at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_r/index.htm

These sections describe how to configure NDE on the MSFC:

- [Enabling NetFlow, page 33-13](#)
- [Configuring the MSFC NDE Source Layer 3 Interface, page 33-13](#)
- [Configuring the NDE Destination, page 33-14](#)



Note

- You must enable NetFlow on the MSFC Layer 3 interfaces to support NDE on the PFC and NDE on the MSFC.
- You must enable NDE on the MSFC to support NDE on the PFC.

Enabling NetFlow

To enable NetFlow, perform this task for each Layer 3 interface from which you want NDE:

	Command	Purpose
Step 1	Router(config)# interface { vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>port_channel_number</i> }	Selects an interface to configure.
Step 2	Router(config-if)# ip route-cache flow	Enables NetFlow.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

Configuring the MSFC NDE Source Layer 3 Interface

To configure the Layer 3 interface used as the source of the NDE packets containing statistics from the MSFC, perform this task:

Command	Purpose
Router(config)# ip flow-export source {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> } { loopback <i>number</i> }}	Configures the interface used as the source of the NDE packets containing statistics from the MSFC: <ul style="list-style-type: none"> • Select an interface configured with an IP address. • You can use a loopback interface.
Router(config)# no ip flow-export source	Clears the NDE source interface configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure a loopback interface as the NDE flow source:

```
Router(config)# ip flow-export source loopback 0
Router(config)#
```

Configuring the NDE Destination

To configure the destination IP address and UDP port to receive the NDE statistics, perform this task:

Command	Purpose
Router(config)# ip flow-export destination <i>ip_address</i> <i>udp_port_number</i>	Configures the NDE destination IP address and UDP port.
Router(config)# no ip flow-export destination	Clears the NDE destination configuration.

This example shows how to configure the NDE flow destination IP address and UDP port:

```
Router(config)# ip flow-export destination 172.20.52.37 200
```



Note

The destination address and UDP port number are saved in NVRAM and are preserved if NDE is disabled and reenabled or if the switch is power cycled. If you are using the NetFlow FlowCollector application for data collection, verify that the UDP port number you configure is the same port number shown in the FlowCollector's `nfconfig.file`. This file is located at `/opt/csconfc/config/nfconfig.file` in the FlowCollector application.

Displaying the NDE Address and Port Configuration

To display the NDE address and port configuration, perform these tasks:

Command	Purpose
Router# show mls nde	Displays the NDE export flow IP address and UDP port configuration.
Router# show ip flow export	Displays the NDE export flow IP address, UDP port, and the NDE source interface configuration.

This example shows how to display the NDE export flow source IP address and UDP port configuration:

```
Router# show mls nde
Netflow Data Export enabled
Netflow Data Export configured for port 0 on Host 0.0.0.0
Source address: 172.20.52.3, port: 8
Version: 0
Include Filter is:
  destination: ip address 0.0.0.0, mask 0.0.0.0, port 35
  source: ip address 0.0.0.0, mask 0.0.0.0, port 0
Exclude Filter is:
  destination: ip address 2.2.2.2, mask 255.255.255.0, port 23
  source: ip address 0.0.0.0, mask 0.0.0.0, port 0
Total Netflow Data Export Packets are:
  0 packets, 0 no packets, 0 records
Router#
```

This example shows how to display the NDE export flow IP address, UDP port, and the NDE source interface configuration:

```
Router# show ip flow export
Flow export is enabled
Exporting flows to 172.20.52.37 (200)
Exporting using source interface FastEthernet5/8
```



```

Version 1 flow records
0 flows exported in 0 udp datagrams
0 flows failed due to lack of export packet
0 export packets were sent up to process level
0 export packets were dropped due to no fib
0 export packets were dropped due to adjacency issues
Router#

```

Configuring NDE Flow Filters

These sections describe NDE flow filters:

- [NDE Flow Filter Overview, page 33-15](#)
- [Configuring a Port Flow Filter, page 33-15](#)
- [Configuring a Host and Port Filter, page 33-16](#)
- [Configuring a Host Flow Filter, page 33-16](#)
- [Configuring a Protocol Flow Filter, page 33-16](#)
- [Clearing an NDE Flow Filter, page 33-17](#)

NDE Flow Filter Overview

By default, all expired flows are exported until you configure a filter. After you configure a filter, only expired and purged flows matching the specified filter criteria are exported. Filter values are stored in NVRAM and are not cleared when NDE is disabled.

To display the configuration of the NDE flow filters you configure, use the **show mls nde** command described in the “[Displaying the NDE Configuration](#)” section on page 33-17.

Configuring a Port Flow Filter

To configure a destination or source port flow filter, perform this task:

Command	Purpose
Router(config)# mls nde flow { exclude include } { dest-port <i>number</i> src-port <i>number</i> }	Configures a port flow filter for an NDE flow.
Router(config)# no mls nde flow { exclude include }	Clears the port flow filter configuration.

This example shows how to configure a port flow filter so that only expired flows to destination port 23 are exported (assuming the flow mask is set to ip-flow):

```

Router(config)# mls nde flow include dest-port 35
Router(config)#

```

Configuring a Host and Port Filter

To configure a host and TCP/UDP port flow filter, perform this task:

Command	Purpose
Router(config)# mls nde flow { exclude include } { destination <i>ip_address mask</i> source <i>ip_address mask</i> { dest-port <i>number</i> src-port <i>number</i> }}	Configures a host and port flow filter for an NDE flow.
Router(config)# no mls nde flow { exclude include }	Clears the port flow filter configuration.

This example shows how to configure a source host and destination TCP/UDP port flow filter so that only expired flows from host 171.69.194.140 to destination port 23 are exported (assuming the flow mask is set to ip-flow):

```
Router(config)# mls nde flow exclude destination 2.2.2.2 255.255.255.0 dest-port 23
```

Configuring a Host Flow Filter

To configure a destination or source host flow filter, perform this task:

Command	Purpose
Router(config)# mls nde flow { exclude include } { destination <i>ip_address mask</i> source <i>ip_address mask</i> protocol { tcp { dest-port <i>number</i> src-port <i>number</i> } udp { dest-port <i>number</i> src-port <i>number</i> }}	Configures a host flow filter for an NDE flow.
Router(config)# no mls nde flow { exclude include }	Clears port filter configuration.

This example shows how to configure a host flow filter to include and export only destinations to host 172.20.52.37:

```
Router(config)# mls nde flow include destination 172.20.52.37 255.255.255.224
Router(config)#
```

Configuring a Protocol Flow Filter

To configure a protocol flow filter, perform this task:

Command	Purpose
Router(config)# mls nde flow { exclude include } protocol { tcp { dest-port <i>number</i> src-port <i>number</i> } udp { dest-port <i>number</i> src-port <i>number</i> }}	Configures a protocol flow filter for an NDE flow.
Router(config)# no mls nde flow { exclude include }	Clears port filter configuration.

This example shows how to configure a TCP protocol flow filter so that only expired flows from destination port 35 are exported:

```
Router(config)# mls nde flow include protocol tcp dest-port 35
Router(config)#
```

Clearing an NDE Flow Filter

To clear the NDE flow filter and reset the filter to the default (all flows exported), perform this task:

Command	Purpose
Router# clear mls nde flow {all exclude include}	Clears the NDE flow filter.

This example shows how to clear the NDE flow filter so that all flows are exported:

```
Router# clear mls nde flow all
Router#
```

To display the status of the NDE flow filters, use the **show mls nde** command described in the [“Displaying the NDE Configuration”](#) section on page 33-17.

Displaying the NDE Configuration

To display the NDE configuration, perform this task:

Command	Purpose
Router# show mls nde	Displays the NDE configuration.

This example shows how to display the NDE configuration:

```
Router# show mls nde
Netflow Data Export enabled
Netflow Data Export configured for port 0 on Host 0.0.0.0
Source address: 172.20.52.3, port: 8
Version: 0
Include Filter is:
  destination: ip address 0.0.0.0, mask 0.0.0.0, port 35
  source: ip address 0.0.0.0, mask 0.0.0.0, port 0
Exclude Filter is:
  destination: ip address 2.2.2.2, mask 255.255.255.0, port 23
  source: ip address 0.0.0.0, mask 0.0.0.0, port 0
Total Netflow Data Export Packets are:
  0 packets, 0 no packets, 0 records
Router#
```




Configuring Local SPAN and RSPAN

This chapter describes how to configure local Switched Port Analyzer (SPAN) and remote SPAN (RSPAN) on the Catalyst 6500 series switches. The Catalyst 6500 series switches support RSPAN with Release 12.1(13)E and later releases.

This chapter consists of these sections:

- [Understanding How Local SPAN and RSPAN Work, page 34-1](#)
- [Local SPAN and RSPAN Configuration Guidelines and Restrictions, page 34-5](#)
- [Configuring Local SPAN and RSPAN, page 34-8](#)



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

Understanding How Local SPAN and RSPAN Work

These sections describe how local SPAN and RSPAN work:

- [Local SPAN and RSPAN Overview, page 34-1](#)
- [Local SPAN and RSPAN Sessions, page 34-3](#)
- [Monitored Traffic, page 34-4](#)
- [SPAN Sources, page 34-4](#)
- [Destination Ports, page 34-5](#)

Local SPAN and RSPAN Overview

Local SPAN and RSPAN both select network traffic to send to a network analyzer such as a SwitchProbe device or other Remote Monitoring (RMON) probe. SPAN does not affect the switching of network traffic on source ports or VLANs. SPAN sends a copy of the packets received or transmitted by the source ports and VLANs to the destination port. You must dedicate the destination port for SPAN use.

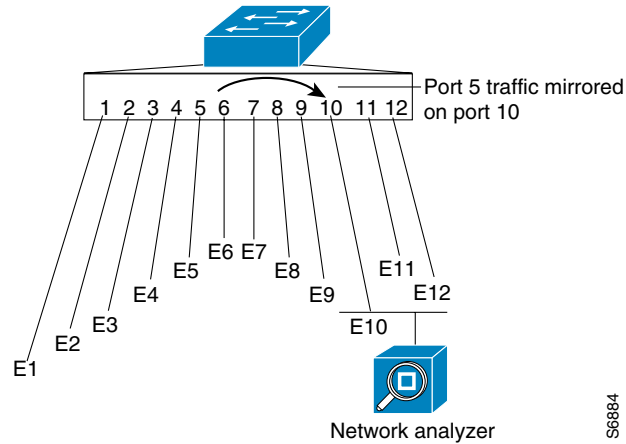
These sections provide an overview of local SPAN and RSPAN:

- [Local SPAN Overview, page 34-2](#)
- [RSPAN Overview, page 34-3](#)

Local SPAN Overview

Local SPAN supports source ports, source VLANs, and destination ports on the same Catalyst 6500 series switch. Local SPAN copies traffic from one or more source ports in any VLAN or from one or more VLANs to a destination port for analysis (see [Figure 34-1](#)). For example, as shown in [Figure 34-1](#), all traffic on Ethernet port 5 (the source port) is copied to Ethernet port 10. A network analyzer on Ethernet port 10 receives all network traffic from Ethernet port 5 without being physically attached to Ethernet port 5.

Figure 34-1 Example SPAN Configuration



S6884

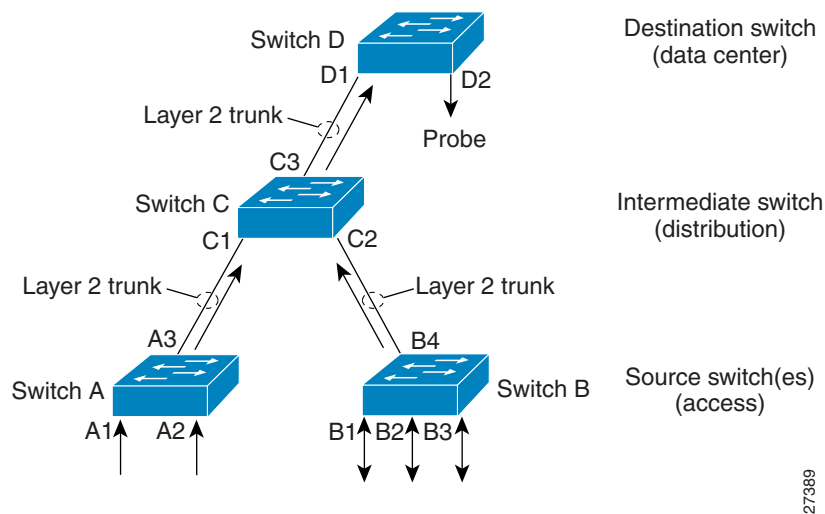
RSPAN Overview

RSPAN supports source ports, source VLANs, and destination ports on different switches, which provides remote monitoring of multiple switches across your network (see [Figure 34-2](#)). The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches.

The RSPAN source ports can be trunks carrying the RSPAN VLAN. Local SPAN and RSPAN do not monitor the RSPAN traffic in the RSPAN VLAN seen on a source trunk.

The RSPAN traffic from the source ports or source VLANs is switched to the RSPAN VLAN and then forwarded to destination ports, which are in the RSPAN VLAN. The sources (ports or VLANs) in an RSPAN session can be different on different source switches but must be the same for all sources on each RSPAN source switch. Each RSPAN source switch must have either ports or VLANs as RSPAN sources.

Figure 34-2 RSPAN Configuration



Local SPAN and RSPAN Sessions

SPAN sessions (local or remote) allow you to monitor traffic on one or more ports, or one or more VLANs, and send the monitored traffic to one or more destination ports.

A local SPAN session is an association of a set of source ports and source VLANs with one or more destination ports. You configure a local SPAN session on a single network device. Local SPAN does not have separate source and destination sessions.

RSPAN consists of an RSPAN source session, an RSPAN VLAN, and an RSPAN destination session. You separately configure RSPAN source sessions and destination sessions on different network devices. To configure an RSPAN source session on one network device, you associate a set of source ports and VLANs with an RSPAN VLAN. To configure an RSPAN destination session on another device, you associate the destination port with the RSPAN VLAN.

Monitored Traffic

These sections describe the traffic that SPAN (local or remote) can monitor:

- [Monitored Traffic Direction, page 34-4](#)
- [Monitored Traffic Type, page 34-4](#)
- [Duplicate Traffic, page 34-4](#)

Monitored Traffic Direction

You can configure SPAN sessions to monitor ingress network traffic (called ingress SPAN), or to monitor egress network traffic (called egress SPAN), or to monitor traffic flowing in both directions.

Ingress SPAN copies network traffic received by the source ports and VLANs for analysis at the destination port. Egress SPAN copies network traffic transmitted from the source ports and VLANs. When you enter the **both** keyword, SPAN copies the network traffic received and transmitted by the source ports and VLANs to the destination port.

Monitored Traffic Type

By default, local SPAN monitors all network traffic, including multicast and bridge protocol data unit (BPDU) frames. RSPAN does not support BPDU monitoring.

Duplicate Traffic

In some configurations, SPAN sends multiple copies of the same source traffic to the destination port. For example, in a configuration with a bidirectional SPAN session (both ingress and egress) for two SPAN sources, called s1 and s2, to a SPAN destination port, called d1, if a packet enters the switch through s1 and is sent for egress from the switch to s2, ingress SPAN at s1 sends a copy of the packet to SPAN destination d1 and egress SPAN at s2 sends a copy of the packet to SPAN destination d1. If the packet was Layer 2 switched from s1 to s2, both SPAN packets would be the same. If the packet was Layer 3 switched from s1 to s2, the Layer-3 rewrite would alter the source and destination Layer 2 addresses, in which case the SPAN packets would be different.

SPAN Sources

These sections describe local SPAN and RSPAN sources:

- [Source Ports, page 34-4](#)
- [Source VLANs, page 34-5](#)

Source Ports

A source port is a port monitored for network traffic analysis. You can configure both switched and routed ports as SPAN source ports. SPAN can monitor one or more source ports in a single SPAN session. You can configure source ports in any VLAN. Trunk ports can be configured as source ports and mixed with nontrunk source ports, but SPAN does not copy the encapsulation from a source trunk port.

Source VLANs

A source VLAN is a VLAN monitored for network traffic analysis. VLAN-based SPAN (VSPAN) uses a VLAN as the SPAN source. All the ports in the source VLANs become source ports.

Destination Ports

A destination port is a Layer 2 or Layer 3 LAN port to which SPAN sends traffic for analysis.

When you configure a port as a SPAN destination port, it can no longer receive any traffic. When you configure a port as a SPAN destination port, the port is dedicated for use only by the SPAN feature. A SPAN destination port does not forward any traffic except that required for the SPAN session.

With Release 12.1(13)E and later releases, you can configure trunk ports as destination ports, which allows destination trunk ports to transmit encapsulated traffic. With earlier releases, trunk ports stop trunking when you configure them as a destination port.

Local SPAN and RSPAN Configuration Guidelines and Restrictions

These sections describe local SPAN and RSPAN configuration guidelines and restrictions:

- [Local SPAN and RSPAN Session Limits, page 34-5](#)
- [Local SPAN and RSPAN Source and Destination Limits, page 34-6](#)
- [Local SPAN and RSPAN Guidelines and Restrictions, page 34-6](#)
- [VSPAN Guidelines and Restrictions, page 34-7](#)
- [RSPAN Guidelines and Restrictions, page 34-7](#)

Local SPAN and RSPAN Session Limits

These are the local SPAN and RSPAN session limits:

Total Sessions per Switch	Local SPAN Sessions	RSPAN Source Sessions	RSPAN Destination Sessions
66	2 (ingress or egress or both)	0	64
	1 ingress	1 (ingress or egress or both)	
	1 or 2 egress	0	

Local SPAN and RSPAN Source and Destination Limits

These are the local SPAN and RSPAN source and destination limits:

Sources and Destinations	Local SPAN Sessions	RSPAN Source Sessions	RSPAN Destination Sessions
Egress sources	1 (0 with a remote SPAN source session configured)	1 (0 with a local SPAN egress source session configured)	1 RSPAN VLAN
Ingress sources	64	64	
Destinations per session	64	1 RSPAN VLAN	64

Local SPAN and RSPAN Guidelines and Restrictions

These guidelines and restrictions apply to both local SPAN and RSPAN:

- Release 12.1(13)E and later releases support RSPAN.
- In releases earlier than 12.1(20)E, ports on the WS-X6548-GE-TX and WS-X6548V-GE-TX switching modules cannot be ingress SPAN sources when the switch is operating in truncated mode.
- You need a network analyzer to monitor destination ports.
- You can configure both Layer 2 LAN ports (LAN ports configured with the **switchport** command) and Layer 3 LAN ports (LAN ports not configured with the **switchport** command) as sources or destinations.
- With Release 12.1(13)E and later releases, you can configure destination ports as trunks to capture tagged traffic. With earlier releases, if you configure a trunk port as a destination port, SPAN suspends trunking on the port.
- A port specified as a destination port in one SPAN session cannot be a destination port for another SPAN session.
- A port configured as a destination port cannot be configured as a source port.
- A port channel interface (an EtherChannel) can be a source.
 - With Release 12.1(13)E and later releases, you cannot configure active member ports of an EtherChannel as source ports. Inactive member ports of an EtherChannel can be configured as sources but they are put into the suspended state and carry no traffic.
 - With releases earlier than 12.1(13)E, if you configure a member port of an EtherChannel as a SPAN source port, it is put into the suspended state and carries no traffic.
- A port channel interface (an EtherChannel) cannot be a destination.
 - With Release 12.1(13)E and later releases, you cannot configure active member ports of an EtherChannel as destination ports. Inactive member ports of an EtherChannel can be configured as destinations but they are put into the suspended state and carry no traffic.
 - With releases earlier than 12.1(13)E, if you configure a member port of an EtherChannel as a SPAN destination port, it is put into the suspended state and carries no traffic.
- You cannot mix individual source ports and source VLANs within a single session.
- If you specify multiple ingress source ports, the ports can belong to different VLANs.
- You cannot mix source VLANs and filter VLANs within a session. You can have source VLANs or filter VLANs, but not both at the same time.

- When enabled, local SPAN or RSPAN uses any previously entered configuration.
- When you specify sources and do not specify a traffic direction (ingress, egress, or both), “both” is used by default.
- You cannot configure destination ports to receive ingress traffic.
- Destination ports never participate in any spanning tree instance. Local SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the destination port are from the source port. RSPAN does not support BPDU monitoring.
- All packets sent through the switch for transmission from a port configured as an egress source are copied to the destination port, including packets that do not exit the switch through the port because STP has put the port into the blocking state, or on a trunk port because STP has put the VLAN into the blocking state on the trunk port.

VSPAN Guidelines and Restrictions

These are VSPAN guidelines and restrictions:

- For VSPAN sessions with both ingress and egress configured, two packets are forwarded from the destination port if the packets get switched on the same VLAN (one as ingress traffic from the ingress port and one as egress traffic from the egress port).
- VSPAN only monitors traffic that leaves or enters Layer 2 ports in the VLAN.
 - If you configure a VLAN as an ingress source and traffic gets routed into the monitored VLAN, the routed traffic is not monitored because it never appears as ingress traffic entering a Layer 2 port in the VLAN.
 - If you configure a VLAN as an egress source and traffic gets routed out of the monitored VLAN, the routed traffic is not monitored because it never appears as egress traffic leaving a Layer 2 port in the VLAN.

RSPAN Guidelines and Restrictions

These are RSPAN guidelines and restrictions:

- Any network device that supports RSPAN VLANs can be an RSPAN intermediate device.
- Networks impose no limit on the number of RSPAN VLANs that the networks carry.
- Intermediate switches might impose limits on the number of RSPAN VLANs that they can support.
- You must configure the RSPAN VLANs in all source, intermediate, and destination network devices. If enabled, the VLAN Trunking Protocol (VTP) can propagate configuration of VLANs numbered 1 through 1024 as RSPAN VLANs. You must manually configure VLANs numbered higher than 1024 as RSPAN VLANs on all source, intermediate, and destination network devices.
- If you enable VTP and VTP pruning, RSPAN traffic is pruned in the trunks to prevent the unwanted flooding of RSPAN traffic across the network.
- RSPAN VLANs can be used only for RSPAN traffic.
- Do not configure a VLAN used to carry management traffic as an RSPAN VLAN.
- Do not assign access ports to RSPAN VLANs. RSPAN puts access ports in an RSPAN VLAN into the suspended state.
- Do not configure any ports in an RSPAN VLAN except those selected to carry RSPAN traffic.

- MAC address learning is disabled on the RSPAN VLAN.
- You can use an output access control list (ACL) on the RSPAN VLAN in the RSPAN source switch to filter the traffic sent to an RSPAN destination.
- RSPAN does not support BPDU monitoring.
- Do not configure RSPAN VLANs as sources in VSPAN sessions.
- You can configure any VLAN as an RSPAN VLAN as long as all participating network devices support configuration of RSPAN VLANs and you use the same RSPAN VLAN for each RSPAN session in all participating network devices.
- Entering SPAN configuration commands does not clear previously configured SPAN parameters. You must enter the **no monitor session** command to clear configured SPAN parameters.

Configuring Local SPAN and RSPAN

These sections describe how to configure local SPAN and RSPAN:

- [Local SPAN and RSPAN Configuration Overview, page 34-8](#)
- [Configuring RSPAN VLANs, page 34-9](#)
- [Configuring Local or RSPAN Sources, page 34-9](#)
- [Monitoring Specific Source VLANs on a Source Trunk Port, page 34-10](#)
- [Configuring Local SPAN and RSPAN Destinations, page 34-10](#)
- [Verifying the Configuration, page 34-12](#)
- [Configuration Examples, page 34-13](#)

**Note**

With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Local SPAN and RSPAN Configuration Overview

To configure a local SPAN session, use the same session number for the sources and the destination ports.

To configure an RSPAN source session, use the same session number for a source and a destination RSPAN VLAN.

To configure an RSPAN destination session, use the same session number for a source RSPAN VLAN and a destination port.

Configuring RSPAN VLANs

To configure a VLAN as an RSPAN VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan <i>vlan_ID</i> {[- <i>vlan_ID</i>] [, <i>vlan_ID</i>]}	Creates or modifies an Ethernet VLAN, a range of Ethernet VLANs, or several Ethernet VLANs specified in a comma-separated list (do not enter space characters).
Step 2	Router(config-vlan)# remote-span Router(config-vlan)# no remote-span	Configures the VLAN as an RSPAN VLAN. Clears the RSPAN VLAN configuration.
Step 3	Router(config-vlan)# end	Updates the VLAN database and returns to privileged EXEC mode.

Configuring Local or RSPAN Sources



Note

To configure an RSPAN source session, configure a source with an RSPAN VLAN as the destination. To configure an RSPAN destination session, configure an RSPAN VLAN as the source and a port as the destination.

To configure a local SPAN or RSPAN source, perform this task:

Command	Purpose
Router(config)# monitor session <i>session_number</i> source <i>{single_interface interface_list interface_range mixed_interface_list single_vlan vlan_list vlan_range mixed_vlan_list}</i> [rx tx both] {remote vlan rspan_vlan_ID}	Configures the session number, the source ports, VLANs, or RSPAN VLAN, and the traffic direction to be monitored.
Router(config)# no monitor session <i>{session_number all local range session_range[, session_range], ...}</i> remote	Clears the monitor configuration.

When configuring monitor sessions, note the following syntax information:

- *single_interface* is **interface type slot/port**; type is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...



Note

In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface type slot/first_port - last_port**
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is a the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...

- *vlan_range* is *first_vlan_ID - last_vlan_ID*
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...

When clearing monitor sessions, note the following syntax information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*



Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

This example shows how to configure session 1 to monitor bidirectional traffic from Fast Ethernet port 5/1:

```
Router(config)# monitor session 1 source interface fastethernet 5/1
```

Monitoring Specific Source VLANs on a Source Trunk Port

To monitor specific VLANs when the local or RSPAN source is a trunk port, perform this task:

Command	Purpose
Router(config)# monitor session <i>session_number</i> filter { <i>vlan_ID</i> } [, -]	Monitors specific VLANs when the source is a trunk port.
Router(config)# no monitor session <i>session_number</i> filter { <i>vlan_ID</i> }	Clears trunk source configuration.

This example shows how to monitor VLANs 1 through 5 and VLAN 9 when the source is a trunk port:

```
Router(config)# monitor session 2 filter vlan 1 - 5 , 9
```

Configuring Local SPAN and RSPAN Destinations

These sections describe how to configure local SPAN and RSPAN destinations:

- [Configuring a Destination Port as an Unconditional Trunk, page 34-11](#)
- [Configuring a Local or RSPAN Destination, page 34-11](#)

Configuring a Destination Port as an Unconditional Trunk

To tag the monitored traffic with Release 12.1(13)E and later releases, configure the destination port as a trunk.

To configure the destination port as a trunk, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching (required only if the LAN port is not already configured for Layer 2 switching).
Step 3	Router(config-if)# switchport trunk encapsulation { isl dot1q }	Configures the encapsulation, which configures the Layer 2 switching port as either an ISL or 802.1Q trunk.
Step 4	Router(config-if)# switchport mode trunk	Configures the port to trunk unconditionally.
Step 5	Router(config-if)# switchport nonegotiate	Configures the trunk not to use DTP.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to configure a port as an unconditional IEEE 802.1q trunk:

```
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# switchport nonegotiate
```

Configuring a Local or RSPAN Destination



Note

To configure an RSPAN source session, configure a source with an RSPAN VLAN as the destination. To configure an RSPAN destination session, configure an RSPAN VLAN as the source and a port as the destination.

To configure a local or RSPAN destination, perform this task:

Command	Purpose
Router(config)# monitor session <i>session_number</i> destination { <i>single_interface</i> <i>interface_list</i> <i>interface_range</i> <i>mixed_interface_list</i> } { remote vlan <i>rspan_vlan_ID</i> }	Configures the session number and the destination ports or RSPAN VLAN.
Router(config)# no monitor session { <i>session_number</i> all local range <i>session_range</i> [[, <i>session_range</i>],...]} remote }	Clears the monitor configuration.



Note

To tag the monitored traffic, you must configure the port to trunk unconditionally before you configure it as a destination (see the “[Configuring a Destination Port as an Unconditional Trunk](#)” section on [page 34-11](#)).

When configuring monitor sessions, note the following syntax information:

- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...



Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...

When clearing monitor sessions, note the following syntax information:

- Enter the **no monitor session** *number* command with no other parameters to clear session *session_number*.
- *session_range* is *first_session_number*-*last_session_number*



Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

This example shows how to configure Fast Ethernet port 5/48 as the destination for SPAN session 1:

```
Router(config)# monitor session 1 destination interface fastethernet 5/48
```

Verifying the Configuration

This example shows how to verify the configuration of session 2:

```
Router# show monitor session 2
Session 2
-----
Type : Remote Source Session

Source Ports:
  RX Only:      Fa3/1
Dest RSPAN VLAN: 901
Router#
```


This example shows how to display the full details of session 2:

```
Router# show monitor session 2 detail
Session 2
-----
Type : Remote Source Session

Source Ports:
  RX Only:      Fa1/1-3
  TX Only:      None
  Both:         None
Source VLANs:
  RX Only:      None
  TX Only:      None
  Both:         None
Source RSPAN VLAN: None
Destination Ports: None
Filter VLANs:   None
Dest RSPAN VLAN: 901
```

Configuration Examples

This example shows how to configure RSPAN source session 2:

```
Router(config)# monitor session 2 source interface fastethernet1/1 - 3 rx
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to clear the configuration for sessions 1 and 2:

```
Router(config)# no monitor session range 1-2
```

This example shows how to configure an RSPAN source session with multiple sources:

```
Router(config)# monitor session 2 source interface fastethernet 5/15 , 7/3 rx
Router(config)# monitor session 2 source interface gigabitethernet 1/2 tx
Router(config)# monitor session 2 source interface port-channel 102
Router(config)# monitor session 2 source filter vlan 2 - 3
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to remove sources for a session:

```
Router(config)# no monitor session 2 source interface fastethernet 5/15 , 7/3
```

This example shows how to remove options for sources for a session:

```
Router(config)# no monitor session 2 source interface gigabitethernet 1/2
Router(config)# no monitor session 2 source interface port-channel 102 tx
```

This example shows how to remove VLAN filtering for a session:

```
Router(config)# no monitor session 2 filter vlan 3
```

This example shows how to configure an RSPAN destination session:

```
Router(config)# monitor session 8 source remote vlan 901
Router(config)# monitor session 8 destination interface fastethernet 1/2 , 2/3
```




Configuring Web Cache Services Using WCCP

This chapter describes how to configure the Catalyst 6500 series switches to redirect traffic to cache engines (web caches) using the Web Cache Communication Protocol (WCCP), and describes how to manage cache engine clusters (cache farms).



Note

- To use the WCCP Layer 2 PFC redirection feature, configure WCCP on the Catalyst 6500 series switch as described in this chapter and configure accelerated WCCP on the cache engine as described in the following publication:
<http://www.cisco.com/univercd/cc/td/doc/product/webcache/uce/acns42/cnfg42/transprt.htm#xtocid34>
- A release of Cisco Application and Content Networking System (ACNS) software later than Release 4.2.2 supports the **ip wccp service accelerated** command with a PFC2.
- A cache engine configured for mask assignment that tries to join a farm where the selected assignment method is hash remains out of the farm as long as the cache engine assignment method does not match that of the existing farm.
- With WCCP Layer 2 PFC redirection as the forwarding method for a service group, the packet counters in the **show ip wccp service_name** command output displays flow counts instead of packet counts.

This chapter consists of these sections:

- [Understanding WCCP, page 35-2](#)
- [Restrictions for WCCPv2, page 35-7](#)
- [Configuring WCCP, page 35-7](#)
- [Verifying and Monitoring WCCP Configuration Settings, page 35-12](#)
- [WCCP Configuration Examples, page 35-12](#)



Note

The tasks in this chapter assume that you have already configured cache engines on your network. For specific information on hardware and network planning associated with Cisco Cache Engines and WCCP, see the Product Literature and Documentation links available on the Cisco.com Web Scaling site at <http://www.cisco.com/warp/public/cc/pd/cxsr/ces/index.shtml>.

Understanding WCCP

These sections describe WCCP:

- [WCCP Overview, page 35-2](#)
- [Hardware Acceleration, page 35-2](#)
- [Understanding WCCPv1 Configuration, page 35-3](#)
- [Understanding WCCPv2 Configuration, page 35-4](#)
- [WCCPv2 Features, page 35-5](#)

WCCP Overview

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that allows you to integrate cache engines (such as the Cisco Cache Engine 550) into your network infrastructure.

**Note**

Cisco Systems replaced the Cache Engine 500 Series platforms with Content Engine Platforms in July 2001. Cache Engine Products were the Cache Engine 505, 550, 570, and 550-DS3. Content Engine Products are the Content Engine 507, 560, 590, and 7320.

The Cisco IOS WCCP feature allows use of Cisco Cache Engines (or other caches running WCCP) to localize web traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables Cisco IOS routing platforms to transparently redirect content requests. The main benefit of transparent redirection is that users need not configure their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a cache engine. The word “transparent” in this case means that the end user does not know that a requested file (such as a web page) came from the cache engine instead of from the originally specified server.

When a cache engine receives a request, it attempts to service it from its own local cache. If the requested information is not present, the cache engine issues its own request to the originally targeted server to get the required information. When the cache engine retrieves the requested information, it forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of cache engines, called a *cache engine cluster*, to provide content to a router or multiple routers. Network administrators can easily scale their cache engines to handle heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each cache member to work in parallel, resulting in linear scalability. Clustering cache engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 cache engines to scale to your desired capacity.

Hardware Acceleration

Catalyst 6500 series switches provide hardware acceleration for directly connected Cisco Cache Engines, which is more efficient than Layer 3 redirection in software on the MSFC with generic route encapsulation (GRE).

With Release 12.1(2)E and later releases, WCCP Layer 2 PFC redirection allows Cisco Cache Engines to use hardware-supported Layer 2 redirection. A directly connected Cache Engine can be configured to negotiate use of the WCCP Layer 2 PFC Redirection feature. The WCCP Layer 2 PFC redirection feature requires no configuration on the MSFC. The **show ip wccp web-cache detail** command displays which redirection method is in use for each cache.

The following guidelines apply to WCCP Layer 2 PFC redirection:

- The WCCP Layer 2 PFC redirection feature sets the IP flow mask to full-flow mode.
- You can configure the Cisco Cache Engine software release 2.2 or later releases to use the WCCP Layer 2 PFC redirection feature.
- Layer 2 redirection takes place on the PFC and is not visible to the MSFC. The **show ip wccp web-cache detail** command on the MSFC displays statistics for only the first packet of a Layer 2 redirected flow, which provides an indication of how many flows, rather than packets, are using Layer 2 redirection. Entering the **show mls entries** command displays the other packets in the Layer 2 redirected flows.

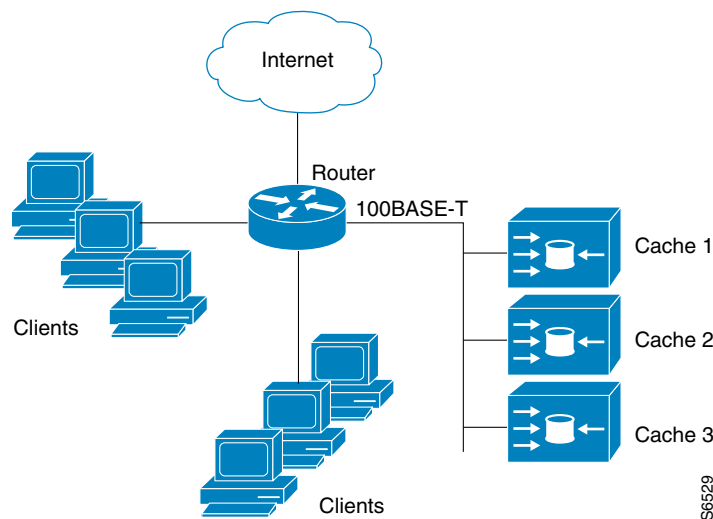
**Note**

A future release of Cisco Application and Content Networking System (ACNS) software (Release 4.2.2 or later) supports the **accelerated** keyword with Release 12.1(13)E and later releases.

Understanding WCCPv1 Configuration

With WCCP-Version 1, only a single router services a cluster. In this scenario, this router is the device that performs all the IP packet redirection. [Figure 35-1](#) illustrates how this configuration appears.

Figure 35-1 Cisco Cache Engine Network Configuration Using WCCP-Version 1



Content is not duplicated on the cache engines. The benefit of using multiple caches is that you can scale a caching solution by clustering multiple physical caches to appear as one logical cache.

The following sequence of events details how WCCPv1 configuration works:

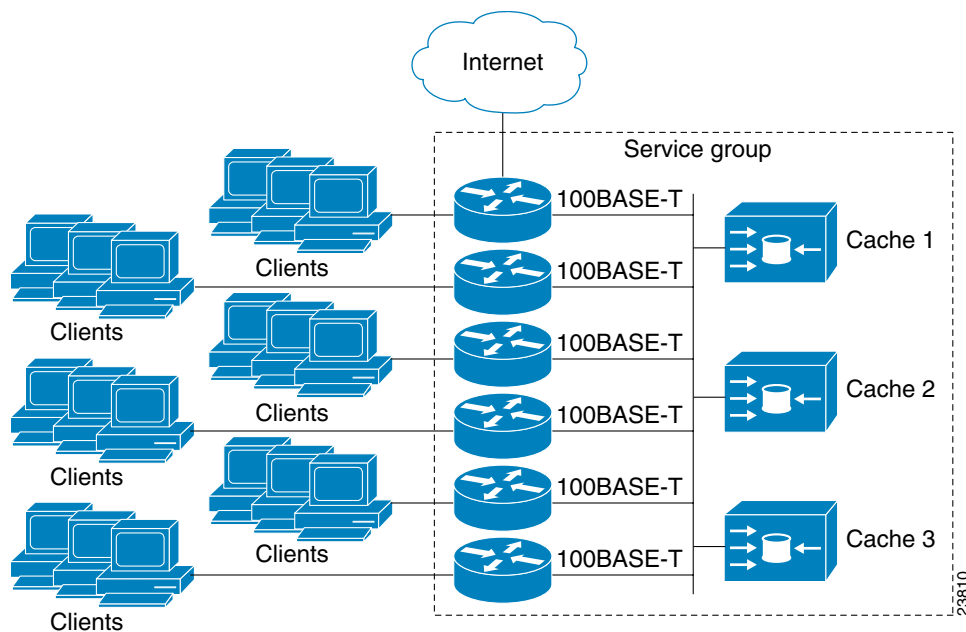
1. Each cache engine is configured by the system administrator with the IP address of the control router. Up to 32 cache engines can connect to a single control router.

2. The cache engines send their IP addresses to the control router using WCCP, indicating their presence. Routers and cache engines communicate to each other via a control channel; this channel is based on UDP port 2048.
3. This information is used by the control router to create a cluster view (a list of caches in the cluster). This view is sent to each cache in the cluster, essentially making all the cache engines aware of each other. A stable view is established after the membership of the cluster remains the same for a certain amount of time.
4. Once a stable view has been established, one cache engine is elected as the lead cache engine. (The lead is defined as the cache engine seen by all the cache engines in the cluster with the lowest IP address). This lead cache engine uses WCCP to indicate to the control router how IP packet redirection should be performed. Specifically, the lead cache engine designates how redirected traffic should be distributed across the cache engines in the cluster.

Understanding WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a cache cluster. This is in contrast to WCCPv1 in which only one router could redirect content requests to a cluster. [Figure 35-2](#) illustrates a sample configuration using multiple routers.

Figure 35-2 Cisco Cache Engine Network Configuration Using WCCP v2



The subset of cache engines within a cluster and routers connected to the cluster that are running the same service is known as a *service group*. Available services include TCP and User Datagram Protocol (UDP) redirection.

Using WCCPv1, the cache engines were configured with the address of the single router. WCCPv2 requires that each cache engine be aware of all the routers in the service group. To specify the addresses of all the routers in a service group, you must choose one of the following methods:

- **Unicast**—A list of router addresses for each of the routers in the group is configured on each cache engine. In this case the address of each router in the group must be explicitly specified for each cache engine during configuration.
- **Multicast**—A single multicast address is configured on each cache engine. In the multicast address method, the cache engine sends a single-address notification that provides coverage for all routers in the service group. For example, a cache engine could indicate that packets should be sent to a multicast address of 224.0.0.100, which would send a multicast packet to all routers in the service group configured for group listening using WCCP (see the **ip wccp group-listen** interface configuration command for details).

The multicast option is easier to configure because you need only specify a single address on each cache engine. This option also allows you to add and remove routers from a service group dynamically, without needing to reconfigure the cache engines with a different list of addresses each time.

The following sequence of events details how WCCPv2 configuration works:

1. Each cache engine is configured with a list of routers.
2. Each cache engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of cache engines in the group.
3. Once the view is consistent across all cache engines in the cluster, one cache engine is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

The following sections describe how to configure WCCPv2 on routers so they may participate in a service group.

WCCPv2 Features

These sections describe WCCPv2 features:

- [Support for Non-HTTP Services](#)
- [Support for Multiple Routers](#)
- [MD5 Security](#)
- [Web Cache Packet Return](#)
- [Load Distribution](#)

Support for Non-HTTP Services

WCCPv2 allows redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. WCCPv1 supported the redirection of HTTP (TCP port 80) traffic only. WCCPv2 supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and real audio, video, and telephony applications.

To accommodate the various types of services available, WCCPv2 introduces the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as “98”) or a predefined service keywords (such as “web-cache”). This information is used to validate that service group members are all using or providing the same service.

The cache engines in service group specify traffic to be redirected by protocol (TCP or UDP) and port (source or destination). Each service group has a priority status assigned to it. Packets are matched against service groups in priority order.

Support for Multiple Routers

WCCPv2 allows multiple routers to be attached to a cluster of cache engines. The use of multiple routers in a service group allows for redundancy, interface aggregation, and distribution of the redirection load.

MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and cache engines become part of the service group using passwords and the HMAC MD5 standard. Shared-secret MD5 one-time authentication (set using the `ip wccp [password [0-7] password]` global configuration command) enables messages to be protected against interception, inspection, and replay.

Web Cache Packet Return

If a cache engine is unable to provide a requested object it has cached due to error or overload, the cache engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the cache engine unserved. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the cache cluster). This provides error handling transparency to clients.

Typical reasons why a cache engine would reject packets and initiate the packet return feature include the following:

- Instances when the cache engine is overloaded and has no room to service the packets
- Instances when the cache engine is filtering for certain conditions that make caching packets counterproductive (for example, when IP authentication has been turned on)

Load Distribution

WCCPv2 can be used to adjust the load being offered to individual cache engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. WCCPv2 allows the designated cache to adjust the load on a particular cache and balance the load across the caches in a cluster. WCCPv2 uses three techniques to perform load distribution:

- **Hot Spot Handling**—Allows an individual hash bucket to be distributed across all the cache engines. Prior to WCCPv2, information from one hash bucket could only go to one cache engine.
- **Load Balancing**—Allows the set of hash buckets assigned to a cache engine to be adjusted so that the load can be shifted from an overwhelmed cache engine to other members that have available capacity.
- **Load Shedding**—Enables the router to selectively redirect the load to avoid exceeding the capacity of a cache engine.

By using these hashing parameters, you can prevent one cache from being overloaded and reduce the potential for congestion.

Restrictions for WCCPv2

The following limitations apply to WCCP v2:

- WCCP works only with IP networks.
- For routers servicing a multicast cluster, the time to live (TTL) value must be set at 15 or fewer.
- Because the messages may now be IP multicast, members may receive messages that will not be relevant or are duplicates. Appropriate filtering needs to be performed.
- Service groups can comprise up to 32 cache engines and 32 routers.
- All cache engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Multicast addresses must be from 224.0.0.0 to 239.255.255.255.

Configuring WCCP

The following configuration tasks assume that you have already installed and configured the cache engines you want to include in your network. You must configure the cache engines in the cluster before configuring WCCP functionality on your routers. Refer to the *Cisco Cache Engine User Guide* for cache engine configuration and setup tasks.

IP must be configured on the router interface connected to the cache engines and on the router interface connected to the Internet. Cisco Cache Engines require use of a Fast Ethernet interface for a direct connection. Examples of router configuration tasks follow this section. For complete descriptions of the command syntax, refer to the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Perform the tasks found in the following sections to configure WCCP on a router:

- [Specifying a Version of WCCP, page 35-7](#) (Optional)
- [Configuring a Service Group Using WCCPv2, page 35-8](#) (Required)
- [Excluding Traffic on a Specific Interface from Redirection, page 35-9](#) (Optional)
- [Registering a Router to a Multicast Address, page 35-10](#) (Optional)
- [Using Access Lists for a WCCP Service Group, page 35-10](#) (Optional)
- [Setting a Password for a Router and Cache Engines, page 35-11](#) (Optional)

Specifying a Version of WCCP

Until you configure a WCCP service using the `ip wccp {web-cache | service-number}` global configuration command, WCCP is disabled on the router. The first use of a form of the `ip wccp` command enables WCCP. By default WCCPv2 is used for services, but you can use WCCPv1 functionality instead. To change the running version of WCCP from Version 2 to Version 1, or to return to WCCPv2 after an initial change, perform this task in EXEC mode:

Command	Purpose
Router# <code>ip wccp version {1 2}</code>	Specifies which version of WCCP to configure on a router. WCCPv2 is the default version.

WCCPv1 does not use the WCCP commands from earlier Cisco IOS versions. Instead, use the WCCP commands documented in this chapter. If a function is not allowed in WCCPv1, an error prompt will be printed to the screen. For example, if WCCPv1 is running on the router and you try to configure a dynamic service, the following message will be displayed: “WCCP V1 only supports the web-cache service.” The **show ip wccp EXEC** command will display the WCCP protocol version number that is currently running on your router.

Configuring a Service Group Using WCCPv2

WCCPv2 uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is web cache, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the cache engines. This service is referred to as a *well-known service*, because the characteristics of the web cache service are known by both the router and cache engines. A description of a well-known service is not required beyond a service identification (in this case, the command line interface (CLI) provides a **web-cache** keyword in the command syntax).

In addition to the web cache service, there can be up to seven dynamic services running concurrently in a service group.



Note

More than one service can run on a router at the same time, and routers and cache devices can be part of multiple service groups at the same time.

The dynamic services are defined by the cache engines; the cache instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group’s traffic, because this information is provided by the first web cache to join the group. In a dynamic service, up to eight ports can be specified within a single protocol.

Cisco Cache Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other cache devices may use this service number for some other service. The following configuration information deals with enabling general services on Cisco routers. Refer to the cache server documentation for information on configuring services on cache devices.

To enable a service on a Catalyst 6500 series switch, perform this task:

	Command	Purpose
Step 1	Router(config)# ip wccp { web-cache <i>service-number</i> } [accelerated] [group-address <i>groupaddress</i>] [redirect-list <i>access-list</i>] [group-list <i>access-list</i>] [password <i>password</i>]	Specifies a web cache or dynamic service to enable on the router, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service.
Step 2	Router(config)# interface <i>type number</i>	Specifies an interface to configure and enters interface configuration mode.
Step 3	Router(config-if)# ip wccp { web-cache <i>service-number</i> } redirect { out in }	Enables WCCP redirection on the specified interface.

**Note**

- A future release of Cisco Application and Content Networking System (ACNS) software (Release 4.2.2 or later) and Release 12.1(13)E and later releases support the **ip wccp service accelerated** command with a PFC2.
- Release 12.1(13)E and later releases support the **ip wccp service redirect in** interface command in software on the MSFC2.

As indicated by the **out** and **in** keyword options in the **ip wccp service redirect** command, redirection can be specified for outbound interfaces or inbound interfaces.

Inbound traffic can be configured to use Cisco Express Forwarding (CEF), distributed Cisco Express Forwarding (dCEF), Fast Forwarding, or Process Forwarding.

Configuring WCCP for redirection for inbound traffic on interfaces allows you to avoid the overhead associated with CEF forwarding for outbound traffic. Setting an output feature on any interface results in the slower switching path of the feature being taken by all packets arriving at all interfaces. Setting an input feature on an interface results in only those packets arriving at that interface taking the configured feature path; packets arriving at other interfaces will use the faster default path. Configuring WCCP for inbound traffic also allows packets to be classified before the routing table lookup, which provides faster redirection of packets.

Specifying a Web Cache Service

To configure a web-cache service, perform this task:

	Command	Purpose
Step 1	Router(config)# ip wccp web-cache	Enables the web cache service on the router.
Step 2	Router(config)# interface <i>type number</i>	Targets an interface number for which the web cache service will run, and enters interface configuration mode.
Step 3	Router(config-if)# ip wccp web-cache redirect { out in }	Enables the check on packets to determine if they qualify to be redirected to a web cache, using the interface specified in Step 2.

Excluding Traffic on a Specific Interface from Redirection

To exclude any interface from redirecting inbound traffic, perform this task in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface <i>type number</i>	Specifies an interface to configure, and enters interface configuration mode.
Step 2	Router(config-if)# ip wccp redirect exclude in	Allows inbound packets on this interface to be excluded from redirection.

Registering a Router to a Multicast Address

If you decide to use the multicast address option for your service group, you must configure the router to listen for the multicast broadcasts on an interface. To configure the router, perform this task:

	Command	Purpose
Step 1	Router(config)# ip wccp {web-cache service-number} group-address groupaddress	Specifies the multicast address for the service group.
Step 2	Router(config)# interface type number	Specifies the interface to be configured for multicast reception.
Step 3	Router(config-if)# ip wccp {web-cache service-number} group-listen	Enables the reception of IP multicast packets (content originating from the cache engines) on the interface specified in Step 2.

For network configurations where redirected traffic needs to traverse an intervening router, the router being traversed must be configured to perform IP multicast routing. You must configure the following two components to enable traversal over an intervening router:

- Enable IP multicast routing using the **ip multicast routing** interface configuration command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ip wccp group-listen** interface configuration command (note that earlier Cisco IOS versions required the use of the **ip pim** interface configuration command).

Using Access Lists for a WCCP Service Group

To configure the router to use an access list to determine which traffic should be directed to which cache engines, perform this task in global configuration mode:

	Command	Purpose
Step 1	Router(config)# access-list access-list permit ip host host-address [destination-address destination-host any]	Creates an access list that enables or disables traffic redirection to the cache engine.
Step 2	Router(config)# ip wccp web-cache group-list access-list	Indicates to the router from which IP addresses of cache engines to accept packets.

To disable caching for certain clients, perform this task in global configuration mode:

	Command	Purpose
Step 1	Router(config)# access-list access-list permit ip host host-address [destination-address destination-host any]	Creates an access list that enables or disables traffic redirection to the cache engine.
Step 2	Router(config)# ip wccp web-cache redirect-list access-list	Sets the access list used to enable redirection.

Setting a Password for a Router and Cache Engines

MD5 password security requires that each router and cache engine that wants to join a service group be configured with the service group password. The password can consist of up to seven characters. Each cache engine or router in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

To configure an MD5 password for use by the router in WCCP communications, perform this task in global configuration mode:

Command	Purpose
Router(config)# ip wccp web-cache password <i>password</i>	Sets an MD5 password on the router.

Verifying and Monitoring WCCP Configuration Settings

To verify and monitor the configuration settings for WCCP, use the following commands in EXEC mode:

Command	Purpose
Router# show ip wccp [web-cache <i>service-number</i>]	Displays global information related to WCCP, including the protocol version currently running, the number of cache engines in the routers service group, which cache engine group is allowed to connect to the router, and which access list is being used.
Router# show ip wccp { web-cache <i>service-number</i> } detail	Queries the router for information on which cache engines of a specific service group the router has detected. The information can be displayed for either the web cache service or the specified dynamic service.
Router# show ip interface	Displays status about whether any ip wccp redirection commands are configured on an interface. For example, “Web Cache Redirect is enabled / disabled.”
Router# show ip wccp { web-cache <i>service-number</i> } view	Displays which devices in a particular service group have been detected and which cache engines are having trouble becoming visible to all other routers to which the current router is connected. The view keyword indicates a list of addresses of the service group. The information can be displayed for either the web cache service or the specified dynamic service. For further troubleshooting information, use the show ip wccp { web-cache <i>service number</i> } service command.

WCCP Configuration Examples

This section provides the following configuration examples:

- [Changing the Version of WCCP on a Router Example, page 35-13](#)
- [Performing a General WCCPv2 Configuration Example, page 35-13](#)
- [Running a Web Cache Service Example, page 35-13](#)
- [Running a Reverse Proxy Service Example, page 35-14](#)
- [Registering a Router to a Multicast Address Example, page 35-14](#)
- [Using Access Lists Example, page 35-14](#)

- [Setting a Password for a Router and Cache Engines Example, page 35-15](#)
- [Verifying WCCP Settings Example, page 35-15](#)

Changing the Version of WCCP on a Router Example

The following example shows the process of changing the WCCP version from the default of WCCPv2 to WCCPv1, and enabling the web-cache service in WCCPv1:

```
Router# show ip wccp
% WCCP version 2 is not enabled
Router# configure terminal
Router(config)# ip wccp version 1
Router(config)# end
Router# show ip wccp
% WCCP version 1 is not enabled

Router# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip wccp web-cache
Router(config)# end
Router# show ip wccp
Global WCCP information:
  Router information:
    Router Identifier:          10.4.9.8
    Protocol Version:          1.0
  . . .
```

Performing a General WCCPv2 Configuration Example

The following example shows a general WCCPv2 configuration session:

```
Router# configure terminal
Router(config)# ip wccp web-cache group-address 224.1.1.100 password alaska1
Router(config)# interface ethernet0
Router(config-if)# ip wccp web-cache redirect out
```

Running a Web Cache Service Example

The following example shows a web cache service configuration session:

```
router# configure terminal
router(config)# ip wccp web-cache
router(config)# interface ethernet 0
router(config-if)# ip wccp web-cache redirect out
Router(config-if)# ^Z
Router# copy running-config startup-config
```

The following example shows a configuration session in which redirection of HTTP traffic arriving on interface 0/1 is enabled:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface ethernet 0/1
Router(config-if)# ip wccp web-cache redirect in
Router(config-if)# ^Z
Router# show ip interface ethernet 0/1
```

```

.
.
.
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
.
.
.

```

Running a Reverse Proxy Service Example

The following example assumes you are configuring a service group using Cisco Cache Engines, which use dynamic service 99 to run a reverse proxy service:

```

router# configure terminal
router(config)# ip wccp 99
router(config)# interface ethernet 0
router(config-if)# ip wccp 99 redirect out

```

Registering a Router to a Multicast Address Example

The following example shows how to register a router to a multicast address of 224.1.1.100:

```

Router(config)# ip wccp web-cache group-address 224.1.1.100
Router(config)# interface ethernet 0
Router(config-if)# ip wccp web-cache group-listen

```

The following example shows a router configured to run a reverse proxy service, using the multicast address of 224.1.1.1. Redirection applies to packets outgoing through interface ethernet 0:

```

Router(config)# ip wccp 99 group-address 224.1.1.1
Router(config)# interface ethernet 0
Router(config-if)# ip wccp 99 redirect out

```

Using Access Lists Example

To achieve better security, you can use a standard access list to notify the router which IP addresses are valid addresses for a cache engine attempting to register with the current router. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```

router(config)# access-list 10 permit host 11.1.1.1
router(config)# access-list 10 permit host 11.1.1.2
router(config)# access-list 10 permit host 11.1.1.3
router(config)# ip wccp web-cache group-list 10

```

To disable caching for certain clients, servers, or client/server pairs, you can use WCCP access lists. The following example shows that any requests coming from 10.1.1.1 to 12.1.1.1 will bypass the cache and that all other requests will be serviced normally:

```

Router(config)# ip wccp web-cache redirect-list 120
Router(config)# access-list 120 deny tcp host 10.1.1.1 any
Router(config)# access-list 120 deny tcp any host 12.1.1.1
Router(config)# access-list 120 permit ip any any

```


The following example configures a router to redirect web-related packets received through interface ethernet 0/1, destined to any host except 209.165.196.51:

```
Router(config)# access-list 100 deny ip any host 209.165.196.51
Router(config)# access-list 100 permit ip any any
Router(config)# ip wccp web-cache redirect-list 100
Router(config)# interface Ethernet 0/1
Router(config-if)# ip wccp web-cache redirect in
```

Setting a Password for a Router and Cache Engines Example

The following example shows a WCCPv2 password configuration session where the password is alaskal:

```
router# configure terminal
router(config)# ip wccp web-cache password alaskal
```

Verifying WCCP Settings Example

To verify your configuration changes, use the **more system:running-config** EXEC command. The following example shows that the both the web cache service and dynamic service 99 are enabled on the router:

```
router# more system:running-config

Building configuration...
Current configuration:
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname router4
!
enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNT1
enable password alabamal
!
ip subnet-zero
ip wccp web-cache
ip wccp 99
ip domain-name cisco.com
ip name-server 10.1.1.1
ip name-server 10.1.1.2
ip name-server 10.1.1.3
!
!
!
interface Ethernet0
ip address 10.3.1.2 255.255.255.0
no ip directed-broadcast
ip wccp web-cache redirect out
ip wccp 99 redirect out
no ip route-cache
no ip mroute-cache
!
```

```
interface Ethernet1
ip address 10.4.1.1 255.255.255.0
no ip directed-broadcast
ip wccp 99 redirect out
no ip route-cache
no ip mroute-cache
!
interface Serial0
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password alaska1
login
!
end
```



Configuring SNMP IfIndex Persistence

This chapter describes how to configure the SNMP ifIndex persistence feature on Catalyst 6500 series switches. Release 12.1(13)E and later releases support SNMP ifIndex persistence.

This chapter consists of these sections:

- [Understanding SNMP IfIndex Persistence, page 36-1](#)
- [Configuring SNMP IfIndex Persistence, page 36-1](#)
- [Configuration Examples, page 36-3](#)

Understanding SNMP IfIndex Persistence

The SNMP ifIndex persistence feature provides an interface index (ifIndex) value that is retained and used when the switch reboots. The ifIndex value is a unique identifying number associated with a physical or logical interface.

There is no requirement in the relevant RFCs that the correspondence between particular ifIndex values and their interfaces be maintained when the switch reboots, but many applications (for example, device inventory, billing, and fault detection) require maintenance of this correspondence.

You can poll the switch at regular intervals to correlate the interfaces to the ifIndexes, but it is not practical to poll constantly. The SNMP ifIndex persistence feature provides permanent ifIndex values, which eliminates the need to poll interfaces.

The following definitions are based on RFC 2233, “The Interfaces Group MIB using SMIV2.” The following terms are values in the Interfaces MIB (IF-MIB):

- **ifIndex**—A unique number (greater than zero) that identifies each interface for SNMP identification of that interface.
- **ifName**—The text-based name of the interface, for example, “ethernet 3/1.”
- **ifDescr**—A description of the interface. Recommended information for this description includes the name of the manufacturer, the product name, and the version of the interface hardware and software.

Configuring SNMP IfIndex Persistence

These sections describe how to configure SNMP ifIndex persistence:

- [Enabling and Disabling SNMP IfIndex Persistence Globally, page 36-2](#) (Optional)
- [Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces, page 36-2](#) (Optional)

**Note**

To verify that ifIndex commands have been configured, use the **more system:running-config** command.

Enabling and Disabling SNMP IfIndex Persistence Globally

SNMP ifIndex persistence is disabled by default. To globally enable SNMP ifIndex persistence, perform this task:

Command	Purpose
Router(config)# snmp-server ifindex persist	Globally enables SNMP ifIndex persistence.

To globally disable SNMP ifIndex persistence after enabling it, perform this task:

Command	Purpose
Router(config)# no snmp-server ifindex persist	Globally disables SNMP ifIndex persistence.

Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces

To enable SNMP ifIndex persistence only on a specific interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 2	Router(config-if)# snmp ifindex persist	Enables SNMP ifIndex persistence on the specified interface.
	Router(config-if)# no snmp ifindex persist	Disables SNMP ifIndex persistence on the specified interface.
Step 3	Router(config-if)# exit	Exits interface configuration mode.

1. *type* = any supported interface type.

**Note**

The **no snmp ifindex persistence** interface command cannot be used on subinterfaces. A command applied to an interface is automatically applied to all the subinterfaces associated with that interface.

To clear the interface-specific SNMP ifIndex persistence setting and configure the interface to use the global configuration setting, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type slot/port</i>	Enters interface configuration mode for the specified interface. Note that the syntax of the interface command will vary depending on the platform you are using.
Step 2	Router(config-if)# snmp ifindex clear	Clears any interface-specific SNMP ifIndex persistence configuration for the specified interface and returns to the global configuration setting.
Step 3	Router(config-if)# exit	Exits interface configuration mode.

Configuration Examples

This section provides the following configuration examples:

- [Enabling SNMP IfIndex Persistence on All Interfaces Example, page 36-3](#)
- [Enabling SNMP IfIndex Persistence on a Specific Interface Example, page 36-3](#)
- [Disabling SNMP IfIndex Persistence on a Specific Interface Example, page 36-3](#)
- [Clearing SNMP IfIndex Persistence Configuration from a Specific Interface Example, page 36-4](#)

Enabling SNMP IfIndex Persistence on All Interfaces Example

In the following example, SNMP ifIndex persistence is enabled for all interfaces:

```
router(config)# snmp-server ifindex persist
```

Enabling SNMP IfIndex Persistence on a Specific Interface Example

In the following example, SNMP ifIndex persistence is enabled for Ethernet interface 3/1 only:

```
router(config)# interface ethernet 3/1
router(config-if)# snmp ifindex persist
router(config-if)# exit
```

Disabling SNMP IfIndex Persistence on a Specific Interface Example

In the following example, SNMP ifIndex persistence is disabled for Ethernet interface 3/1 only:

```
router(config)# interface ethernet 3/1
router(config-if)# no snmp ifindex persist
router(config-if)# exit
```

Clearing SNMP IfIndex Persistence Configuration from a Specific Interface Example

In the following example, any previous setting for SNMP ifIndex persistence on Ethernet interface 3/1 is removed from the configuration. If SNMP ifIndex persistence is globally enabled, SNMP ifIndex persistence will be enabled for Ethernet interface 3/1. If SNMP ifIndex persistence is globally disabled, SNMP ifIndex persistence will be disabled for Ethernet interface 3/1.

```
router(config)# interface ethernet 3/1  
router(config-if)# snmp ifindex clear  
router(config-if)# exit
```



Configuring the Switch Fabric Module

This chapter describes how to configure the Switch Fabric Module (SFM) for the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How the Switch Fabric Module Works, page 37-1](#)
- [Configuring the Switch Fabric Module, page 37-3](#)
- [Monitoring the Switch Fabric Module, page 37-5](#)

Understanding How the Switch Fabric Module Works

These sections describe how the Switch Fabric Module works:

- [Switch Fabric Module Overview, page 37-1](#)
- [Switch Fabric Module Slots, page 37-2](#)
- [Switch Fabric Redundancy, page 37-2](#)
- [Forwarding Decisions for Layer 3-Switched Traffic, page 37-2](#)
- [Switching Modes, page 37-2](#)

Switch Fabric Module Overview



Note

The Switch Fabric Module is supported only with Supervisor Engine 2.

The Switch Fabric Module creates a dedicated connection between fabric-enabled modules and provides uninterrupted transmission of frames between these modules. In addition to the direct connection between fabric-enabled modules provided by the Switch Fabric Module, fabric-enabled modules also have a direct connection to the 32-Gbps forwarding bus.

The Switch Fabric Module does not have a console. A two-line LCD display on the front panel shows fabric utilization, software revision, and basic system information.

Switch Fabric Module Slots

With a WS-C6513 chassis, install the Switch Fabric Modules in either slot 7 or 8.

**Note**

In a WS-C6513 chassis, only slots 9 through 13 support dual-switch fabric interface switching modules (for example, WS-X6816-GBIC).

With all other chassis, install the Switch Fabric Modules in either slot 5 or 6.

Switch Fabric Redundancy

The Switch Fabric Module first installed functions as the primary module. For redundancy, you can install a redundant Switch Fabric Module. When two Switch Fabric Modules are installed at the same time, the module in the upper slot functions as the primary module, and the one in the lower slot functions as the backup. If you reset the module installed in the upper slot, the one in the lower slot becomes active.

No configuration is required for Switch Fabric Module redundancy. The module in the upper slot functions as the primary module and a redundant Switch Fabric Module in the lower slot automatically takes over if the primary module fails.

Forwarding Decisions for Layer 3-Switched Traffic

Either a PFC2 or a Distributed Feature Card (DFC) makes the forwarding decision for Layer 3-switched traffic as follows:

- A PFC2 makes all forwarding decisions for each packet that enters the switch through a module without a DFC.
- A DFC makes all forwarding decisions for each packet that enters the switch on a DFC-enabled module in these situations:
 - If the egress port is on the same module as the ingress port, the DFC forwards the packet locally (the packet never leaves the module).
 - If the egress port is on a different fabric-enabled module, the DFC sends the packet across the SFM to the egress module, which sends it out the egress port.
 - If the egress port is on a different nonfabric-enabled module, the DFC sends the packet across the SFM to the Supervisor Engine 2. The Supervisor Engine 2 fabric interface transfers the packet to the 32-Gbps switching bus where it is received by the egress module and is sent out the egress port.

Switching Modes

When you install a Switch Fabric Module, the traffic is forwarded to and from modules in one of the following modes:

- Compact mode—The switch uses this mode for all traffic when only fabric-enabled modules are installed. In this mode, a compact version of the DBus header is forwarded over the switch fabric channel, which provides the best possible performance.

- Truncated mode—The switch uses this mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.
- Bus mode—The switch uses this mode for traffic between nonfabric-enabled modules and for traffic between a nonfabric-enabled module and a fabric-enabled module. In this mode, all traffic passes between the local bus and the supervisor engine bus.

Table 37-1 shows the switch modes used with fabric-enabled and nonfabric-enabled modules installed.

Table 37-1 Switching Modes with Switch Fabric Module Installed

Modules	Switching Modes
Between fabric-enabled modules (when no nonfabric-enabled modules are installed)	Compact ¹
Between fabric-enabled modules (when nonfabric-enabled modules are also installed)	Truncated ²
Between fabric-enabled and nonfabric-enabled modules	Bus
Between non-fabric-enabled modules	Bus

1. In **show** commands, displayed as **dcef** mode for fabric-enabled modules with DFC installed; displayed as **fabric** mode for other fabric-enabled modules.
2. Displayed as **fabric** mode in **show** commands.

Configuring the Switch Fabric Module

These section describe configuring the Switch Fabric Module:

- [Configuring the Switching Mode, page 37-3](#)
- [Configuring Fabric-Required Mode, page 37-4](#)
- [Configuring an LCD Message, page 37-5](#)



Note

With Release 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Configuring the Switching Mode



Note

The commands in this section are supported only with Release 12.1(11b)E and later releases.

To configure the switching mode, perform this task:

Command	Purpose
Router(config)# [no] fabric switching-mode allow { bus-mode { truncated [{ threshold [<i>number</i>]}]}}	Configures the switching mode.

When configuring the switching mode, note the following syntax information:

- To allow use of nonfabric-enabled modules or to allow fabric-enabled modules to use bus mode, enter the **fabric switching-mode allow bus-mode** command.
- To prevent use of nonfabric-enabled modules or to prevent fabric-enabled modules from using bus mode, enter the **no fabric switching-mode allow bus-mode** command.



Caution

When you enter the **no fabric switching-mode allow bus-mode** command, power is removed from any nonfabric-enabled modules installed in the switch.

- To allow fabric-enabled modules to use truncated mode, enter the **fabric switching-mode allow truncated** command.
- To prevent fabric-enabled modules from using truncated mode, enter the **no fabric switching-mode allow truncated** command.
- To configure how many fabric-enabled modules must be installed before they use truncated mode instead of bus mode, enter the **fabric switching-mode allow truncated threshold number** command.
- To return to the default truncated-mode threshold, enter the **no fabric switching-mode allow truncated threshold** command.

Configuring Fabric-Required Mode



Note

The commands in this section are supported only with Release 12.1(11b)E and later.

To configure fabric-required mode, which prevents all switching modules from operating unless there is a Switch Fabric Module installed, perform this task:

Command	Purpose
Router(config)# fabric required	Configures fabric-required mode, which prevents switching modules from operating without a switch-fabric module.
Router(config)# no fabric required	Clears fabric-required mode.



Caution

If you enter the **fabric required** command on a switch that does not have a Switch Fabric Module installed, all modules except the supervisor engine turn off.

When configuring fabric-required mode, note the following syntax information:

- If you boot the switch with fabric-required mode configured but without a Switch Fabric Module installed, only the supervisor engine receives power; no switching modules power up.
- When the switch is operating with fabric-required mode configured and a Switch Fabric Module installed, if you remove the switch fabric module or if it fails, the switch removes power from all switching modules; only the supervisor engine remains active.
- When the switch is operating with fabric-required mode configured and with redundant Switch Fabric Modules installed, if you remove both switch fabric modules or if both fail, the switch removes power from all switching modules; only the supervisor engine remains active.

Configuring an LCD Message

To configure a message for display on the LCD, perform this task:

Command	Purpose
Router(config)# fabric lcd-banner <i>d message d</i>	Configures a message for display on the LCD.
Router(config)# no fabric lcd-banner	Clears the message displayed on the LCD.

When configuring a message for display on the LCD, note the following syntax information:

- The *d* parameter is a delimiting character. You cannot use the delimiting character in the message. The delimiter is a character of your choice—a pound sign (#), for example.
- You can use the following tokens, in the form \$(token), in the message text:
 - \$(hostname)—Displays the switch’s host name.
 - \$(domain)—Displays the switch’s domain name.

Monitoring the Switch Fabric Module

The Switch Fabric Module supports a number of **show** commands for monitoring purposes. A fully automated startup sequence brings the module online and runs the connectivity diagnostics on the ports.

These sections describe how to monitor the Switch Fabric Module:

- [Displaying the Module Information, page 37-5](#)
- [Displaying the Switch Fabric Module Redundancy Status, page 37-6](#)
- [Displaying Fabric Channel Switching Modes, page 37-6](#)
- [Displaying the Fabric Status, page 37-7](#)
- [Displaying the Fabric Utilization, page 37-7](#)
- [Displaying Fabric Errors, page 37-7](#)



Note

The Switch Fabric Module does not require any user configuration.

Displaying the Module Information

To display the module information, perform this task:

Command	Purpose
Router# show module {5 6 7 8}	Displays module information.

This example shows how to display module information:

```
Router# show module 5
Mod Ports Card Type Model Serial No.
-----
 5    0 Switching Fabric Module WS-C6500-SFM SAD04420JR5

Mod MAC addresses Hw Fw Sw Status
-----
 5 0001.0002.0003 to 0001.0002.0003 1.0 6.1(3) 6.2(0.97) Ok
```

Displaying the Switch Fabric Module Redundancy Status

To display the switch fabric module redundancy status, perform this task:

Command	Purpose
Router# show fabric active	Displays switch fabric module redundancy status.

```
Router# show fabric active
Active fabric card in slot 5
No backup fabric card in the system
Router#
```

Displaying Fabric Channel Switching Modes

To display the fabric channel switching mode of one or all modules, perform this task:

Command	Purpose
Router# show fabric switching-mode [module {slot_number all}]	Displays fabric channel switching mode of one or all modules.

This example shows how to display the fabric channel switching mode of all modules:

```
Router# show fabric switching-mode all
bus-only mode is allowed
Module Slot Switching Mode
1 Bus
2 Bus
3 DCEF
4 DCEF
5 No Interfaces
6 DCEF
Router#
```

Displaying the Fabric Status

To display the fabric status of one or all switching modules, perform this task:

Command	Purpose
Router# show fabric status [<i>slot_number</i> all]	Displays fabric status.

This example shows how to display the fabric status of all modules:

```
Router# show fabric status all
  slot      channel      module      fabric
  status
  1          0          OK          OK
  3          0          OK          OK
  3          1          OK          OK
  4          0          OK          OK
Router#
```

Displaying the Fabric Utilization

To display the fabric utilization of one or all modules, perform this task:

Command	Purpose
Router# show fabric utilization [<i>slot_number</i> all]	Displays fabric utilization.

This example shows how to display the fabric utilization of all modules:

```
Router# show fabric utilization all
  slot      channel  Ingress %  Egress %
  1          0          0          0
  3          0          0          0
  3          1          0          0
  4          0          0          0
  4          1          0          0
  6          0          0          0
  6          1          0          0
  7          0          0          0
  7          1          0          0
Router#
```

Displaying Fabric Errors

To display fabric errors of one or all modules, perform this task:

Command	Purpose
Router# show fabric errors [<i>slot_number</i> all]	Displays fabric errors.

This example shows how to display fabric errors on all modules:

```
Router# show fabric errors
  slot    channel  module  module  module  fabric
          channel  crc     hbeat   sync    sync
  1        0        0       0       0       0
  3        0        0       0       0       0
  3        1        0       0       0       0
  4        0        0       0       0       0
  4        1        0       0       0       0
  6        0        0       0       0       0
  6        1        0       0       0       0
  7        0        0       0       0       0
  7        1        0       0       0       0
Router#
```



Power Management and Environmental Monitoring

This chapter describes the power management and environmental monitoring features in the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference* publication.

This chapter consists of these sections:

- [Understanding How Power Management Works, page 38-1](#)
- [Understanding How Environmental Monitoring Works, page 38-4](#)

Understanding How Power Management Works

These sections describe power management in the Catalyst 6500 series switches:

- [Enabling or Disabling Power Redundancy, page 38-2](#)
- [Using the CLI to Power Modules Off and On, page 38-3](#)
- [Using the CLI to View System Power Status, page 38-3](#)
- [Using the CLI to Power Cycle Modules, page 38-4](#)
- [Determining System Power Requirements, page 38-4](#)



Note

In systems with redundant power supplies, both power supplies must be of the same wattage. The Catalyst 6500 series switches allow you to mix AC-input and DC-input power supplies in the same chassis. For detailed information on supported power supply configurations, refer to the *Catalyst 6500 Series Switch Installation Guide*.

The modules have different power requirements, and certain configurations require more power than a single power supply can provide. Although the power management feature allows you to power all installed modules with two power supplies, redundancy is not supported in this configuration. Redundant and nonredundant power configurations are discussed in the following sections.

To determine the power requirements for your system, see the [“Determining System Power Requirements” section on page 38-4](#).

Enabling or Disabling Power Redundancy

From global configuration mode, enter the **power redundancy-mode combined | redundant** commands to disable or enable redundancy (redundancy is enabled by default). You can change the configuration of the power supplies to redundant or nonredundant at any time.

Specifying the **combined** keyword disables redundancy. In a nonredundant configuration, the power available to the system is the combined power capability of both power supplies. The system powers up as many modules as the combined capacity allows. However, if one supply should fail and there is not enough power for all previously powered up modules, the system powers down those modules for which there is not enough power.

Specifying the **redundant** keyword enables redundancy. In a redundant configuration, the total power drawn from both supplies is at no time greater than the capability of one supply. If one supply malfunctions, the other supply can take over the entire system load. When you install and turn on two power supplies, each concurrently provides approximately half of the required power to the system. Load sharing and redundancy are enabled automatically; no software configuration is required.

Enter the **show power** command to view the current state of modules and the total power available for modules (see the “Using the CLI to View System Power Status” section on page 38-3).

Table 38-1 describes how the system responds to changes in the power supply configuration.

Table 38-1 Effects of Power Supply Configuration Changes

Configuration Change	Effect
Redundant to nonredundant	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is increased to the combined power capability of both supplies. Modules marked <i>power-deny</i> in the show power oper state field are brought up if there is sufficient power.
Nonredundant to redundant (both supplies must be of equal wattage)	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is decreased to the power capability of one supply. If there is not enough power for all previously powered-up modules, some modules are powered down and marked as <i>power-deny</i> in the show power oper state field.
Equal wattage power supply is inserted with redundancy enabled	<ul style="list-style-type: none"> System log and syslog messages are generated. System power equals the power capability of one supply. No change in module status since power capability is unchanged.
Equal wattage power supply is inserted with redundancy disabled	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is increased to the combined power capability of both supplies. Modules marked <i>power-deny</i> in the show power oper state field are brought up if there is sufficient power.
Higher or lower wattage power supply is inserted with redundancy enabled	<ul style="list-style-type: none"> System log and syslog messages are generated. The system does not allow you to operate a power supply of different wattage even if the wattage is higher than the installed supply. The inserted supply shuts down.
Higher or lower wattage power supply is inserted with redundancy disabled	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is increased to the combined power capability of both supplies. Modules marked <i>power-deny</i> in the show power oper state field are brought up if there is sufficient power.

Table 38-1 Effects of Power Supply Configuration Changes (continued)

Configuration Change	Effect
Power supply is removed with redundancy enabled	<ul style="list-style-type: none"> System log and syslog messages are generated. No change in module status since power capability is unchanged.
Power supply is removed with redundancy disabled	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is decreased to the power capability of one supply. If there is not enough power for all previously powered-up modules, some modules are powered down and marked as <i>power-deny</i> in the show power oper state field.
System is booted with power supplies of different wattage installed and redundancy enabled	<ul style="list-style-type: none"> System log and syslog messages are generated. The system does not allow you to have power supplies of different wattage installed in a redundant configuration. The lower wattage supply shuts down.
System is booted with power supplies of equal or different wattage installed and redundancy disabled	<ul style="list-style-type: none"> System log and syslog messages are generated. System power equals the combined power capability of both supplies. The system powers up as many modules as the combined capacity allows.

Using the CLI to Power Modules Off and On

You can power down a module from the command-line interface (CLI) by entering the **no power enable module slot** command.



Note

When you enter the **no power enable module slot** command to power down a module, the module's configuration is not saved.

From global configuration mode, enter the **power enable module slot** command to turn the power on for a module that was previously powered down.

Using the CLI to View System Power Status

Enter the **show power** command to view the current power status of system components as follows:

```
Router# show power
system power redundancy mode = redundant
system power total =      1153.32 Watts (27.46 Amps @ 42V)
system power used =       397.74 Watts ( 9.47 Amps @ 42V)
system power available =  755.58 Watts (17.99 Amps @ 42V)
Power-Capacity PS-Fan Output Oper
PS  Type          Watts  A @42V Status Status State
-----
1   WS-CAC-2500W   1153.32 27.46 OK     OK     on
2   none
Pwr-Requested Pwr-Allocated Admin Oper
Slot Card-Type  Watts  A @42V Watts  A @42V State State
-----
1   WS-X6K-SUP2-2GE  142.38 3.39  142.38 3.39  on  on
2   -                -      -    142.38 3.39  -  -
5   WS-X6248-RJ-45  112.98 2.69  112.98 2.69  on  on
Router#
```

Using the CLI to Power Cycle Modules

From global configuration mode, enter the **power cycle module slot** command to power cycle (reset) a module; the module powers off for 5 seconds and then powers on.

Determining System Power Requirements

System power requirements are dependent on the size of the power supply. You could have configuration limitations when using the 1000 W and 1300 W power supplies depending on the size of chassis and type of modules installed. For information about power consumption, refer to the *Release Notes for the Catalyst 6000 Family Switches and Cisco 7600 Internet Router for Cisco IOS* publication at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/relnotes/index.htm>

Understanding How Environmental Monitoring Works

Environmental monitoring of chassis components provides early warning indications of possible component failure to ensure safe and reliable system operation and avoid network interruptions. This section describes the monitoring of these critical system components, enabling you to identify and rapidly correct hardware-related problems in your system.

Using CLI Commands to Monitor System Environmental Status

Enter the **show environment [alarm | status | temperature]** command to display system status information. The keywords display the following:

- **alarm**—Displays environmental alarms
 - **status**—Displays alarm status
 - **thresholds**—Displays alarm thresholds
- **status**—Displays field-replaceable unit (FRU) operational status and power and temperature information
- **temperature**—Displays FRU temperature information

Understanding LED Environmental Indications

The LEDs can indicate two alarm types: major and minor. Major alarms indicate a critical problem that could lead to the system being shut down. Minor alarms are for informational purposes only, giving you notice of a problem that could turn critical if corrective action is not taken.

When the system has an alarm (major or minor), indicating an overtemperature condition, the alarm is not canceled or any action taken (such as module reset or shutdown) for 5 minutes. If the temperature falls 5°C (41°F) below the alarm threshold during this period, the alarm is canceled.

Table 38-2 lists the environmental indicators for the supervisor engine and switching modules.



Note

Refer to the *Catalyst 6500 Series Switch Module Installation Guide* for additional information on LEDs, including the supervisor engine SYSTEM LED.

Table 38-2 Environmental Monitoring for Supervisor Engine and Switching Modules

Component	Alarm Type	LED Indication	Action
Supervisor engine temperature sensor exceeds major threshold ¹	Major	STATUS ² LED red ³	Syslog message and SNMP trap generated. If redundancy, system switches to redundant supervisor engine and the active supervisor engine shuts down. If there is no redundancy and the overtemperature condition is not corrected, the system shuts down after 5 minutes.
Supervisor engine temperature sensor exceeds minor threshold	Minor	STATUS LED orange	Syslog message and SNMP trap generated. Monitor the condition.
Redundant supervisor engine temperature sensor exceeds major or minor threshold	Major	STATUS LED red	Syslog message and SNMP trap generated. If major alarm and the overtemperature condition is not corrected, the system shuts down after 5 minutes.
	Minor	STATUS LED orange	If minor alarm, monitor the condition.
Switching module temperature sensor exceeds major threshold	Major	STATUS LED red	Syslog message and SNMP trap generated. Power down the module ⁴ .
Switching module temperature sensor exceeds minor threshold	Minor	STATUS LED orange	Syslog message and SNMP trap generated. Monitor the condition.

1. Temperature sensors monitor key supervisor engine components including daughter cards.
2. A STATUS LED is located on the supervisor engine front panel and all module front panels.
3. The STATUS LED is red on the failed supervisor engine. If there is no redundant supervisor, the SYSTEM LED is red also.
4. See the “[Understanding How Power Management Works](#)” section on page 38-1 for instructions.



Acronyms

[Table A-1](#) defines the acronyms used in this publication.

Table A-1 *List of Acronyms*

Acronym	Expansion
AAL	ATM adaptation layer
ACE	access control entry
ACL	access control list
AFI	authority and format identifier
Agport	aggregation port
ALPS	Airline Protocol Support
AMP	Active Monitor Present
APaRT	Automated Packet Recognition and Translation
ARP	Address Resolution Protocol
ATA	Analog Telephone Adaptor
ATM	Asynchronous Transfer Mode
AV	attribute value
BDD	binary decision diagrams
BECN	backward explicit congestion notification
BGP	Border Gateway Protocol
BPDU	bridge protocol data unit
BRF	bridge relay function
BSC	Bisync
BSTUN	Block Serial Tunnel
BUS	broadcast and unknown server
BVI	bridge-group virtual interface
CAM	content-addressable memory
CAR	committed access rate
CCA	circuit card assembly
CDP	Cisco Discovery Protocol

Table A-1 List of Acronyms (continued)

Acronym	Expansion
CEF	Cisco Express Forwarding
CHAP	Challenge Handshake Authentication Protocol
CIR	committed information rate
CLI	command-line interface
CLNS	Connection-Less Network Service
CMNS	Connection-Mode Network Service
COPS	Common Open Policy Server
COPS-DS	Common Open Policy Server Differentiated Services
CoS	class of service
CPLD	Complex Programmable Logic Device
CRC	cyclic redundancy check
CRF	concentrator relay function
CST	Common Spanning Tree
CUDD	University of Colorado Decision Diagram
DCC	Data Country Code
dCEF	distributed Cisco Express Forwarding
DDR	dial-on-demand routing
DE	discard eligibility
DEC	Digital Equipment Corporation
DFC	Distributed Forwarding Card
DFI	Domain-Specific Part Format Identifier
DFP	Dynamic Feedback Protocol
DISL	Dynamic Inter-Switch Link
DLC	Data Link Control
DLSw	Data Link Switching
DMP	data movement processor
DNS	Domain Name System
DoD	Department of Defense
DOS	denial of service
dot1q	802.1Q
DRAM	dynamic RAM
DRiP	Dual Ring Protocol
DSAP	destination service access point
DSCP	differentiated services code point
DSPU	downstream SNA Physical Units
DTP	Dynamic Trunking Protocol

Table A-1 List of Acronyms (continued)

Acronym	Expansion
DTR	data terminal ready
DXI	data exchange interface
EAP	Extensible Authentication Protocol
EARL	Enhanced Address Recognition Logic
EEPROM	electrically erasable programmable read-only memory
EHSA	enhanced high system availability
EIA	Electronic Industries Association
ELAN	Emulated Local Area Network
EOBC	Ethernet out-of-band channel
EOF	end of file
ESI	end-system identifier
FAT	File Allocation Table
FECN	forward explicit congestion notification
FM	feature manager
FRU	field replaceable unit
fsck	file system consistency check
FSM	feasible successor metrics
GARP	General Attribute Registration Protocol
GMRP	GARP Multicast Registration Protocol
GVRP	GARP VLAN Registration Protocol
HSRP	Hot Standby Routing Protocol
ICC	Inter-card Communication
ICD	International Code Designator
ICMP	Internet Control Message Protocol
IDB	interface descriptor block
IDP	initial domain part or Internet Datagram Protocol
IDS	Intrusion Detection System Module
IFS	IOS File System
IGMP	Internet Group Management Protocol
IGRP	Interior Gateway Routing Protocol
ILMI	Integrated Local Management Interface
IP	Internet Protocol
IPC	interprocessor communication
IPX	Internetwork Packet Exchange
IS-IS	Intermediate System-to-Intermediate System Intradomain Routing Protocol

Table A-1 List of Acronyms (continued)

Acronym	Expansion
ISL	Inter-Switch Link
ISO	International Organization of Standardization
ISR	Integrated SONET router
LAN	local area network
LANE	LAN Emulation
LAPB	Link Access Procedure, Balanced
LCP	Link Control Protocol
LDA	Local Director Acceleration
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LEM	link error monitor
LER	link error rate
LES	LAN Emulation Server
LLC	Logical Link Control
LTL	Local Target Logic
MAC	Media Access Control
MD5	Message Digest 5
MFD	multicast fast drop
MSFC	Multilayer Switch Feature Card
MIB	Management Information Base
MII	media-independent interface
MLS	Multilayer Switching
MLSE	maintenance loop signaling entity
MOP	Maintenance Operation Protocol
MOTD	message-of-the-day
MLSE	maintenance loops signaling entity
MRM	multicast routing monitor
MSDP	Multicast Source Discovery Protocol
MSFC	Multilayer Switching Feature Card
MSM	Multilayer Switch Module
MTU	maximum transmission unit
MVAP	multiple VLAN access port
NAM	Network Analysis Module
NBP	Name Binding Protocol
NCIA	Native Client Interface Architecture
NDE	NetFlow Data Export

Table A-1 List of Acronyms (continued)

Acronym	Expansion
NET	network entity title
NetBIOS	Network Basic Input/Output System
NFFC	NetFlow Feature Card
NMP	Network Management Processor
NSAP	network service access point
NTP	Network Time Protocol
NVRAM	nonvolatile RAM
OAM	Operation, Administration, and Maintenance
ODM	order dependent merge
OSI	Open System Interconnection
OSM	Optical Services Module
OSPF	open shortest path first
PAE	port access entity
PAgP	Port Aggregation Protocol
PBD	packet buffer daughterboard
PC	Personal Computer (formerly PCMCIA)
PCM	pulse code modulation
PCR	peak cell rate
PDP	policy decision point
PDU	protocol data unit
PEP	policy enforcement point
PFC	Policy Feature Card
PGM	Pragmatic General Multicast
PHY	physical sublayer
PIB	policy information base
PPP	Point-to-Point Protocol
PRID	Policy Rule Identifiers
PVST+	Per VLAN Spanning Tree+
QDM	QoS device manager
QM	QoS manager
QoS	quality of service
RACL	router interface access control list
RADIUS	Remote Access Dial-In User Service
RAM	random-access memory
RCP	Remote Copy Protocol
RGMP	Router-Ports Group Management Protocol

Table A-1 List of Acronyms (continued)

Acronym	Expansion
RIB	routing information base
RIF	Routing Information Field
RMON	remote network monitor
ROM	read-only memory
ROMMON	ROM monitor
RP	route processor or rendezvous point
RPC	remote procedure call
RPF	reverse path forwarding
RSPAN	remote SPAN
RST	reset
RSVP	ReSerVation Protocol
SAID	Security Association Identifier
SAP	service access point
SCM	service connection manager
SCP	Switch-Module Configuration Protocol
SDLC	Synchronous Data Link Control
SGBP	Stack Group Bidding Protocol
SIMM	single in-line memory module
SLB	server load balancing
SLCP	Supervisor Line-Card Processor
SLIP	Serial Line Internet Protocol
SMDS	Software Management and Delivery Systems
SMF	software MAC filter
SMP	Standby Monitor Present
SMRP	Simple Multicast Routing Protocol
SMT	Station Management
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SPAN	Switched Port Analyzer
SREC	S-Record format, Motorola defined format for ROM contents
SSTP	Cisco Shared Spanning Tree
STP	Spanning Tree Protocol
SVC	switched virtual circuit
SVI	switched virtual interface
TACACS+	Terminal Access Controller Access Control System Plus
TARP	Target Identifier Address Resolution Protocol

Table A-1 List of Acronyms (continued)

Acronym	Expansion
TCAM	Ternary Content Addressable Memory
TCL	table contention level
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
TIA	Telecommunications Industry Association
TopN	Utility that allows the user to analyze port traffic by reports
TOS	type of service
TLV	type-length-value
TTL	Time To Live
TVX	valid transmission
UDLD	UniDirectional Link Detection Protocol
UDP	User Datagram Protocol
UNI	User-Network Interface
UTC	Coordinated Universal Time
VACL	VLAN access control list
VCC	virtual channel circuit
VCI	virtual circuit identifier
VCR	Virtual Configuration Register
VINES	Virtual Network System
VLAN	virtual LAN
VMPS	VLAN Membership Policy Server
VPN	virtual private network
VRF	VPN routing and forwarding
VTP	VLAN Trunking Protocol
VVID	voice VLAN ID
WAN	wide area network
WCCP	Web Cache Communications Protocol
WFQ	weighted fair queueing
WRED	weighted random early detection
WRR	weighted round-robin
XNS	Xerox Network System

Numerics

4K VLANs (support for 4,096 VLANs) [2](#)

802.10 SAID (default) [6](#)

802.1Q

encapsulation [4](#)

Layer 2 protocol tunneling

See Layer 2 protocol tunneling

mapping to ISL VLANs [12](#)

trunks [2](#)

restrictions [6](#)

tunneling

configuration guidelines [4](#)

configuring tunnel ports [5](#)

overview [1](#)

802.1Q Ethertype

specifying custom [16](#)

802.1s

See MST

802.1w

See MST

802.1X

See port-based authentication

802.3ad

See LACP

802.3Z Flow Control [14](#)

using with WCCP [10](#)

access port, configuring [14](#)

ACEs and ACLs [1](#)

IPX MLS, flow masks and [3](#)

acronyms, list of [1](#)

addresses

IP, see IP addresses

MAC, see MAC addresses

advertisements, VTP [2](#)

aggregate policing

see QoS policing

aging-time

IP MLS [8,10](#)

IPX MLS [7](#)

alarms

major [4](#)

minor [4](#)

ARP throttling [5](#)

audience [21](#)

authentication

See also port-based authentication

Authentication, Authorization, and Accounting

See also AAA [1](#)

Authentication, Authorization, and Accounting (AAA) [1](#)

authorized ports with 802.1X [4](#)

auto-sync command [3,6,7](#)

auxiliary VLAN

See voice VLAN

A

AAA [1](#)

abbreviating commands [5](#)

access control entries and lists [1](#)

access lists

B

BackboneFast

See STP BackboneFast

blocking state, STP [8](#)

boot bootldr command [26, 27](#)

boot command [22](#)

boot config command [26](#)

BOOTLDR environment variable

- (example) [27](#)
- configuring [27](#)
- description [25](#)
- setting [27](#)

boot loader image [25, 27](#)

boot system command [21, 26](#)

boot system flash command [22](#)

BPDUs guard

- See STP BPDUs guard

bridge ID

- See STP bridge ID

bridge priority, STP [30](#)

bridge protocol data units

- see BPDUs

broadcast storms

- see traffic-storm control

broadcast suppression [1](#)

- enabling [3](#)

C

cache

- IP MLS
 - overview [3](#)
- IPX MLS
 - displaying entries [9](#)
 - overview [2](#)

cache engine clusters [1](#)

cache engines [1](#)

cache farms

- See cache engine clusters

cautions for passwords

- encrypting [17](#)
- TACACS+ [17](#)

CBAC [5](#)

CDP

- configuration task lists [1](#)
- enabling on an interface [2](#)
- monitoring and maintaining [3](#)
- overview [1](#)

cdp enable command [2](#)

CEF [1](#)

- configuring
 - MSFC2 [5](#)
 - supervisor engine [5](#)
- examples [3](#)
- Layer 3 switching [2](#)
- packet rewrite [2](#)

CEF for PFC2

- See CEF

CGMP [1](#)

- multicast router ports
 - specifying [12](#)

channel-group group

- command [8](#)
- command example [8](#)

checking

- configuration, system [10](#)

Cisco Cache Engines [2](#)

Cisco Discovery Protocol

- See CDP

Cisco Group Management Protocol

- See CGMP

Cisco IOS firewall feature set [5](#)

Cisco IOS Unicast Reverse Path Forwarding [19](#)

CiscoView [2](#)

CIST [15](#)

class command [43](#)

class-map command [38](#)

class map configuration [40](#)

clear cdp counters command [3](#)

clear cdp table command [3](#)

clear counters command [18](#)

- clearing IPX MLS cache entries [13](#)
 - clear interface command [19](#)
 - clear ip route command
 - IP MLS restriction [6](#)
 - clear ipx route command
 - IPX MLS restriction [5](#)
 - clear mls command
 - clearing IP MLS cache entries [12](#)
 - clear mls ip multicast statistics command
 - clearing IP MMLS statistics [19](#)
 - clear mls ipx command
 - clearing IPX MLS cache entries [13](#)
 - clear mls nde flow command [17](#)
 - CLI
 - accessing [1](#)
 - backing out one level [5](#)
 - console configuration mode [5](#)
 - getting list of commands [5](#)
 - global configuration mode [5](#)
 - history substitution [3](#)
 - interface configuration mode [5](#)
 - privileged EXEC mode [5](#)
 - ROM monitor [6](#)
 - software basics [4](#)
 - command line processing [3](#)
 - commands, getting list of [5](#)
 - Common and Internal Spanning Tree
 - See also CIST [15](#)
 - Common Spanning Tree
 - See CST
 - community ports [1](#)
 - community VLANs [2](#)
 - CONFIG_FILE environment variable
 - configuration file, viewing [26](#)
 - description [26](#)
 - config-register command [23](#)
 - config terminal command [10](#)
 - configuration
 - file, saving [11](#)
 - interfaces [8 to 10](#)
 - register
 - changing settings [23](#)
 - configuration [21 to 24](#)
 - settings at startup [22](#)
 - configuration register boot field
 - listing value [24](#)
 - modification tasks [23](#)
 - configure command [9](#)
 - configure terminal command [23, 2](#)
 - configuring [1, 5, 42](#)
 - global parameters
 - procedure [3](#)
 - sample configuration [3 to 8](#)
 - interfaces [8 to 9](#)
 - using configuration mode [10](#)
 - congestion avoidance
 - see QoS congestion avoidance
 - console configuration mode [5](#)
 - copy running-config startup-config command [11](#)
 - copy system
 - running-config nvram
 - startup-config command [26](#)
 - CoS
 - override priority [7, 8](#)
 - counters
 - clearing interface [18, 19](#)
 - CST [15](#)
 - common spanning tree [18](#)
-
- ## D
- dCEF [4, 5](#)
 - debug commands
 - IP MMLS [18](#)
 - debugging mls comand [14](#)
 - default configuration
 - 802.1X [5](#)
 - IP MLS [6](#)

- IP MMLS 7
- IPX MLS 5
- supervisor engine 1
- UDLD 3
- voice VLAN 4
- VTP 5
- default gateway, configuring 12
- default NDE configuration 7
- default VLAN 10
- denial of service protection 1
- description command 16
- destination flow mask 3
- destination-ip flow mask 3,2
- destination-source flow mask 3
- destination-source-ip flow mask 2
- differentiated services codepoint
 - See QoS DSCP
- dir command 27
- disabling
 - IP MLS
 - on router interfaces 7
 - IPX MLS
 - on switch interfaces 6
- displaying IPX MLS VLAN statistics 12
- distributed Cisco Express Forwarding
 - See dCEF
- documentation, related 23
- document organization 21
- drop thresholds
 - see QoS congestion avoidance
- DSCP
 - See QoS DSCP
- duplex command 8,9
- duplex mode
 - configuring interface 7
- redundancy 1
- Embedded CiscoView 2
- enable command 10,23
- enable mode 5
- enabling
 - IP MLS, on router interfaces 7
 - IP MMLS
 - on router interfaces 11
 - IPX MLS, on router interfaces 6
- enabling IP MLS 6
- encapsulation 4
- enhanced high system availability
 - See EHSA
- environmental monitoring
 - LED indications 4
 - SNMP traps 4
 - supervisor engine and switching modules 4
 - Syslog messages 4
 - using CLI commands 4
- environment variables
 - BOOTLDR 25
 - (example) 27
 - setting 27
 - CONFIG_FILE 26
 - controlling 26
 - viewing 26
- erase startup-config command
 - configuration files cleared with 14
- EtherChannel
 - channel-group group
 - command 8
 - command example 8
 - configuration guidelines 5
 - configuring
 - Layer 2 8
 - configuring (tasks) 6
 - DFC restriction, see CSCdt27074 in the Release Notes
 - interface port-channel
 - command example 7

E

EHSA

- interface port-channel (command) [7](#)
 - lACP system-priority
 - command example [10](#)
 - Layer 2
 - configuring [8](#)
 - load balancing
 - configuring [11](#)
 - understanding [5](#)
 - modes [3](#)
 - PAgP
 - Understanding [3](#)
 - port-channel interfaces [5](#)
 - port-channel load-balance
 - command [10,11](#)
 - command example [11](#)
 - STP [5](#)
 - understanding [1](#)
 - EtherChannel Guard
 - See STP EtherChannel Guard
 - Ethernet
 - setting port duplex [15](#)
 - examples
 - configuration
 - interface [8 to 9](#)
 - software configuration register [21 to 24](#)
 - configuring global parameters [3](#)
 - extended range VLANs [2](#)
 - See VLANs
 - Extensible Authentication Protocol over LAN [1](#)
-
- F**
- fabric switching mode
 - See switch fabric module
 - fastEthernet [2](#)
 - fiber-optic, detecting unidirectional links [1](#)
 - filters
 - protocol
 - See protocol filtering
 - filters, NDE
 - clearing [17](#)
 - destination host filter, specifying [16](#)
 - destination TCP/UDP port, specifying [15](#)
 - overview [6](#)
 - protocol [16](#)
 - source host and destination TCP/UDP port [16](#)
 - firewall [5](#)
 - Flash memory
 - configuration process [25](#)
 - configuring router to boot from [25](#)
 - loading system image from [24](#)
 - security precautions [25](#)
 - write protection [25](#)
 - flow control [14](#)
 - flow masks
 - IP MLS
 - destination-ip [3,2](#)
 - destination-source-ip [2](#)
 - interface-destination-source-ip [2](#)
 - ip-flow [3](#)
 - ip-full [3](#)
 - ip-interface-full [3](#)
 - source-destination-ip [3](#)
 - IP MMLS and [3](#)
 - IPX MLS
 - destination [3](#)
 - destination-source [3](#)
 - IPX MLS entries and [3](#)
 - minimum [8,9](#)
 - modes [3](#)
 - overview [3,2](#)
 - flows
 - IP MLS [2](#)
 - IP MMLS
 - completely and partially switched [4](#)
 - IPX MLS [2](#)
 - forward-delay time, STP [32](#)
 - forwarding information base [5](#)

frame distribution

See EtherChannel load balancing

G

gateway, configuring 12

global configuration mode 5

global parameters, configuring 3

H

hardware Layer 3 switching

guidelines 4

hello time, STP 32

history

CLI 3

I

I-BPDU 16

ICMP unreachable messages 1

IEEE 802.10 SAID (default) 6

IEEE 802.1Q

See 802.1Q

IEEE 802.1Q Ethertype

specifying custom 16

IEEE 802.1s

See MST

IEEE 802.1w

See MST

IEEE 802.3ad

See LACP

IEEE 802.3Z Flow Control 14

IGMP

configuration guidelines 6

enabling 9

Internet Group Management Protocol 1

join messages 2

leave processing

enabling 11

queries 3

query interval

configuring 11

snooping

fast leave 5

joining multicast group 2

leaving multicast group 4

understanding 2

snooping querier

enabling 7

guidelines and restrictions 7

understanding 2

IGMPv3 10

IGMP v3lite 10

IGRP, configuring 7

interface

command 10

configuration 8 to 9

configuration mode 5

Layer 2 modes 4

number 2

parameters, configuring 8

interface-destination-source-ip flow mask 2

interface port-channel

command example 7

interface port-channel (command) 7

interfaces

configuring 2

configuring, duplex mode 7

configuring, speed 7

configuring, overview 1

counters, clearing 18, 19

descriptive name, adding 16

displaying information about 17

maintaining 17

monitoring 17

naming 16

- range of [4](#)
- restarting [19](#)
- shutting down
 - task [19](#)
- interfaces command [1,2](#)
- interfaces range command [4](#)
- interfaces range macro command [6](#)
- Interior Gateway Routing Protocol
 - See IGRP, configuring
- Internal Sub Tree Protocol
 - See ISTP [15](#)
- Internet Group Management Protocol
 - See IGMP
- IP
 - default gateway, configuring [12](#)
 - static routes [12](#)
- IP accounting, IP MMLS and [9](#)
- IP addresses
 - assigned by BOOTP protocol [14](#)
 - set to default [14](#)
- IP CEF
 - topology (figure) [3](#)
- ip flow-export destination command [14](#)
- ip flow-export source command [13](#)
- ip-flow flow mask [3](#)
- ip-full flow mask [3](#)
- ip http server [1](#)
- ip-interface-full flow mask [3](#)
- IP MLS
 - aging-time [8,10](#)
 - cache
 - clearing entries [11](#)
 - overview [3](#)
 - cache, displaying
 - by destination address [10](#)
 - by source address [10](#)
 - by specific flow [11](#)
 - cache entries, displaying [9](#)
 - debugging, on switch [14](#)
 - default configuration [6](#)
 - disabling
 - on router interface [7](#)
 - displaying interface configuration [7](#)
 - displaying IP MLS VLAN statistics [13](#)
 - enabling [6](#)
 - enabling on router interface [7](#)
 - flow masks
 - destination-ip [3,2](#)
 - destination-source-ip [2](#)
 - interface-destination-source-ip [2](#)
 - ip flow [3](#)
 - ip-full [3](#)
 - ip-interface-full [3](#)
 - minimum [8,9](#)
 - overview [3,2](#)
 - source-destination-ip [3](#)
- flows [2](#)
- NDE
 - See NDE
- operational overview [5](#)
- overview [2](#)
- packet rewrite [4](#)
- router
 - disabling on interfaces [7](#)
 - enabling on interfaces [7](#)
- statistics
 - displaying for MLS cache entries [12](#)
 - displaying IP MLS statistics and contention tables [12](#)
- troubleshooting [14](#)
- IP MMLS
 - cache, overview [2](#)
 - configuration guideline [8](#)
 - debug commands [18](#)
 - default configuration [7](#)
 - enabling
 - on router interfaces [11](#)
 - flow mask [3](#)
 - flows

- completely and partially switched 4
- Layer 3 MLS cache 2
- overview 2
- packet rewrite 3
- router
 - displaying interface information 14
 - enabling globally 10
 - enabling on interfaces 11
 - multicast routing table, displaying 16
 - PIM, enabling 10
- switch
 - statistics, clearing 19
- unsupported features 9
- IP multicast
 - IGMP snooping and 8
 - overview 1
- IP multicast MLS
 - See IP MMLS
- ip multicast-routing command
 - enabling IP multicast 10
- IP phone
 - configuring 5
- ip pim command
 - enabling IP PIM 10
- ip routing command
 - IP MLS restriction 6
- IPsec 6
- ip security command
 - IP MLS restriction 6
- ip wccp version command 7
- IPX MLS
 - access lists, flow masks and 3
 - aging-time 7
 - cache
 - overview 2
 - cache, displaying
 - all entries 9
 - by destination address 10
 - by source address 10
 - cache entries, displaying 9
 - clearing cache entries 13
 - configuration guidelines
 - interaction with other features 5
 - MTU 6
 - default configuration 5
 - disabling
 - on router interface 6
 - disabling on interfaces 6
 - displaying VLAN statistics 12
 - enabling 6
 - enabling on interface 6
 - enabling on interfaces 6
 - flow masks
 - access lists and 3
 - destination 3
 - destination-source 3
 - IPX MLS entries and 3
 - minimum 8
 - modes 3
 - overview 3
 - flows 2
 - operational overview 4
 - overview 2
 - packet rewrite 3
 - statistics
 - displaying for MLS cache entries 11
- ipx routing command
 - IPX MLS restriction 6
- ipx security command
 - IPX MLS restriction 6
- ISL encapsulation 4
- ISL trunks 2
- isolated port 1
- isolated VLANs 2
- ISTP 15

J

join messages, IGMP [2](#)
 jumbo frames [10](#)

K

keyboard shortcuts [3](#)

L

LACP

system ID [4](#)

Layer 2

configuring interfaces [7](#)

access port [14](#)

trunk [8](#)

defaults [5](#)

interface modes [4](#)

show interfaces [13, 14, 7, 13](#)

switching

understanding [1](#)

trunks

understanding [2](#)

VLAN

interface assignment [12](#)

Layer 2 Interfaces

configuring [1](#)

Layer 2 protocol tunneling

configuring Layer 2 tunnels [8](#)

overview [7](#)

Layer 3

IP MMLS and MLS cache [2](#)

shortcuts

See IP MLS, IP MMLS, or IPX MLS

Layer 3 switched packet rewrite

CEF [2](#)

Layer 3 switching

CEF [2](#)

Layer 4 port operations (ACLs) [3](#)

leave processing, IGMP

enabling [11](#)

link negotiation [8](#)

load balancing [15](#)

logical operation unit

See LOU

loop guard

See STP loop guard

LOU

description [4](#)

determining maximum number of [4](#)

M

MAC address

adding to BOOTP configuration file [14](#)

MAC address-based blocking [8](#)

mac move notification

configuring [22](#)

main-cpu command [3, 6, 7](#)

mapping 802.1Q VLANs to ISL VLANs [12](#)

markdown

see QoS markdown

marking

see QoS

match protocol [31](#)

maximum aging time, STP [33](#)

microflow policing rule

see QoS policing

MLS

configuring [6](#)

configuring threshold [11](#)

MSFC

threshold [11](#)

mls aging command

configuring IP MLS [11](#)

mls flow command

configuring IP MLS [8, 9, 10](#)

- mls flow ipx command
 - configuring IPX MLS flow mask [8](#)
 - mls ip multicast command
 - enabling IP MMLS [11, 12, 13, 14](#)
 - mls ipx command
 - enabling IPX MLS on a router interface [6](#)
 - mls nde flow command
 - configuring a host and port filter [16](#)
 - configuring a host flow filter [16](#)
 - configuring a port filter [15](#)
 - configuring a protocol flow filter [16](#)
 - mls nde src_address command [8](#)
 - monitoring
 - traffic suppression [5](#)
 - MST [15](#)
 - boundary ports [19](#)
 - configuration [18](#)
 - configuring [34](#)
 - edge ports [20](#)
 - enabling [34](#)
 - hop count [20](#)
 - instances [18](#)
 - interoperability [16](#)
 - interoperability with PVST+ [16](#)
 - link type [20](#)
 - master [19](#)
 - message age [20](#)
 - regions [18, 19](#)
 - MSTP
 - M-record [16](#)
 - M-tree [16](#)
 - MTU
 - IPX MLS and [6](#)
 - MTU size (default) [6](#)
 - multicast
 - broadcast suppression [3](#)
 - IGMP snooping and [8](#)
 - NetFlow statistics [1](#)
 - non-RPF [5](#)
 - overview [1](#)
 - RGMP [1](#)
 - router, specifying port for [12](#)
 - multicast, displaying routing table [16](#)
 - multicast groups
 - joining [2](#)
 - leaving [4](#)
 - multicast multilayer switching
 - See IP MMLS
 - multicast RPF [2](#)
 - multicast storms
 - see traffic-storm control
 - multilayer switch feature card
 - see MSFC
 - Multilayer Switching
 - See IP MLS, IP MMLS, or IPX MLS
 - multiple forwarding paths [15](#)
 - Multiple Spanning Tree
 - See MST
 - Multiple Spanning Tree Protocol
 - See MSTP [15](#)
-
- ## N
- native vlan [11](#)
 - NBAR [1, 31](#)
 - NDE
 - configuration, displaying [17](#)
 - displaying configuration [17](#)
 - enabling [8](#)
 - filters
 - clearing [17](#)
 - destination host, specifying [16](#)
 - destination TCP/UDP port, specifying [15](#)
 - overview [6](#)
 - protocol, specifying [16](#)
 - source host and destination TCP/UDP port, specifying [16](#)
 - multicast [1](#)

- overview [1](#)
- specifying
 - destination host filters [16](#)
 - destination TCP/UDP port filters [16](#)
 - protocol filters [16](#)
- NDE configuration, default [7](#)
- NetFlow Data Export
 - See NDE
- Network-Based Application Recognition [1](#)
- network fault tolerance [15](#)
- network management
 - configuring [1](#)
- non-RPF multicast [5](#)
- nonvolatile random-access memory
 - See NVRAM
- normal-range VLANs
 - See VLANs
- NVRAM
 - saving settings [11](#)

O

- OIR [17](#)
- online insertion and removal
 - See OIR
- operating system image
 - See system image
- out of profile
 - see QoS out of profile

P

- packet rewrite
 - CEF [2](#)
 - IP MLS and [4](#)
 - IP MMLS and [3](#)
 - IPX MLS and [3](#)
- packets

- multicast [11](#)
- PAgP
 - understanding [3](#)
- passwords
 - configuring
 - enable password [15](#)
 - enable secret [15](#)
 - line password [16](#)
 - static enable password [15](#)
 - TACACS+ [16](#)
 - TACACS+ (caution) [17](#)
 - encrypting [17](#)
 - (caution) [17](#)
 - recovering lost enable passwords [19](#)
- PBR [2](#)
- PFC2
 - NetFlow
 - table, displaying entries [6](#)
- PIM, IP MMLS and [10](#)
- police command [46](#)
- policing
 - See QoS policing
- policy [38](#)
- policy-based routing
 - See PBR
- Policy Feature Card
 - See PFC
- policy map [42](#)
 - attaching to an interface [49](#)
- policy-map command [38, 42](#)
- Port Aggregation Protocol
 - see PAgP
- port-based authentication
 - authentication server
 - defined [2](#)
 - RADIUS server [2](#)
 - client, defined [2](#)
 - configuration guidelines [6](#)
 - configuring

- initializing authentication of a client **11**
- manual reauthentication of a client **11**
- quiet period **11**
- RADIUS server **10**
- RADIUS server parameters on the switch **8**
- switch-to-authentication-server retransmission time **13**
- switch-to-client EAP-request frame retransmission time **13**
- switch-to-client frame-retransmission number **14**
- switch-to-client retransmission time **12**
- default configuration **5**
- described **1**
- device roles **2**
- displaying statistics **15**
- EAPOL-start frame **3**
- EAP-request/identity frame **3**
- EAP-response/identity frame **3**
- enabling
 - 802.1X authentication **7,8**
 - periodic reauthentication **10**
- encapsulation **2**
- initiation and message exchange **3**
- method lists **7**
- ports
 - authorization state and dot1x port-control command **4**
 - authorized and unauthorized **4**
- resetting to default values **15**
- switch
 - as proxy **2**
 - RADIUS client **2**
 - topologies, supported **4**
- port-based QoS features
 - see QoS
- port-channel
 - see EtherChannel
- port-channel load-balance
 - command **10,11**
 - command example **10,11**
- port cost, STP **29**
- port debounce timer
 - disabling **15**
 - displaying **15**
 - enabling **15**
- PortFast
 - See STP PortFast
- PortFast BPDU filtering
 - See STP PortFast BPDU filtering
- port negotiation **8**
- port priority, STP **27**
- ports
 - secure **1**
 - setting the debounce timer **15**
- port security
 - aging **4**
 - configuring **3**
 - default configuration **2**
 - described **1**
 - displaying **6**
 - violations **2**
 - with other features **2**
- power management
 - enabling/disabling redundancy **2**
 - overview **1**
 - powering modules up or down **3**
 - system power requirements, nine-slot chassis **4**
- primary VLANs **2**
- priority
 - overriding CoS **7,8**
- private VLANs **1**
 - community VLANs **2**
 - configuration guidelines **2**
 - configuring **5**
 - host ports **8**
 - promiscuous ports **9**
 - routing secondary VLAN ingress traffic **7**
 - secondary VLANs with primary VLANs **6**
 - VLANs as private **5**

- isolated VLANs [2](#)
 - ports
 - community [1](#)
 - isolated [1](#)
 - promiscuous [1](#)
 - primary VLANs [2](#)
 - secondary VLANs [2](#)
 - trunks [2](#)
 - privileged EXEC mode [5](#)
 - privileges
 - changing default [18](#)
 - configuring
 - multiple levels [17](#)
 - privilege level [18](#)
 - exiting [19](#)
 - logging in [18](#)
 - procedures
 - global parameters, configuring [3 to 8](#)
 - interfaces, configuring [8 to 9](#)
 - using configuration mode [10](#)
 - promiscuous ports [1](#)
 - protocol filtering
 - configuring [3](#)
 - overview [1](#)
 - protocol tunneling
 - See Layer 2 protocol tunneling [7](#)
 - pruning, VTP
 - See VTP, pruning
 - PVLANS
 - See private VLANs
 - PVRST
 - See Rapid-PVST [14](#)
-
- Q**
- QoS ACL [18](#)
 - attaching [21](#)
 - QoS classification (definition) [4](#)
 - QoS congestion avoidance
 - definition [4](#)
 - receive queue [14](#)
 - QoS CoS
 - and ToS final L3 Switching Engine values [21](#)
 - and ToS final values from L3 Switching Engine [21](#)
 - definition [3](#)
 - port value, configuring [54](#)
 - QoS default configuration [25](#)
 - QoS definitions [3](#)
 - QoS drop thresholds
 - see QoS congestion avoidance
 - QoS DSCP
 - definition [4](#)
 - internal values [17](#)
 - maps, configuring [66](#)
 - QoS dual transmit queue
 - thresholds
 - configuring [58](#)
 - QoS Ethernet egress port
 - feature summary [12](#)
 - scheduling [22](#)
 - scheduling, congestion avoidance, and marking [10, 21](#)
 - QoS Ethernet ingress port
 - classification, marking, scheduling, and congestion avoidance [7](#)
 - feature summary [11](#)
 - marking, scheduling, congestion avoidance, and classification [12](#)
 - scheduling [14](#)
 - scheduling and congestion avoidance [13](#)
 - QoS feature set summary [11](#)
 - QoS final L3 Switching Engine CoS and ToS values [21](#)
 - QoS internal DSCP values [17](#)
 - QoS L3 Switching Engine
 - classification, marking, and policing [8, 16](#)
 - feature summary [11](#)
 - QoS labels (definition) [3](#)
 - QoS mapping
 - CoS values to DSCP values [66](#)

- DSCP markdown values [25, 68](#)
 - DSCP values to CoS values [67](#)
 - IP precedence values to DSCP values [67](#)
 - QoS markdown [19](#)
 - QoS marking [24](#)
 - definition [4](#)
 - trusted ports [13](#)
 - untrusted ports [13](#)
 - QoS MSFC
 - marking [9](#)
 - QoS multilayer switch feature card [12](#)
 - QoS OSM egress port
 - feature summary [12](#)
 - QoS OSM ingress port
 - feature summary [11](#)
 - QoS out of profile [19](#)
 - QoS policing
 - definition [4](#)
 - microflow, enabling for nonrouted traffic [50](#)
 - QoS policing rule [19](#)
 - aggregate [19](#)
 - creating [37](#)
 - microflow [19](#)
 - QoS port
 - trust state [53](#)
 - QoS port-based or VLAN-based [51, 52](#)
 - QoS port keywords [12](#)
 - QoS queues
 - transmit, allocating bandwidth between [64](#)
 - QoS receive queue [13, 61](#)
 - drop thresholds [16](#)
 - QoS scheduling (definition) [4](#)
 - QoS single-receive, dual-transmit queue ports
 - configuring [59](#)
 - QoS statistics data export [24](#)
 - configuring [70](#)
 - configuring destination host [75](#)
 - configuring time interval [74, 77](#)
 - QoS strict priority receive queue [14](#)
 - QoS ToS
 - and CoS final values from L3 Switching Engine [21](#)
 - definition [4](#)
 - QoS traffic flow through QoS features [6](#)
 - QoS transmit queue
 - size ratio [64, 65](#)
 - thresholds
 - configuring [54](#)
 - QoS transmit queues [21, 60, 61](#)
 - QoS trust-cos
 - port keyword [11, 12](#)
 - QoS trust-dscp
 - port keyword [11, 12](#)
 - QoS trust-ipprec
 - port keyword [11, 12](#)
 - QoS untrusted port keyword [11, 12](#)
 - QoS VLAN-based or port-based [21, 51, 52](#)
 - queries, IGMP [3](#)
-
- ## R
- range
 - command [4](#)
 - macro [6](#)
 - of interfaces [4](#)
 - Rapid-PVST
 - enabling [33](#)
 - overview [14](#)
 - Rapid Spanning Tree
 - See RSTP [13](#)
 - rapid spanning tree protocol [15](#)
 - rate-limiting [5](#)
 - receive queues
 - see QoS receive queues
 - reduced MAC address [3](#)
 - redundancy
 - configuring supervisor engine [2](#)
 - displaying supervisor engine configuration [4](#)
 - EHSA [1](#)

redundancy (RPR+) [1](#)
 configuring [6](#)
 configuring supervisor engine [5](#)
 displaying supervisor engine configuration [8](#)
 redundancy command [6,7](#)
 route processor redundancy plus [2](#)
 redundancy command [3](#)
 related documentation [23](#)
 reload command [23](#)
 reserved-range VLANs
 See VLANs
 rewrite, packet
 CEF [2](#)
 IP MLS [4](#)
 IP MMLS [3](#)
 IPX MLS [3](#)
 RGMP [1](#)
 overview [1](#)
 packet types [2](#)
 RIF cache monitoring [18](#)
 rommon command [24](#)
 ROM monitor
 boot process and [20](#)
 CLI [6](#)
 root bridge, STP [25](#)
 root guard
 See STP root guard
 route processor redundancy
 See redundancy (RPR+)
 router, multicast [12](#)
 router-port group management protocol
 See RGMP
 routing table, multicast [16](#)
 RPF
 failure [5](#)
 multicast [2](#)
 non-RPF multicast [5](#)
 unicast [19](#)
 RPR+

See redundancy (RPR+)

RSTP [15](#)
 port roles [13](#)
 port states [14](#)

S

SAID [6](#)
 sample configuration [2 to 10](#)
 Sampled NetFlow
 description [6](#)
 saving the configuration file [11](#)
 scheduling
 see QoS
 secondary VLANs [2](#)
 secure ports, configuring [1](#)
 security
 configuring [1](#)
 security, port [1](#)
 security precautions with Flash memory card [25](#)
 serial interfaces
 clearing [19](#)
 synchronous
 maintaining [19](#)
 service-policy command [38](#)
 service-policy input command [49](#)
 set power redundancy enable/disable command [2](#)
 setup command [2](#)
 shortcuts, Layer 3
 See IP MLS, IP MMLS, or IPX MLS
 show boot command [26](#)
 show bootvar command [27](#)
 show catalyst6000 chassis-mac-address command [3](#)
 show cdp command [2,3](#)
 show cdp entry command [3](#)
 show cdp interface command [3](#)
 show cdp neighbors command [3](#)
 show cdp traffic command [3](#)
 show ciscoview package command [3](#)

- show ciscoview version command [3](#)
- show configuration command [16](#)
- show debugging command [3](#)
- show eobc command [18](#)
- show hardware command [3](#)
- show history command [4](#)
- show ibc command [18](#)
- show interfaces command [2, 13, 14, 16, 18, 7, 13](#)
 - clearing interface counters [18](#)
 - displaying, interface type numbers [2](#)
 - displaying, speed and duplex mode [9](#)
- show ip flow export command
 - displaying NDE export flow IP address and UDP port [14](#)
- show ip interface command
 - displaying IP MLS interface configuration [7](#)
 - displaying IP MMLS interfaces [15](#)
- show ip mroute command
 - displaying IP multicast routing table [16](#)
- show ip pim interface command
 - displaying IP MMLS router configuration [15](#)
- show ipx interface command
 - display IPX MLS interface configuration [7](#)
- show mls aging command [11](#)
- show mls entry command [6](#)
- show mls ip command
 - displaying MSFC information [9](#)
- show mls ip destination
 - displaying IP MLS destination address [10](#)
- show mls ip flow command
 - displaying IP MLS cache entries [11](#)
- show mls ip multicast group command
 - displaying IP MMLS group [17](#)
- show mls ip multicast interface command
 - displaying IP MMLS interface [17](#)
- show mls ip multicast source command
 - displaying IP MMLS source [17](#)
- show mls ip multicast statistics command
 - displaying IP MMLS statistics [17](#)
- show mls ip multicast summary
 - displaying IP MMLS configuration [17](#)
- show mls ip source command
 - displaying IP MLS source address [10](#)
- show mls ipx command
 - displaying IPX MLS cache entries [9](#)
- show mls nde command [17](#)
 - displaying NDE flow IP address [14](#)
- show mls rp command
 - displaying IP MLS configuration [9](#)
 - display IPX MLS flow mask [8](#)
- show mls table-contention command
 - displaying IP MLS contention tables [13](#)
 - displaying IPX MLS contention table [12](#)
- show mls vlan-statistics command
 - displays IPX MLS VLAN statistics [12](#)
- show module command [4, 8](#)
- show protocols command [18](#)
- show rif command [18](#)
- show running-config command [10, 16, 18](#)
- show startup-config command [11](#)
- show version command [9, 23, 24, 18](#)
- shutdown command [19](#)
- shutdown interfaces
 - result [19](#)
- Single Spanning Tree
 - See SST [15](#)
- slot number, description [1](#)
- SNMP
 - support and documentation [1](#)
- snooping
 - See IGMP snooping
- software configuration register functions [21 to 24](#)
- source-destination-ip flow mask [3](#)
- source-destination-vlan flow mask [3](#)
- source specific multicast with IGMPv3, IGMP v3lite, and URD [10](#)
- SPAN
 - configuration guidelines [5](#)

- configuring [8](#)
 - destinations [10](#)
 - sources [9](#)
 - VLAN filtering [10](#)
- overview [1](#)
- spanning-tree backbonefast
 - command [13, 14](#)
 - command example [13, 14](#)
- spanning-tree cost
 - command [29](#)
 - command example [29, 30](#)
- spanning-tree portfast
 - command [8, 9](#)
 - command example [8](#)
- spanning-tree portfast bpduguard
 - command [11](#)
- spanning-tree port-priority
 - command [27, 28](#)
- spanning-tree uplinkfast
 - command [12](#)
 - command example [12](#)
- spanning-tree vlan
 - command [23, 24, 26, 27, 14](#)
 - command example [23, 25, 26, 27](#)
- spanning-tree vlan cost
 - command [29](#)
- spanning-tree vlan forward-time
 - command [32](#)
 - command example [33](#)
- spanning-tree vlan hello-time
 - command [32](#)
 - command example [32](#)
- spanning-tree vlan max-age
 - command [33](#)
 - command example [33](#)
- spanning-tree vlan port-priority
 - command [27](#)
 - command example [28](#)
- spanning-tree vlan priority
 - command [31](#)
 - command example [31](#)
- speed
 - configuring interface [7](#)
- speed command [8](#)
- SST [15](#)
 - interoperability [16](#)
- static route, configuring [12](#)
- statistics
 - 802.1X [15](#)
- storm control
 - see traffic-storm control
- STP
 - configuring [22](#)
 - bridge priority [30](#)
 - enabling [23, 24](#)
 - forward-delay time [32](#)
 - hello time [32](#)
 - maximum aging time [33](#)
 - port cost [29](#)
 - port priority [27](#)
 - root bridge [25](#)
 - secondary root switch [26](#)
 - defaults [21](#)
 - EtherChannel [5](#)
 - understanding [2](#)
 - 802.1Q Trunks [12](#)
 - Blocking State [8](#)
 - BPDUs [4](#)
 - disabled state [12](#)
 - forwarding state [11](#)
 - learning state [10](#)
 - listening state [9](#)
 - overview [2](#)
 - port states [6](#)
 - protocol timers [5](#)
 - root bridge election [4](#)
 - topology [5](#)
- STP BackboneFast

- and MST [16](#)
- configuring [13](#)
- figure
 - adding a switch [7](#)
- spanning-tree backbonefast
 - command [13, 14](#)
 - command example [13, 14](#)
- understanding [4](#)
- STP BPDU Guard
 - and MST [16](#)
 - configuring [11](#)
 - spanning-tree portfast bpdu-guard
 - command [11](#)
 - understanding [2](#)
- STP bridge ID [3](#)
- STP EtherChannel guard [6](#)
- STP loop guard
 - and MST [16](#)
 - configuring [15](#)
 - overview [6](#)
- STP PortFast
 - and MST [16](#)
 - BPDU filter
 - configuring [10](#)
 - BPDU filtering [2](#)
 - configuring [8](#)
 - spanning-tree portfast
 - command [8, 9](#)
 - command example [8](#)
 - understanding [2](#)
- STP Portfast BPDU filtering
 - and MST [16](#)
- STP root guard [6, 14](#)
 - and MST [16](#)
- STP UplinkFast
 - and MST [16](#)
 - configuring [12](#)
 - spanning-tree uplinkfast
 - command [12](#)
 - command example [12](#)
 - understanding [3](#)
 - strict-priority queue
 - see QoS strict priority
 - supervisor engine
 - configuring [1](#)
 - default configuration [1](#)
 - default gateways [12](#)
 - environmental monitoring [4](#)
 - redundancy [1](#)
 - ROM monitor [20](#)
 - startup configuration [20](#)
 - static routes [12](#)
 - synchronizing configurations [3, 7](#)
 - supervisor engine redundancy
 - configuring [2, 5](#)
 - supervisor engines
 - displaying redundancy configuration [4, 8](#)
- Switched Port Analyzer
 - See SPAN
- switch fabric module [1](#)
 - configuring [3](#)
 - monitoring [5](#)
 - slot locations [2](#)
- switchport
 - configuring [14](#)
 - example [13](#)
 - show interfaces [13, 14, 7, 13](#)
- switchport access vlan [10, 14](#)
 - example [15](#)
- switchport mode access [4, 14](#)
 - example [15](#)
- switchport mode dynamic [9](#)
- switchport mode dynamic auto [4](#)
- switchport mode dynamic desirable [4](#)
 - default [5](#)
 - example [13](#)
- switchport mode trunk [4, 9](#)
- switchport nonegotiate [4](#)

- switchport trunk allowed vlan [11](#)
- switchport trunk encapsulation [9](#)
- switchport trunk encapsulation dot1q [4](#)
 - example [13](#)
- switchport trunk encapsulation isl [4](#)
- switchport trunk encapsulation negotiate [4](#)
 - default [5](#)
- switchport trunk native vlan [11](#)
- switchport trunk pruning vlan [12](#)
- system
 - configuration register
 - configuration [21 to 24](#)
 - settings at startup [22](#)
 - configuring global parameters [3 to 8](#)
- system image
 - determining if and how to load [22](#)
 - loading from Flash [24](#)

T

- TACACS+ [1](#)
- TCP Intercept [18](#)
- Telnet
 - accessing CLI [2](#)
- thresholds
 - see QoS congestion avoidance
- traffic-storm control
 - command
 - broadcast [2](#)
 - described [1](#)
 - monitoring [4](#)
 - thresholds [1](#)
- traffic suppression
 - see traffic-storm control
- translational bridge numbers (defaults) [6](#)
- transmit queues
 - see QoS transmit queues
- troubleshooting
 - IP MLS [14](#)

- trunks [2](#)
 - 802.1Q Restrictions [6](#)
 - allowed VLANs [11](#)
 - configuring [8](#)
 - default interface configuration [7](#)
 - default VLAN [10](#)
 - different VTP domains [3](#)
 - encapsulation [4](#)
 - native vlan [11](#)
 - private VLANs [2](#)
 - to non-DTP device [5](#)
 - VLAN 1 minimization [12](#)
- trust-dscp
 - see QoS trust-dscp
- trust-ipprec
 - see QoS trust-ipprec
- tunneling, 802.1Q
 - See 802.1Q [1](#)

U

- UDLD
 - default configuration [3](#)
 - enabling
 - globally [3](#)
 - on ports [4, 5](#)
 - overview [1](#)
- unauthorized ports with 802.1X [4](#)
- unicast flood protection [21](#)
 - configuring [21](#)
- unicast RPF [19](#)
- unicast storms
 - see traffic-storm control
- UniDirectional Link Detection Protocol
 - see UDLD
- untrusted
 - see QoS trust-cos
 - see QoS untrusted
- UplinkFast

See STP UplinkFast

URD [10](#)

user EXEC mode [5](#)

V

VACLs [8](#)

configuring [11](#)

examples [16](#)

Layer 3 VLAN interfaces [15](#)

Layer 4 port operations [3](#)

logging

configuration example [18](#)

configuring [17](#)

restrictions [18](#)

MAC address based [12](#)

multicast packets [11](#)

overview [8](#)

SVIs [15](#)

WAN interfaces [8](#)

virtual LAN

See VLANs

vlan

command [10, 12, 9](#)

command example [11](#)

VLAN access control lists, see VACLs

vlan database

command [10, 12, 9](#)

example [11](#)

vlan mapping dot1q

command [13](#)

command example [13](#)

VLANs

allowed on trunk [11](#)

configuration guidelines [8](#)

configuration options

global configuration mode [9](#)

VLAN database mode [9](#)

configuring [1](#)

configuring (tasks) [9](#)

defaults [6](#)

extended range [2](#)

ID (default) [6](#)

interface assignment [12](#)

name (default) [6](#)

normal range [2](#)

private

See private VLANs

protocol filtering and [1](#)

reserved range [2](#)

support for 4,096 VLANs [2](#)

token ring [3](#)

trunks

understanding [2](#)

understanding [1](#)

VLAN 1 minimization [12](#)

VTP domain [3](#)

VLAN Trunking Protocol

See VTP

voice VLAN

Cisco 7960 phone, port connections [1](#)

configuration guidelines [4](#)

configuring IP phone for data traffic

override CoS of incoming frame [7, 8](#)

configuring ports for voice traffic in

802.1Q frames [5](#)

connecting to an IP phone [5](#)

default configuration [4](#)

overview [1](#)

VTP

advertisements [2](#)

client, configuring [8](#)

configuration guidelines [5](#)

default configuration [5](#)

disabling [8](#)

domains [2](#)

VLANs [3](#)

modes

- client [2](#)
- server [2](#)
- transparent [2](#)
- monitoring [10](#)
- overview [1](#)
- pruning
 - configuration [12](#)
 - configuring [7](#)
 - overview [3](#)
- server, configuring [8](#)
- statistics [10](#)
- transparent mode, configuring [8](#)
- version 2
 - enabling [7](#)
 - overview [3](#)

W

WCCP

- configuring on a router [2, 15](#)
- service groups [8](#)
- specifying protocol version [7](#)

web browser interface [1](#)

Web Cache Communication Protocol

- See WCCP [1](#)

web caches

- See cache engines

web cache services

- description [5](#)

web caching

- See web cache services
- See also WCCP

web scaling [1](#)

