ProCurve Networking
HP Innovation

Advanced Traffic Management Guide

6200yl
5400zl
3500yl

ProCurve Switches

K.11.XX

www.procurve.com

ProCurve

Series 5400zl Switches
Series 3500yl Switches
6200yl Switch

July 2006
K.11.xx

Advanced Traffic Management Guide

Publication Number

5991-3827
July 2006

Applicable Products

ProCurve Switch 5406zl	(J8697A)
ProCurve Switch 5412zl	(J8698A)
ProCurve Switch 3500yl-24G-PWR Intelligent Edge	(J8692A)
ProCurve Switch 3500yl-48G-PWR Intelligent Edge	(J8693A)
ProCurve Switch 6200yl-24G	(J8992A)

Trademark Credits

Microsoft, Windows, and Microsoft Windows NT are US registered trademarks of Microsoft Corporation.

Disclaimer

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD COMPANY MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Warranty

See the Customer Support/Warranty booklet included with the product.

A copy of the specific warranty terms applicable to your Hewlett-Packard products and replacement parts can be obtained from your HP Sales and Service Office or authorized dealer.

Hewlett-Packard Company
8000 Foothills Boulevard, m/s 5551
Roseville, California 95747-5551
http://www.procurve.com

Contents

Product Documentation

1 Getting Started

2 Static Virtual LANs (VLANs)

3 GVRP

4 Multiple Instance Spanning-Tree Operation

5 Switch Meshing

8 Stack Management for the Series 3500yl Switches and the 6200yl Switch

Product Documentation

About Your Switch Manual Set

The switch manual set includes the following documentation:

- *Read Me First*—a printed guide shipped with your switch. Provides software update information, product notes, and other information.
- *Installation and Getting Started Guide*—a printed guide shipped with your switch. This guide explains how to prepare for and perform the physical installation and connect the switch to your network.
- *Management and Configuration Guide*—a PDF file on the ProCurve Networking Web Site. This guide describes how to configure, manage, and monitor basic switch operation.
- *Advanced Traffic Management Guide*—a PDF file on the ProCurve Networking Web Site. This guide explains how to configure traffic management features such as VLANs, MSTP, QoS, and Meshing.
- *Multicast and Routing Guide*—a PDF file on the ProCurve Networking Web Site. This guide explains how to configure IGMP, PIM, IP routing, and VRRP features.
- *Access Security Guide*—a PDF file on the ProCurve Networking Web Site. This guide explains how to configure access security features and user authentication on the switch.
- *Release Notes*—posted on the ProCurve Networking web site to provide information on software updates. The release notes describe new features, fixes, and enhancements that become available between revisions of the main product guide.

Note

For the latest version of all ProCurve switch documentation, including Release Notes covering recently added features, visit the ProCurve Networking web sit at **www.procurve.com,** click on **Technical support**, and then click on **Product manuals (all)**.

Feature Index

For the manual set supporting your switch model, the following feature index indicates which manual to consult for information on a given software feature.

Feature	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
802.1Q VLAN Tagging		X		
802.1X Port-Based Priority	X			
802.1X Multiple Authenticated Clients per port				X
ACLs		X		
AAA Authentication				X
Authorized IP Managers				X
Authorized Manager List (web, telnet, TFTP)				X
Auto MDIX Configuration	X			
BOOTP	X			
Config File	X			
Console Access	X			
Copy Command	X			
CoS (Class of Service)		X		
Debug	X			
DHCP Configuration		X		
DHCP Option 82			X	
DHCP/Bootp Operation	X			
Diagnostic Tools	X			
Downloading Software	X			
Eavesdrop Protection				X
Event Log	X			

Feature	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
Factory Default Settings	X			
Flow Control (802.3x)	X			
File Management	X			
File Transfers	X			
Friendly Port Names	X			
Guaranteed Minimum Bandwidth (GMB)	X			
GVRP		X		
Identity-Driven Management (IDM)		X		
IGMP			X	
Interface Access (Telnet, Console/Serial, Web)	X			
IP Addressing	X			
IP Routing			X	
Jumbos Support		X		
LACP	X			
Link	X			
LLDP	X			
LLDP-Med	X			
MAC Address Management	X			
MAC Lockdown				X
MAC Lockout				X
MAC-based Authentication				X
MAC authentication RADIUS support				X
Management VLAN		X		
Meshing		X		
Monitoring and Analysis	X			
Multicast Filtering				X
Multiple Configuration Files	X			

Feature	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
Network Management Applications (SNMP)	X			
OpenView Device Management	X			
OSPF			X	
Passwords and Password Clear Protection				X
PCM	X			
PIM-DM; PIM-SM			X	
Ping	X			
Port Configuration	X			
Port Monitoring		X		
Port Security				X
Port Status	X			
Port Trunking (LACP)	X			
Port-Based Access Control				X
Port-Based Priority (802.1Q)	X			
Power over Ethernet (PoE)	X			
Protocol Filters				X
Protocol VLANS		X		
Quality of Service (QoS)		X		
RADIUS Authentication and Accounting				X
RADIUS-Based Configuration				X
Rate-limiting	X			
RIP			X	
RMON 1,2,3,9	X			
Routing			X	
Routing - IP Static			X	
Secure Copy	X			
SFLOW				

Feature	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
SFTP	X			
SNMPv3	X			
Software Downloads (SCP/SFTP, TFPT, Xmodem)	X			
Source-Port Filters				X
Spanning Tree (STP, RSTP, MSTP)		X		
SSHv2 (Secure Shell) Encryption				X
SSL (Secure Socket Layer)				X
Stack Management (3500yl and 6200yl switches only)		X		
Syslog	X			
System Information	X			
TACACS+ Authentication				X
Telnet Access	X			
TFTP	X			
Time Protocols (TimeP, SNTP)	X			
Traffic/Security Filters				X
Troubleshooting	X			
UDP Forwarder			X	
Virus Throttling (connection-rate filtering)				X
VLANs		X		
VLAN Mirroring (1 static VLAN)		X		
Voice VLAN		X		
VRRP			X	
Web Authentication RADIUS Support				X
Web-based Authentication				X
Web UI	X			
Xmodem	X			

Getting Started

Contents

Introduction

This *Management and Configuration Guide* is intended for use with the following switches:

- ProCurve Switch 5406zl
- ProCurve Switch 5412zl
- ProCurve Switch 3500yl-24G-PWR Intelligent Edge
- ProCurve Switch 3500yl-48G-PWR Intelligent Edge
- ProCurve Switch 6200yl-24G mGBIC Premium Edge

This guide describes how to use the command line interface (CLI), Menu interface, and web browser to configure, manage, monitor, and troubleshoot switch operation.

For an overview of other product documentation for the above switches, refer to *"Product Documentation"* on page xiii.

You can download documentation from the ProCurve Networking Web Site, **www.procurve.com**.

Conventions

This guide uses the following conventions for command syntax and displayed information.

Feature Descriptions by Model

In cases where a software feature is not available in all of the switch models covered by this guide, the section heading specifically indicates which product or product series offer the feature.

For example, (the switch is highlighted here in ***bold italics***):

"QoS Pass-Through Mode on the ***Series 5400zl Switches***".

Command Syntax Statements

Syntax: ip < default-gateway < *ip-addr* >> | routing >

Syntax: show interfaces [*port-list*]

- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate optional elements.
- Braces (< >) enclose required elements.
- Braces within square brackets ([< >]) indicate a required element within an optional choice.
- Boldface indicates use of a CLI command, part of a CLI command syntax, or other displayed element in general text. For example:

  "Use the **copy tftp** command to download the key from a TFTP server."
- Italics indicate variables for which you must supply a value when executing the command. For example, in this command syntax, you must provide one or more port numbers:

  Syntax: aaa port-access authenticator < *port-list* >

Command Prompts

In the default configuration, your switch displays a CLI prompt similar to the following:

```
ProCurve 5406zl#
ProCurve 5412zl#
ProCurve 3500yl#
ProCurve 6200yl#
```

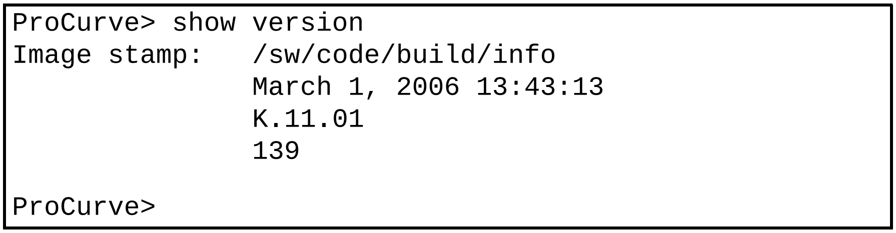
To simplify recognition, this guide uses **ProCurve** to represent command prompts for all models. For example:

```
ProCurve#
```

(You can use the **hostname** command to change the text in the CLI prompt.)

Screen Simulations

Displayed Text. Figures containing simulated screen text and command output look like this:



```
ProCurve> show version
Image stamp:   /sw/code/build/info
               March 1, 2006 13:43:13
               K.11.01
               139

ProCurve>
```

Figure 1-1. Example of a Figure Showing a Simulated Screen

In some cases, brief command-output sequences appear without figure identification. For example:

```
ProCurve(config)# clear public-key
ProCurve(config)# show ip client-public-key
show_client_public_key: cannot stat keyfile
```

Port Identity Examples

This guide describes software applicable to both chassis-based and stackable ProCurve switches. Where port identities are needed in an example, this guide uses the chassis-based port identity system, such as “A1, “B3-B5”, “C7”, etc. However, unless otherwise noted, such examples apply equally to the stackable switches, which typically use only numbers, such as “1”, “3-5”, “15”, etc. for port identities.

Configuration and Operation Examples

Unless otherwise noted, examples using a particular switch model apply to all switch models covered by this guide.

Keys

Simulations of actual keys use a bold, sans-serif typeface with square brackets. For example, the Tab key appears as **[Tab]** and the “Y” key appears as **[Y]**.

Sources for More Information

For additional information about switch operation and features not covered in this guide, consult the following sources:

- Feature Index—For information on which product manual to consult for a given software feature, refer to the "Feature Index" on page xiv.

Note

For the latest version of all ProCurve switch documentation, including Release Notes covering recently added features, visit the ProCurve Networking web site at **www.procurve.com**, click on **Technical support**, and then click on **Product Manuals (all)**.

- Software Release Notes—Release notes are posted on the ProCurve Networking web site and provide information on new software updates:
  - information on the ProCurve Premium Edge License (This option is used on the 3500yl and 5400zl switches to enable certain software features described in the manual set for these switches. The 6200yl switch is available only as a Premium Edge switch.)
  - new features and how to configure and use them
  - software management, including downloading software to the switch
  - software fixes addressed in current and previous releases

  To view and download a copy of the latest software release notes for your switch, refer to "Getting Documentation From the Web" on page 1-7.
- Product Notes and Software Update Information—The printed *Read Me First* shipped with your switch provides software update information, product notes, and other information. For the latest version, refer to "Getting Documentation From the Web" on page 1-7.
- *Installation and Getting Started Guide*—Use the *Installation and Getting Started Guide* shipped with your switch to prepare for and perform the physical installation. This guide also steps you through connecting the switch to your network and assigning IP addressing, as well as describing the LED indications for correct operation and trouble analysis. You can download a copy from the ProCurve Networking web site. (See "Getting Documentation From the Web" on page 1-7.)

- *Management and Configuration Guide*—Use this guide for information on topics such as:
  - various interfaces available on the switch
  - memory and configuration operation
  - interface access
  - IP addressing
  - time protocols
  - port configuration, trunking, traffic control, and PoE operation
  - SNMP, LLDP, and other network management topics
  - file transfers, switch monitoring, troubleshooting, and MAC address management
- *Advanced Traffic Management Guide*—Use this guide for information on topics such as:
  - VLANs: Static port-based and protocol VLANs, and dynamic GVRP VLANs
  - spanning-Tree: 802.1D (STP), 802.1w (RSTP), and 802.1s (MSTP)
  - meshing
  - Quality-of-Service (QoS)
  - Access Control Lists (ACLs)
- *Multicast and Routing Guide*—Use this guide for information topics such as:
  - IGMP
  - PIM (SM and DM)
  - IP routing
  - VRRP
- *Access Security Guide*—Use this guide for information on topics such as:
  - Local username and password security
  - Web-Based and MAC-based authentication
  - RADIUS and TACACS+ authentication
  - SSH (Secure Shell) and SSL (Secure Socket Layer) operation
  - 802.1X access control
  - Port security operation with MAC-based control
  - Authorized IP Manager security
  - Key Management System (KMS)

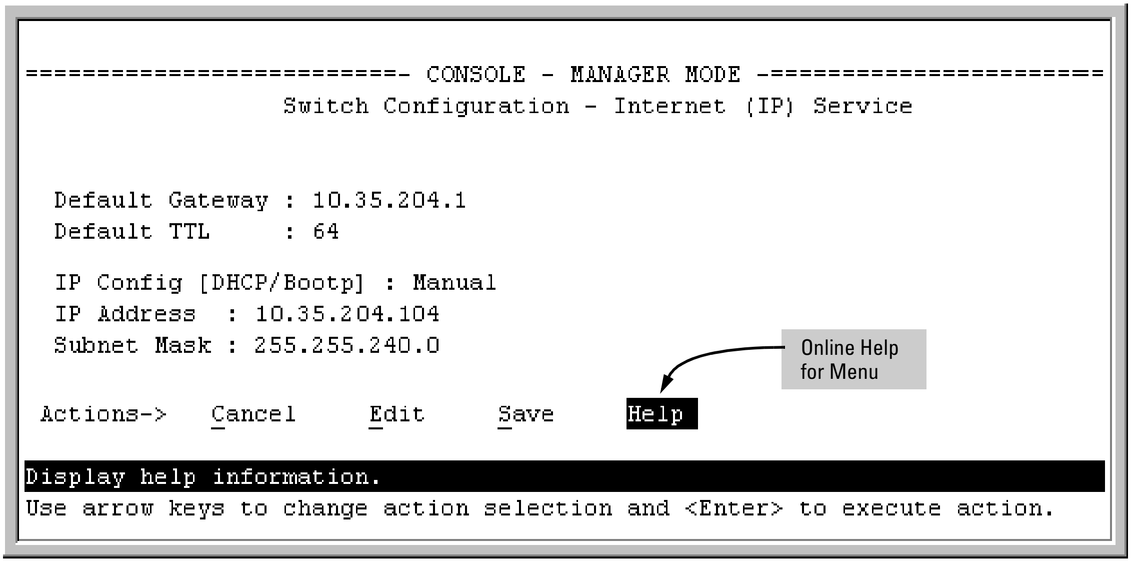
Getting Documentation From the Web

1. Go to the ProCurve Networking web site at

   www.procurve.com
2. Click on **Technical support**.
3. Click on **Product manuals**.
4. Click on the product for which you want to view or download a manual.

Online Help

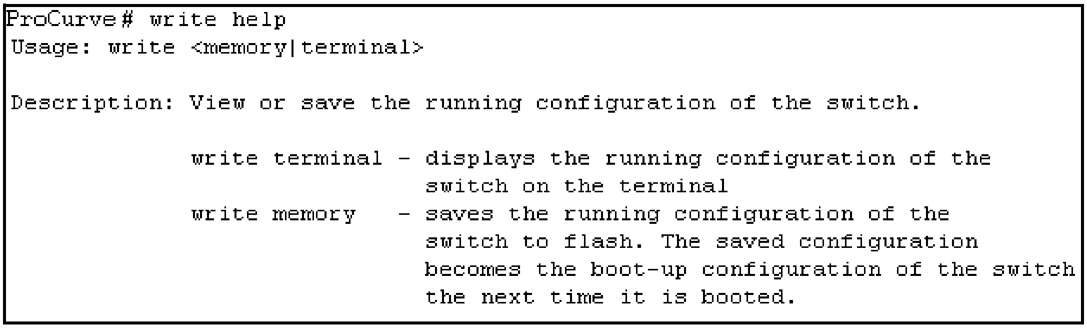
If you need information on specific parameters in the menu interface, refer to the online help provided in the interface. For example:



```
=========================- CONSOLE - MANAGER MODE -=======================
                 Switch Configuration - Internet (IP) Service


  Default Gateway : 10.35.204.1
  Default TTL     : 64

  IP Config [DHCP/Bootp] : Manual
  IP Address  : 10.35.204.104
  Subnet Mask : 255.255.240.0

 Actions->   Cancel     Edit     Save     Help

Display help information.
Use arrow keys to change action selection and <Enter> to execute action.
```

Online Help for Menu

If you need information on a specific command in the CLI, type the command name followed by “help”. For example:



```
ProCurve# write help
Usage: write <memory|terminal>

Description: View or save the running configuration of the switch.

             write terminal - displays the running configuration of the
                              switch on the terminal
             write memory   - saves the running configuration of the
                              switch to flash. The saved configuration
                              becomes the boot-up configuration of the switch
                              the next time it is booted.
```

If you need information on specific features in the ProCurve Web Browser Interface (hereafter referred to as the "web browser interface"), use the online help available for the web browser interface. For more information on web browser Help options, refer to "Online Help for the ProCurve Web Browser Interface" in the Management and Configuration Guide.

If you need further information on ProCurve switch technology, visit the ProCurve Networking web site at:

www.procurve.com

Need Only a Quick Start?

IP Addressing

If you just want to give the switch an IP address so that it can communicate on your network, or if you are not using VLANs, ProCurve recommends that you use the Switch Setup screen to quickly configure IP addressing. To do so, do one of the following:

- Enter **setup** at the CLI Manager level prompt.

  ```
  Procurve# setup
  ```

- In the Main Menu of the Menu interface, select

  8. Run Setup

For more on using the Switch Setup screen, see the *Installation and Getting Started Guide* you received with the switch.

To Set Up and Install the Switch in Your Network

Physical Installation

Use the ProCurve *Installation and Getting Started Guide* (shipped with the switch) for the following:

- Notes, cautions, and warnings related to installing and using the switch and its related modules
- Instructions for physically installing the switch in your network
- Quickly assigning an IP address and subnet mask, set a Manager password, and (optionally) configure other basic features.
- Interpreting LED behavior.

For the latest version of the *Installation and Getting Started Guide* for your switch, refer to “Getting Documentation From the Web” on page 1-7.

Premium Edge Switch Features

The ProCurve 3500yl and 5400zl switches ship with the ProCurve Intelligent Edge software feature set. Additional Premium Edge switch software features for these switches can be acquired by purchasing a Premium Edge license and installing it on the Intelligent Edge version of these switches. Part numbers for the Premium Edge licenses are:

- 3500yl switches: J8993A
- 5400zl switches: J8994A

(Note that the ProCurve 6200yl switch is available only as a Premium Edge switch.)

For the most current information about the features included in the Premium Edge package, refer to the release notes for your product on the ProCurve Networking web site. The Premium Edge License is available from your ProCurve reseller.

—This page is intentionally unused—

Static Virtual LANs (VLANs)

Contents

Overview

This chapter describes how to configure and use static, port-based and protocol-based VLANs on the switches covered in this guide.

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

- Chapter 3, "Using the Menu Interface"
- Chapter 4, "Using the Command Line Interface (CLI)"
- Chapter 5, "Using the Web Browser Interface
- Chapter 6, "Switch Memory and Configuration"

Introduction

VLAN Features

Feature	Default	Menu	CLI	Web
view existing VLANs	n/a	page 2-23 thru 2-28	page 2-29	page 2-39
configuring static VLANs	default VLAN with VID = 1	page 2-23 thru 2-28	page 2-28	page 2-39

VLANs enable you to group users by logical function instead of physical location. This helps to control bandwidth usage within your network by allowing you to group high-bandwidth users on low-traffic segments and to organize users from different LAN segments according to their need for common resources and/or their use of individual protocols. You can also improve traffic control at the edge of your network by separating traffic of different protocol types. VLANs can also enhance your network security by creating separate subnets to help control in-band access to specific network resources.

General VLAN Operation

A VLAN is comprised of multiple ports operating as members of the same subnet (broadcast domain). Ports on multiple devices can belong to the same VLAN, and traffic moving between ports in the same VLAN is bridged (or “switched”). (Traffic moving between different VLANs must be routed.) A s*tatic* VLAN is an 802.1Q-compliant VLAN configured with one or more ports that remain members regardless of traffic usage. (A *dynamic* VLAN is an 802.1Q-compliant VLAN membership that the switch temporarily creates on a port to provide a link to another port in the same VLAN on another device.)

This chapter describes *static* VLANs configured for port-based or protocol-based operation. Static VLANs are configured with a name, VLAN ID number (VID), and port members. (For *dynamic* VLANs, refer to chapter 3, “GVRP” .)

By default, the switches covered in this guide are 802.1Q VLAN-enabled and allow up to 2048 static and dynamic VLANs. (The default static VLAN setting is 8). 802.1Q compatibility enables you to assign each switch port to multiple VLANs, if needed.

Types of Static VLANs Available in the Switch

Port-Based VLANs

This type of static VLAN creates a specific layer-2 broadcast domain comprised of member ports that bridge IPv4 traffic among themselves. Port-Based VLAN traffic is routable on the switches covered in this guide.

Protocol-Based VLANs

This type of static VLAN creates a layer-3 broadcast domain for traffic of a particular protocol, and is comprised of member ports that bridge traffic of the specified protocol type among themselves. Some protocol types are routable on the switches covered in this guide. Refer to table 2-1 on page 2-7.

Designated VLANs

The switch uses these static, port-based VLAN types to separate switch management traffic from other network traffic. While these VLANs are not limited to management traffic only, they can provide improved security and availability for management traffic.

- **The Default VLAN:** This port-based VLAN is always present in the switch and, in the default configuration, includes all ports as members (page 2-45).
- **The Primary VLAN:** The switch uses this port-based VLAN to run certain features and management functions, including DHCP/Bootp responses for switch management. In the default configuration, the Default VLAN is also the Primary VLAN. However, you can designate another, port-based, non-default VLAN, as the Primary VLAN (page 2-45).
- **The Secure Management VLAN:** This optional, port-based VLAN establishes an isolated network for managing the ProCurve switches that support this feature. Access to this VLAN and to the switch's management functions are available only through ports configured as members (page 2-46).
- **Voice VLANs:** This optional, port-based VLAN type enables you to separate, prioritize, and authenticate voice traffic moving through your network, and to avoid the possibility of broadcast storms affecting VoIP (Voice-over-IP) operation (page 2-51).

Note

In a multiple-VLAN environment that includes some older switch models there may be problems related to the same MAC address appearing on different ports and VLANs on the same switch. In such cases the solution is to impose some cabling and VLAN restrictions. For more on this topic, refer to "Multiple VLAN Considerations" on page 2-18.

Terminology

Dynamic VLAN: An 802.1Q VLAN membership temporarily created on a port linked to another device, where both devices are running GVRP. (See also Static VLAN.) For more information, refer to chapter 3, "GVRP" .

Static VLAN: A port-based or protocol-based VLAN configured in switch memory. (**See also Dynamic VLAN**.)

Tagged Packet: A packet that carries an IEEE 802.1Q VLAN ID (VID), which is a two-byte extension that precedes the source MAC address field of an ethernet frame. A VLAN tag is layer 2 data and is transparent to higher layers.

Tagged VLAN: A VLAN that complies with the 802.1Q standard, including priority settings, and allows a port to join multiple VLANs. (See also **Untagged VLAN**.)

Untagged Packet: A packet that does not carry an IEEE 802.1Q VLAN ID (VID).

Untagged VLAN: A VLAN that does not use or forward 802.1Q VLAN tagging, including priority settings. A port can be a member of only one untagged VLAN of a given type (port-based and the various protocol-based types). (**See also Tagged VLAN**.)

VID: The acronym for a VLAN Identification Number. Each 802.1Q-compliant VLAN must have its own unique VID number, and that VLAN must be given the same VID in every device in which it is configured.

Static VLAN Operation

A group of networked ports assigned to a VLAN form a broadcast domain that is separate from other VLANs that may be configured on the switch. On a given switch, packets are bridged between source and destination ports that belong to the same VLAN. Thus, all ports passing traffic for a particular subnet address should be configured to the same VLAN. Cross-domain broadcast traffic in the switch is eliminated and bandwidth is saved by not allowing packets to flood out all ports.

Table 2-1. Comparative Operation of Port-Based and Protocol-Based VLANs

	Port-Based VLANs	Protocol-Based VLANs
IP Addressing	Usually configured with at least one unique IP address. You can create a port-based VLAN without an IP address. However, this limits the switch features available to ports on that VLAN. (Refer to "How IP Addressing Affects Switch Operation" in the chapter on configuring IP addressing in the *Management and Configuration Guide* for the switch.) You can also use multiple IP addresses to create multiple subnets within the same VLAN. (For more on this topic, refer to the chapter on configuring IP addressing in the *Management and Configuration Guide* for the switch.)	You can configure IP addresses on all protocol VLANs. However, IP addressing is used only on IPv4 and IPv6 protocol VLANs.
Untagged VLAN Membership	A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged.	A port can be an untagged member of one protocol VLAN of a specific protocol type (such as IPX or IPv6). If the same protocol type is configured in multiple protocol VLANs, then a port can be an untagged member of only one of those protocol VLANs. For example, if you have two protocol VLANs, 100 and 200, and both include IPX, then a port can be an untagged member of either VLAN 100 or VLAN 200, but not both VLANs. A port's untagged VLAN memberships can include up to four different protocol types. This means that a port can be an untagged member of one of the following: • Four single-protocol VLANs • Two protocol VLANs where one VLAN includes a single protocol and the other includes up to three protocols • One protocol VLAN where the VLAN includes four protocols

	Port-Based VLANs	**Protocol-Based VLANs**
Tagged VLAN Membership	A port can be a tagged member of any port-based VLAN. See above.	A port can be a tagged member of any protocol-based VLAN. See above.
Routing	The switch can internally route IP (IPv4) traffic between port-based VLANs and between port-based and IPv4 protocol-based VLANs if the switch configuration enables IP routing. If the switch is not configured to route traffic internally between port-based VLANs, then an external router must be used to move traffic between VLANs.	If the switch configuration enables IP routing, the switch can internally route IPv4 traffic as follows: • Between multiple IPv4 protocol-based VLANs • Between IPv4 protocol-based VLANs and port-based VLANs. Other protocol-based VLANs require an external router for moving traffic between VLANs. **Note:** NETbeui and SNA are non-routable protocols. End stations intended to receive traffic in these protocols must be attached to the same physical network.
Commands for Configuring Static VLANs	vlan < ***VID*** > [tagged | untagged < [e] ***port-list*** >]	vlan < *VID* > protocol < ipx | ipv4 | ipv6 | arp | appletalk | sna | netbeui > vlan < *VID* > [tagged | untagged < [e] ***port-list*** >]

VLAN Environments

You can configure different VLAN types in any combination. Note that the default VLAN will always be present. (For more on the default VLAN, refer to "VLAN Support and the Default VLAN" on page 2-45.)

Table 2-2. VLAN Environments

VLAN Environment	**Elements**
The default VLAN (port-based; VID of "1") Only	In the default VLAN configuration, all ports belong to VLAN 1 as untagged members. VLAN 1 is a port-based VLAN, for IPv4 traffic.
Multiple VLAN Environment	In addition to the default VLAN, the configuration can include one or more other port-based VLANs and one or more protocol VLANs. (The switches covered in this guide allow up to 2048 (vids up to 4094) VLANs of all types.) Using VLAN tagging, ports can belong to multiple VLANs of all types. Enabling routing on the switch enables the switch to route IPv4 traffic between port-based VLANs and between port-based VLANs and IPv4 protocol VLANs. Routing other types of traffic between VLANs requires an external router capable of processing the appropriate protocol(s).

VLAN Operation

The Default VLAN. In figure 2-1, all ports belong to the default VLAN, and devices connected to these ports are in the same broadcast domain. Except for an IP address and subnet, no configuration steps are needed.

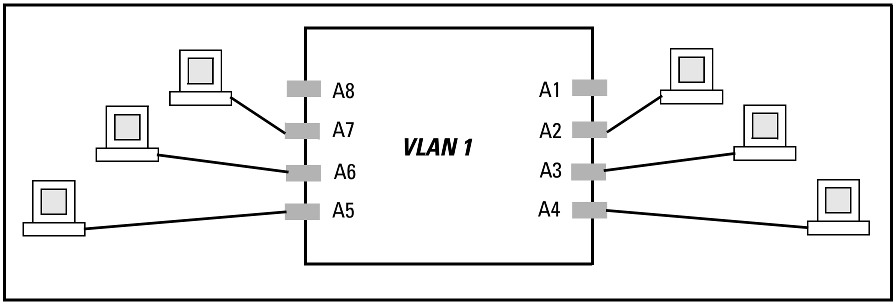


Figure 2-1. Example of a Switch in the Default VLAN Configuration

Multiple Port-Based VLANs. In figure 2-2, routing within the switch is disabled (the default). This means that communication between any routable VLANs on the switch must go through the external router. In this case, VLANs "W" and "X" can exchange traffic through the external router, but traffic in VLANs "Y" and "Z" is restricted to the respective VLANs. Note that VLAN 1, the default VLAN, is also present, but not shown. (The default VLAN cannot be deleted from the switch. However, ports assigned to other VLANs can be removed from the default VLAN, if desired.) If internal (IP) routing is enabled on the switch, then the external router is not needed for traffic to move between port-based VLANs.

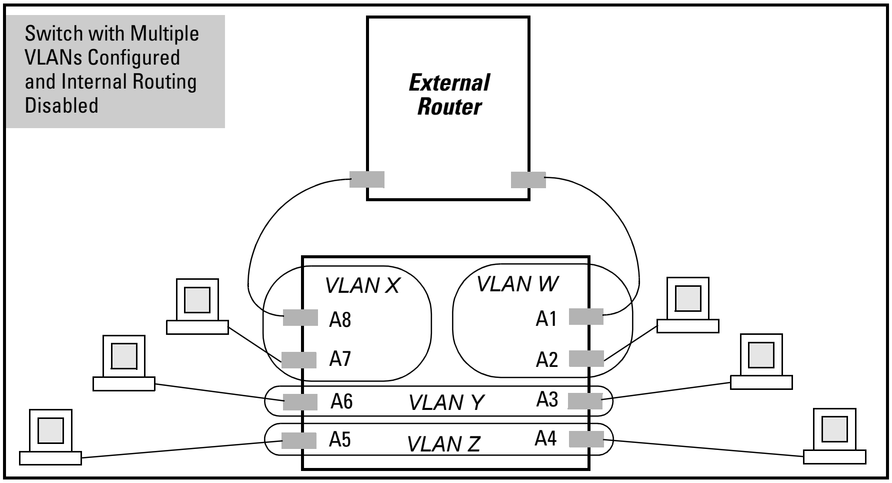

Switch with Multiple VLANs Configured and Internal Routing Disabled
External Router
VLAN X
VLAN W
A8
A7
A1
A2
A6
VLAN Y
A3
A5
VLAN Z
A4

Figure 2-2. Example of Multiple VLANs on the Switch

Protocol VLAN Environment. Figure 2-2 can also be applied to a protocol VLAN environment. In this case, VLANs "W" and "X" represent routable protocol VLANs. VLANs "Y" and "Z" can be any protocol VLAN. As noted for the discussion of multiple port-based VLANs, VLAN 1 is not shown. Enabling internal (IP) routing on the switch allows IP traffic to move between VLANs on the switch. However, routable, non-IP traffic always requires an external router.

Routing Options for VLANs

Table 2-3. Options for Routing Between VLAN Types in the Switch

	Port-Based	IPX	IPv4	IPv6	ARP	Apple-Talk	SNA[2]	Netbeui[2]
Port-Based	Yes	—	Yes	—	—	—	—	—
Protocol								
IPX	—	Yes[1]	—	—	—	—	—	—
IPv4	Yes	—	Yes	—	—	—	—	—
IPv6	—	—	—	Yes[1]	—	—	—	—
ARP	—	—	—	—	Yes[1]	—	—	—
AppleTalk	—	—	—	—	—	Yes[1]	—	—
SNA[2]	—	—	—	—	—	—	—	—
NETbeui[2]	—	—	—	—	—	—	—	—

[1] Requires an external router to route between VLANs.

[2] Not a routable protocol type. End stations intended to receive traffic in these protocols must be attached to the same physical network.

Overlapping (Tagged) VLANs

A port can be a member of more than one VLAN of the same type if the device to which the port connects complies with the 802.1Q VLAN standard. For example, a port connected to a central server using a network interface card (NIC) that complies with the 802.1Q standard can be a member of multiple VLANs, allowing members of multiple VLANs to use the server. Although these VLANs cannot communicate with each other through the server, they can all access the server over the same connection from the switch. Where VLANs overlap in this way, VLAN "tags" are used in the individual packets to distinguish between traffic from different VLANs. A VLAN tag includes the particular VLAN I.D. (VID) of the VLAN on which the packet was generated.

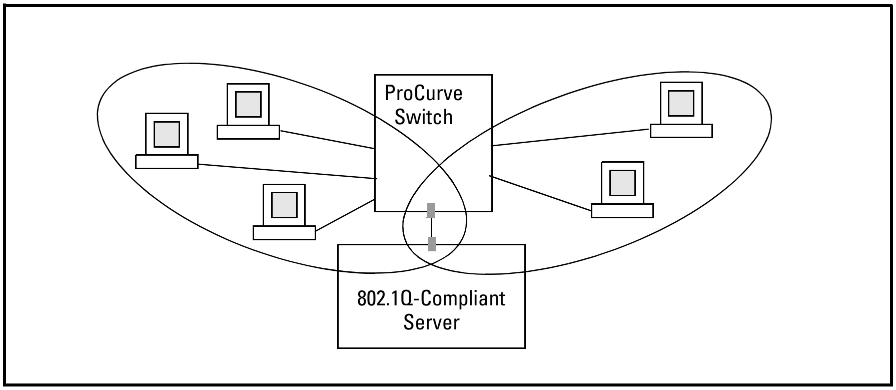


Figure 2-3. Example of Overlapping VLANs Using the Same Server

Similarly, using 802.1Q-compliant switches, you can connect multiple VLANs through a single switch-to-switch link.

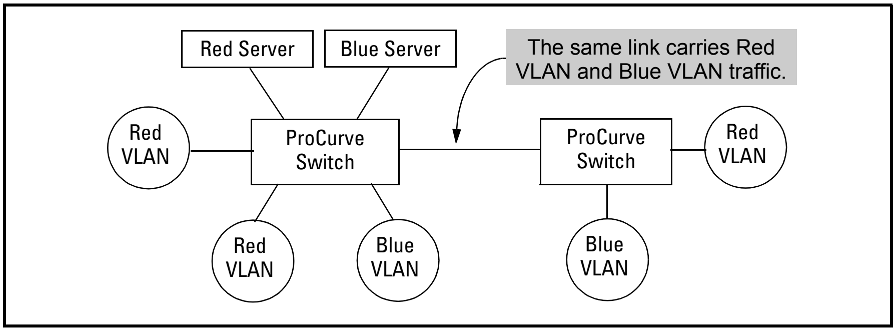


Figure 2-4. Example of Connecting Multiple VLANs Through the Same Link

Introducing Tagged VLAN Technology into Networks Running Legacy (Untagged) VLANs. You can introduce 802.1Q-compliant devices into networks that have built untagged VLANs based on earlier VLAN technology. The fundamental rule is that legacy/untagged VLANs require a separate link for each VLAN, while 802.1Q, or tagged VLANs can combine several VLANs in one link. This means that on the 802.1Q-compliant device, separate ports (configured as untagged) must be used to connect separate VLANs to non-802.1Q devices.

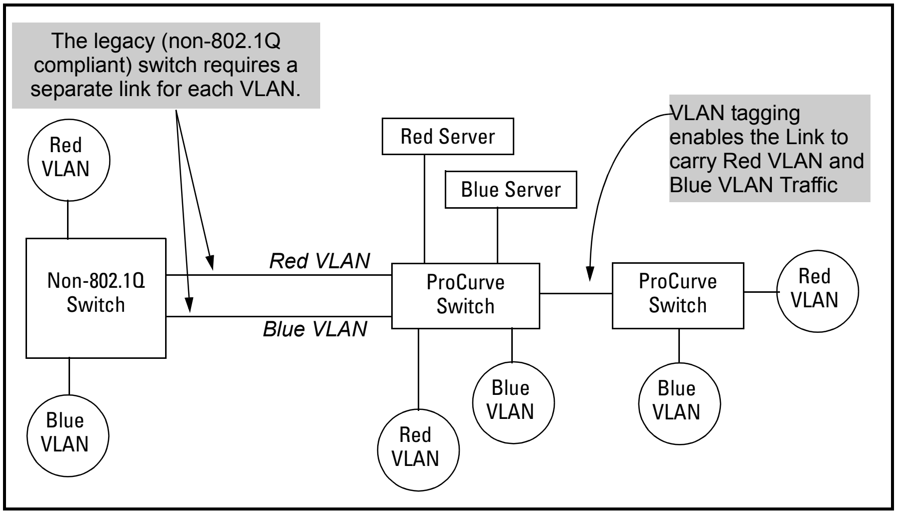

The legacy (non-802.1Q compliant) switch requires a separate link for each VLAN.
Red VLAN
Non-802.1Q Switch
Blue VLAN
Red VLAN
Blue VLAN
Red Server
Blue Server
ProCurve Switch
Red VLAN
Blue VLAN
VLAN tagging enables the Link to carry Red VLAN and Blue VLAN Traffic
ProCurve Switch
Red VLAN
Blue VLAN

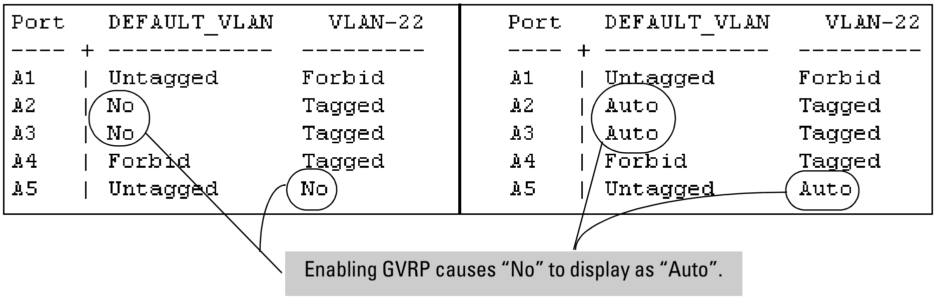
Figure 2-5. Example of Tagged and Untagged VLAN Technology in the Same Network

For more information on VLANs, refer to:

- "Overview of Using VLANs" (page 2-45)
- "Menu: Configuring VLAN Parameters (page 2-22)
- "CLI: Configuring VLAN Parameters" (page 2-22)
- "Web: Viewing and Configuring VLAN Parameters" (page 2-39)
- "VLAN Tagging Information" (page 2-40)
- "Effect of VLANs on Other Switch Features" (page 2-53)
- "VLAN Restrictions" (page 2-55)

Per-Port Static VLAN Configuration Options

The following figure and table show the options you can use to assign individual ports to a static VLAN. Note that GVRP, if configured, affects these options and VLAN behavior on the switch. The display below shows the per-port VLAN configuration options. Table 2-4 briefly describes these options.



Example of Per-Port VLAN Configuration with GVRP Disabled (the default)

```
Port   DEFAULT_VLAN    VLAN-22
---- + ------------  ---------
A1   | Untagged      Forbid
A2   | No            Tagged
A3   | No            Tagged
A4   | Forbid        Tagged
A5   | Untagged      No
```

Example of Per-Port VLAN Configuration with GVRP Enabled

```
Port   DEFAULT_VLAN    VLAN-22
---- + ------------  ---------
A1   | Untagged      Forbid
A2   | Auto          Tagged
A3   | Auto          Tagged
A4   | Forbid        Tagged
A5   | Untagged      Auto
```

Enabling GVRP causes “No” to display as “Auto”.

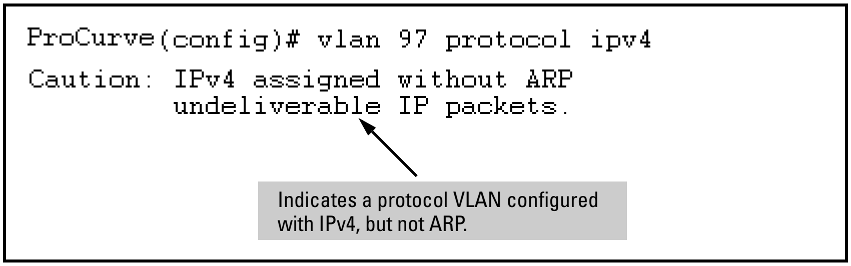
Figure 2-6. Comparing Per-Port VLAN Options With and Without GVRP

Table 2-4. Per-Port VLAN Configuration Options

Parameter	Effect on Port Participation in Designated VLAN
Tagged	Allows the port to join multiple VLANs.
Untagged	Allows VLAN connection to a device that is configured for an untagged VLAN instead of a tagged VLAN. A port can be an untagged member of only one port-based VLAN. A port can also be an untagged member of only one protocol-based VLAN for any given protocol type. For example, if the switch is configured with the default VLAN plus three protocol-based VLANs that include IPX, then port 1 can be an untagged member of the default VLAN and one of the protocol-based VLANS.
No **- or -** **Auto**	**No**: Appears when the switch is not GVRP-enabled; prevents the port from joining that VLAN. **Auto**: Appears when GVRP is enabled on the switch; allows the port to dynamically join any advertised VLAN that has the same VID
Forbid	Prevents the port from joining the VLAN, even if GVRP is enabled on the switch.

VLAN Operating Rules

- **DHCP/Bootp:** If you are using DHCP/Bootp to acquire the switch's configuration, packet time-to-live, and TimeP information, you must designate the VLAN on which DHCP is configured for this purpose as the Primary VLAN. (In the factory-default configuration, the DEFAULT_VLAN is the Primary VLAN.)
- **Per-VLAN Features:** IGMP and some other features operate on a "per VLAN" basis. This means you must configure such features separately for each VLAN in which you want them to operate.
- **Default VLAN:** You can rename the default VLAN, but you cannot change its VID (1) or delete it from the switch.
- **VLAN Port Assignments:** Any ports *not* specifically removed from the default VLAN remain in the DEFAULT_VLAN, regardless of other port assignments. Also, a port must always be a tagged or untagged member of at least one port-based VLAN.
- **Voice-Over-IP (VoIP):** VoIP operates only over static, port-based VLANs.
- **Multiple VLAN Types Configured on the Same Port:** A port can simultaneously belong to both port-based and protocol-based VLANs.
- **Protocol Capacity:** A protocol-based VLAN can include up to four protocol types. In protocol VLANs using the IPv4 protocol, ARP must be one of these protocol types (to support normal IP network operation). Otherwise, IP traffic on the VLAN is disabled. If you configure an IPv4 protocol VLAN that does not already include the ARP VLAN protocol, the switch displays this message:



```
ProCurve(config)# vlan 97 protocol ipv4
Caution: IPv4 assigned without ARP
         undeliverable IP packets.
```

Indicates a protocol VLAN configured with IPv4, but not ARP.

- **Deleting Static VLANs:** On the switches covered in this guide you can delete a VLAN regardless of whether there are currently any ports belonging to that VLAN. (The ports are moved to the default VLAN.)

- **Adding or Deleting VLANs:** Changing the number of VLANs supported on the switch requires a reboot. (From the CLI, you must perform a **write memory** command before rebooting.) Other VLAN configuration changes are dynamic.
- **Inbound Tagged Packets:** If a tagged packet arrives on a port that is not a tagged member of the VLAN indicated by the packet's VID, the switch drops the packet. Similarly, the switch will drop an inbound, tagged packet if the receiving port is an *untagged* member of the VLAN indicated by the packet's VID.
- **Untagged Packet Forwarding:** To enable an inbound port to forward an untagged packet, the port must be an untagged member of either a protocol VLAN matching the packet's protocol or an untagged member of a port-based VLAN. That is, when a port receives an incoming, untagged packet, it processes the packet according to the following ordered criteria:

  a. If the port has no untagged VLAN memberships, the switch drops the packet.

  b. If the port has an untagged VLAN membership in a protocol VLAN that matches the protocol type of the incoming packet, then the switch forwards the packet on that VLAN.

  c. If the port is a member of an untagged, port-based VLAN, the switch forwards the packet to that VLAN. Otherwise, the switch drops the packet.

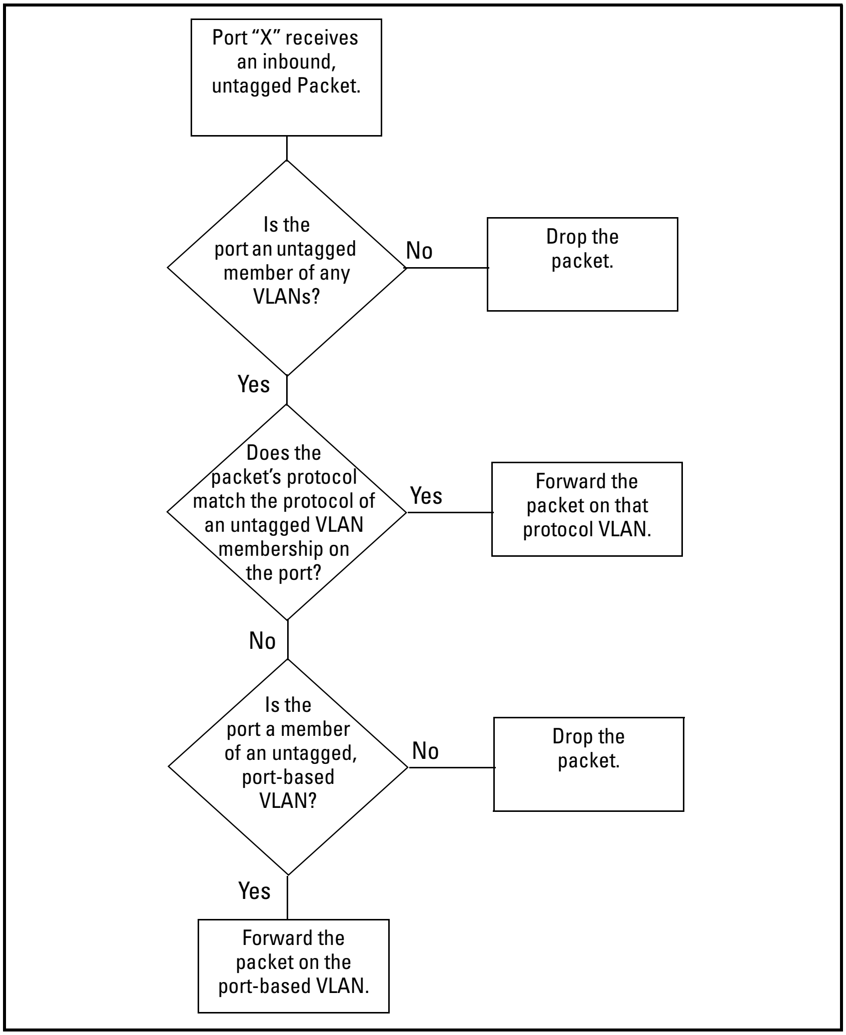

Port "X" receives an inbound, untagged Packet.
Is the port an untagged member of any VLANs?
No
Drop the packet.
Yes
Does the packet's protocol match the protocol of an untagged VLAN membership on the port?
Yes
Forward the packet on that protocol VLAN.
No
Is the port a member of an untagged, port-based VLAN?
No
Drop the packet.
Yes
Forward the packet on the port-based VLAN.

Figure 2-7. Untagged VLAN Operation

- **Tagged Packet Forwarding:** If a port is a tagged member of the same VLAN as an inbound, tagged packet received on that port, then the switch forwards the packet to an outbound port on that VLAN. (To enable the forwarding of tagged packets, any VLAN to which the port belongs as a tagged member must have the same VID as that carried by the inbound, tagged packets generated on that VLAN.)

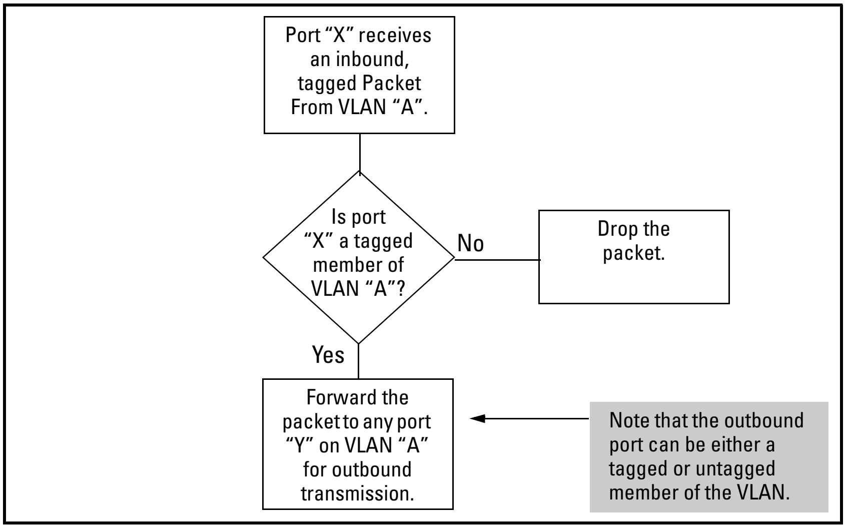

Port "X" receives an inbound, tagged Packet From VLAN "A".
Is port "X" a tagged member of VLAN "A"?
No
Drop the packet.
Yes
Forward the packet to any port "Y" on VLAN "A" for outbound transmission.
Note that the outbound port can be either a tagged or untagged member of the VLAN.

Figure 2-8. Tagged VLAN Operation

See also "Multiple VLAN Considerations" on page 2-18.

General Steps for Using VLANs

1. Plan your VLAN strategy and create a map of the logical topology that will result from configuring VLANs. Include consideration for the interaction between VLANs and other features such as Spanning Tree Protocol, port trunking, and IGMP. (Refer to "Effect of VLANs on Other Switch Features" on page 2-53.) If you plan on using dynamic VLANs, include the port configuration planning necessary to support this feature. (Refer to chapter 3, "GVRP" .)

   By default, VLAN support is enabled and the switch is configured for eight VLANs.

2. Configure at least one VLAN in addition to the default VLAN.
3. Assign the desired switch ports to the new VLAN(s).

4. If you are managing VLANs with SNMP in an IP network, the VLAN through which you are managing the switch must have an IP address. Refer to the chapter titled "Configuring IP Addressing", in the *Management and Configuration Guide* for your switch.

Multiple VLAN Considerations

Switches use a *forwarding database* to maintain awareness of which external devices are located on which VLANs. Some switches, such as the switches covered in this guide, have a *multiple forwarding database*, which means the switch allows multiple database entries of the same MAC address, with each entry showing the (different) source VLAN and source port. Other switch models have a *single forwarding database*, which means they allow only one database entry of a unique MAC address, along with the source VLAN and source port on which it is found. All VLANs on a switch use the same MAC address. Thus, connecting a multiple forwarding database switch to a single forwarding database switch where multiple VLANs exist imposes some cabling and port VLAN assignment restrictions. Table 2-5 illustrates the functional difference between the two database types.

Table 2-5. Example of Forwarding Database Content

Multiple Forwarding Database		
MAC Address	**Destination VLAN ID**	**Destination Port**
0004ea-84d9f4	1	A5
0004ea-84d9f4	22	A12
0004ea-84d9f4	44	A20
0060b0-880a81	33	A20
This database allows multiple destinations for the same MAC address. If the switch detects a new destination for an existing MAC entry, it just **adds** a new instance of that MAC to the table.		

Single Forwarding Database		
MAC Address	**Destination VLAN ID**	**Destination Port**
0004ea-84d9f4	100	A9
0060b0-880af9	105	A10
0060b0-880a81	107	A17
This database allows only one destination for a MAC address. If the switch detects a new destination for an existing MAC entry, it **replaces** the existing MAC instance with a new instance showing the new destination.		

Table 2-6 lists the database structure of current ProCurve switch models.

Table 2-6. Forwarding Database Structure for Managed ProCurve Switches

Multiple Forwarding Databases*	Single Forwarding Database*
Series 6400cl switches	Series 2800 switches
Switch 6200yl	Series 2600 switches
Switch 6108	Switch 1600M/2400M/2424M
Series 5400zl switches	Switch 4000M/8000M
Series 5300xl switches	Series 2500 switches
Series 4200vl switches	Switch 800T
Series 4100gl switches	Switch 2000
Series 3500yl switches	
Series 3400cl switches	

*To determine whether other vendors' devices use single-forwarding or multiple-forwarding database architectures, refer to the documentation provided for those devices.

Single Forwarding Database Operation

When a packet arrives with a destination MAC address that matches a MAC address in the switch's forwarding table, the switch tries to send the packet to the port listed for that MAC address. But, if the destination port is in a different VLAN than the VLAN on which the packet was received, the switch drops the packet. This is not a problem for a switch with a multiple forwarding database (refer to table 2-6, above) because the switch allows multiple instances of a given MAC address; one for each valid destination. However, a switch with a single forwarding database allows only one instance of a given MAC address. If (1) you connect the two types of switches through multiple ports or trunks belonging to different VLANs, and (2) enable routing on the switch having the multiple forwarding database; then, on the switch having the single forwarding database, the port and VLAN record it maintains for the connected multiple-forwarding-database switch can frequently change. This causes poor performance and the appearance of an intermittent or broken connection.

Example of an Unsupported Configuration and How To Correct It

The Problem. In figure 2-9, the MAC address table for Switch 8000M will sometimes record the switch as accessed on port A1 (VLAN 1), and other times as accessed on port B1 (VLAN 2):

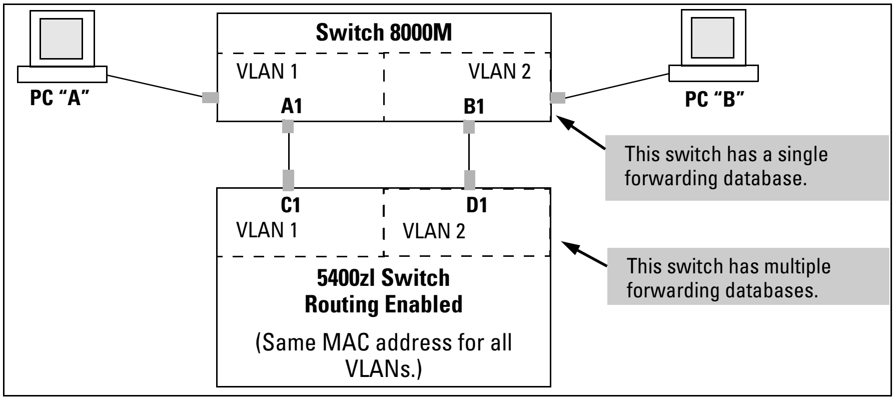

Switch 8000M
VLAN 1
VLAN 2
A1
B1
PC "A"
PC "B"
This switch has a single forwarding database.
C1
D1
VLAN 1
VLAN 2
This switch has multiple forwarding databases.
5400zl Switch
Routing Enabled
(Same MAC address for all VLANs.)

Figure 2-9. Example of Invalid Configuration for Single-Forwarding to Multiple-Forwarding Database Devices in a Multiple VLAN Environment

In figure 2-9, PC "A" sends an IP packet to PC "B".

1. The packet enters VLAN 1 in the Switch 8000 with the Series 5400zl switch's MAC address in the destination field. Because the 8000M has not yet learned this MAC address, it does not find the address in its address table, and floods the packet out all ports, including the VLAN 1 link (port "A1") to the Series 5400zl switch. The Series 5400zl switch then routes the packet through the VLAN 2 link to the 8000M, which forwards the packet on to PC "B". Because the 8000M received the packet from the Series 5400zl switch on VLAN 2 (port "B1"), the 8000M's single forwarding database records the Series 5400zl switch as being on port "B1" (VLAN 2).
2. PC "A" now sends a second packet to PC "B". The packet again enters VLAN 1 in the Switch 8000 with the Series 5400zl switch's MAC address in the destination field. However, this time the Switch 8000M's single forwarding database indicates that the Series 5400zl is on port B1 (VLAN 2), and the 8000M drops the packet instead of forwarding it.
3. Later, the Series 5400zl switch transmits a packet to the 8000M through the VLAN 1 link, and the 8000M updates its address table to indicate that the Series 5400zl switch is on port A1 (VLAN 1) instead of port B1 (VLAN 2). Thus, the 8000M's information on the location of the Series 5400zl

switch changes over time. For this reason, the 8000M discards some packets directed through it for the Series 5400zl switch, resulting in poor performance and the appearance of an intermittent or broken link.

The Solution. To avoid the preceding problem, use only one cable or port trunk between the single-forwarding and multiple-forwarding database devices, and configure the link with multiple, tagged VLANs.

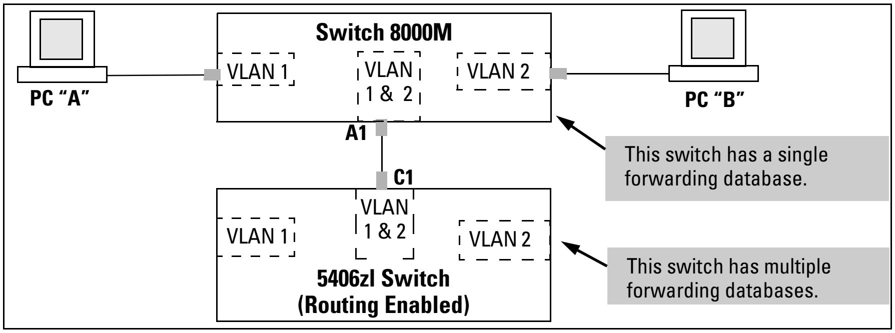

Switch 8000M
VLAN 1
VLAN 1 & 2
VLAN 2
PC "A"
PC "B"
A1
C1
VLAN 1
VLAN 1 & 2
VLAN 2
5406zl Switch (Routing Enabled)
This switch has a single forwarding database.
This switch has multiple forwarding databases.

Figure 2-10. Example of a Solution for Single-Forwarding to Multiple-Forwarding Database Devices in a Multiple VLAN Environment

Now, the 8000M forwarding database always lists the 5400zl MAC address on port A1, and the 8000M will send traffic to either VLAN on the 5400zl.

To increase the network bandwidth of the connection between the devices, you can use a trunk of multiple physical links rather than a single physical link.

Multiple Forwarding Database Operation

If you want to connect one of the switches covered by this guide to another switch that has a multiple forwarding database, you can use either or both of the following connection options:

- A separate port or port trunk interface for each VLAN. This results in a forwarding database having multiple instances of the same MAC address with different VLAN IDs and port numbers. (See table 2-5.) The fact that the switches covered by this guide use the same MAC address on all VLAN interfaces causes no problems.
- The same port or port trunk interface for multiple (tagged) VLANs. This results in a forwarding database having multiple instances of the same MAC address with different VLAN IDs, but the same port number.

Allowing multiple entries of the same MAC address on different VLANs enables topologies such as the following:

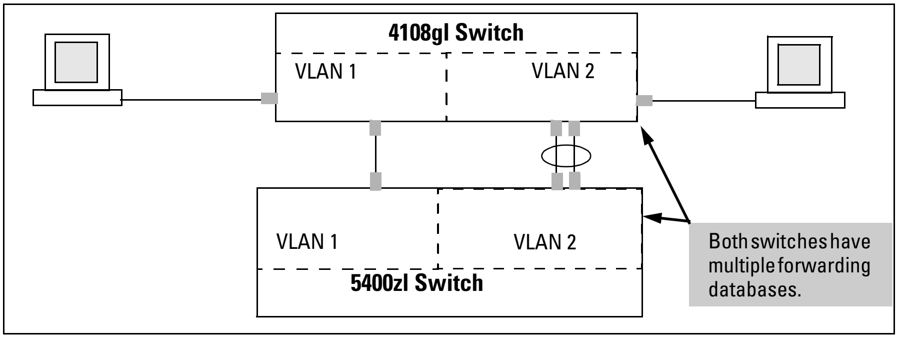


Figure 2-11. Example of a Valid Topology for Devices Having Multiple Forwarding Databases in a Multiple VLAN Environment

Configuring VLANs

Menu: Configuring Port-Based VLAN Parameters

The Menu interface enables you to configure and view port-based VLANs.

Note

The Menu interface configures and displays only port-based VLANs. The CLI configures and displays port-based *and* protocol-based VLANs (page 2-28).

In the factory default state, support is enabled for up to eight VLANs. (You can reconfigure the switch to support up to 2048 (vids up to 4094) VLANs.) Also, in the default configuration, all ports on the switch belong to the default VLAN and are in the same broadcast/multicast domain. (The default VLAN is also the default Primary VLAN—refer to "The Primary VLAN" on page 2-45.) In addition to the default VLAN, you can configure additional static VLANs by adding new VLAN names and VIDs, and then assigning one or more ports to each VLAN. (The maximum of 2048 VLANs includes the default VLAN, all additional static VLANs you configure, and any dynamic VLANs the switch creates if you enable GVRP—page 3-1.) Note that each port can be assigned to multiple VLANs by using VLAN tagging. (See "802.1Q VLAN Tagging" on page 2-40.)

To Change VLAN Support Settings

This section describes:

- Changing the maximum number of VLANs to support
- Changing the Primary VLAN selection (See "Changing the Primary VLAN" on page 2-34.)
- Enabling or disabling dynamic VLANs (Refer to chapter 3, "GVRP" .)

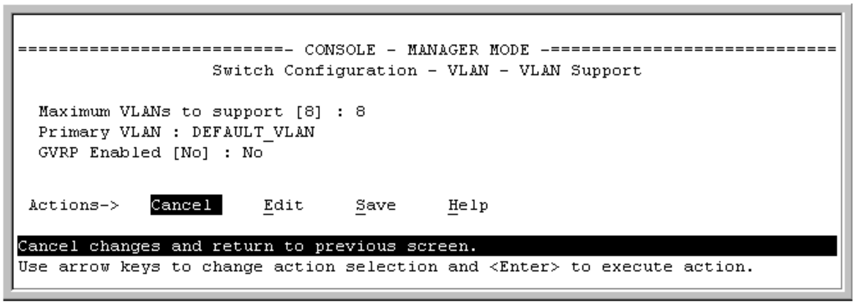
1. From the Main Menu select:

   2. Switch Configuration

   8. VLAN Menu ...

   1. VLAN Support

   You will then see the following screen:



```
==========================- CONSOLE - MANAGER MODE -==========================
                  Switch Configuration - VLAN - VLAN Support

  Maximum VLANs to support [8] : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : No

 Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 2-12. The Default VLAN Support Screen

2. Press **[E]** (for **Edit**), then do one or more of the following:
   - To change the maximum number of VLANs, type the new number (1 - 2048 allowed; default 8).
   - To designate a different VLAN as the Primary VLAN, select the **Primary VLAN** field and use the space bar to select from the existing options. (Note that the Primary VLAN must be a static, port-based VLAN.)
   - To enable or disable dynamic VLANs, select the **GVRP Enabled** field and use the Space bar to toggle between options. (For GVRP information, refer to chapter 3, "GVRP" .)

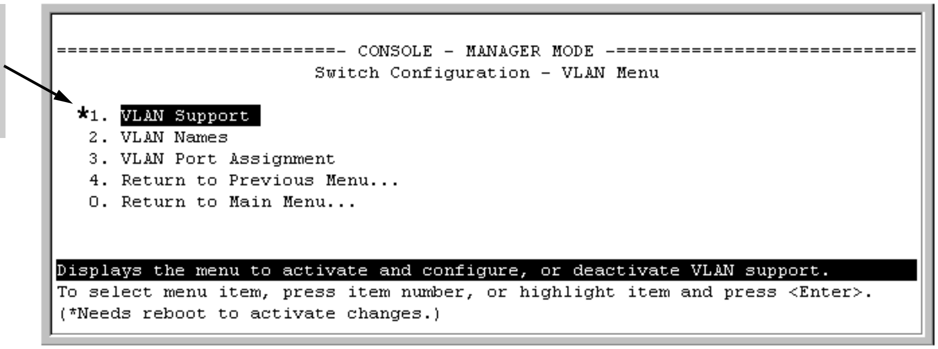
Note

For optimal switch memory utilization, set the number of VLANs at the number you will likely be using or a few more. If you need more VLANs later, you can increase this number, but a switch reboot will be required at that time.

3. Press **[Enter]** and then **[S]** to save the VLAN support configuration and return to the VLAN Menu screen.

If you changed the value for **Maximum VLANs to support**, you will see an asterisk next to the **VLAN Support** option (see below).

An asterisk indicates you must reboot the switch to implement the new Maximum VLANs setting.



```
=========================- CONSOLE - MANAGER MODE -===========================
                        Switch Configuration - VLAN Menu

 *1. VLAN Support
  2. VLAN Names
  3. VLAN Port Assignment
  4. Return to Previous Menu...
  0. Return to Main Menu...


Displays the menu to activate and configure, or deactivate VLAN support.
To select menu item, press item number, or highlight item and press <Enter>.
(*Needs reboot to activate changes.)
```

Figure 2-13. VLAN Menu Screen Indicating the Need To Reboot the Switch

- If you changed the VLAN Support option, you must reboot the switch before the Maximum VLANs change can take effect. You can go on to configure other VLAN parameters first, but remember to reboot the switch when you are finished.
- If you did not change the VLAN Support option, a reboot is not necessary.

4. Press **[0]** to return to the Main Menu.

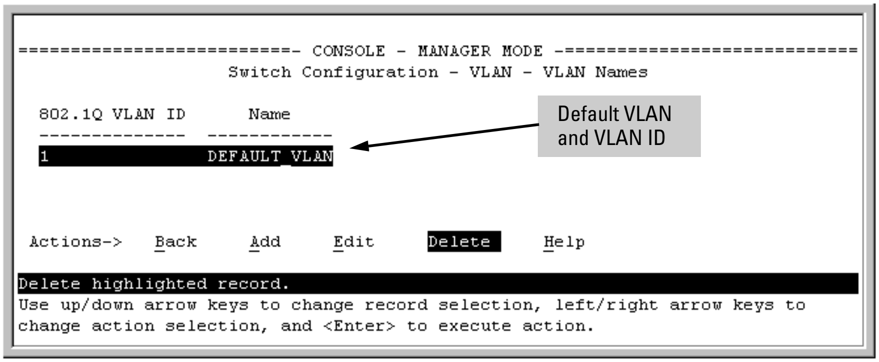
Adding or Editing VLAN Names

Use this procedure to add a new VLAN or to edit the name of an existing VLAN.

1. From the Main Menu select:

   2. Switch Configuration
   8. VLAN Menu ...
   2. VLAN Names

   If multiple VLANs are not yet configured you will see a screen similar to figure 2-14:



```
==========================- CONSOLE - MANAGER MODE -============================
                     Switch Configuration - VLAN - VLAN Names

  802.1Q VLAN ID      Name
  --------------   ------------
  1                DEFAULT_VLAN



 Actions->   Back     Add     Edit     Delete     Help

Delete highlighted record.
Use up/down arrow keys to change record selection, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Default VLAN and VLAN ID

Figure 2-14. The Default VLAN Names Screen

2. Press **[A]** (for **Add**). You will then be prompted for a new VLAN name and VLAN ID:

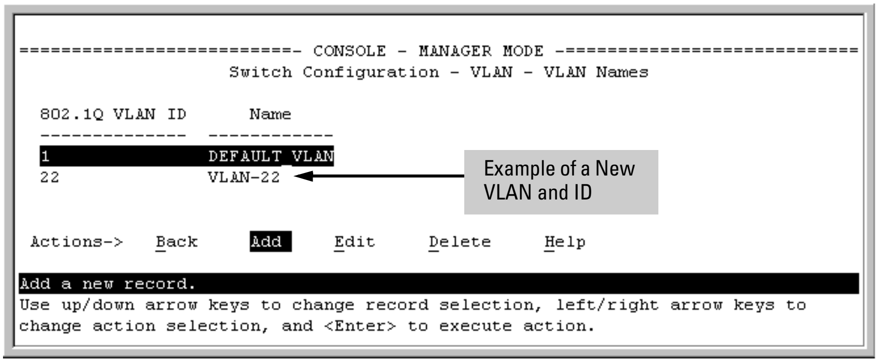
   802.1Q VLAN ID : 1
   Name : _

3. Type in a VID (VLAN ID number). This can be any number from 2 to 4094 that is not already being used by another VLAN. (The switch reserves "1" for the default VLAN.)

   Remember that a VLAN *must* have the same VID in every switch in which you configure that same VLAN. (GVRP dynamically extends VLANs with correct VID numbering to other switches. Refer to chapter 3, "GVRP" .)

4. Press [↓] to move the cursor to the **Name** line and type the VLAN name (up to 12 characters, with no spaces) of a new VLAN that you want to add, then press **[Enter]**.
   (Avoid these characters in VLAN names: **2, #, $, ^, &,** *, **(**, and **)**.)

5. Press **[S]** (for **Save**). You will then see the VLAN Names screen with the new VLAN listed.



```
=========================- CONSOLE - MANAGER MODE -============================
                  Switch Configuration - VLAN - VLAN Names

  802.1Q VLAN ID      Name
  --------------   ------------
  1                DEFAULT_VLAN
  22               VLAN-22

 Actions->   Back      Add      Edit      Delete      Help

Add a new record.
Use up/down arrow keys to change record selection, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Example of a New VLAN and ID

Figure 2-15. Example of VLAN Names Screen with a New VLAN Added

6. Repeat steps 2 through 5 to add more VLANs.

   Remember that you can add VLANs until you reach the number specified in the **Maximum VLANs to support** field on the VLAN Support screen (see figure 2-12 on page 2-23). This includes any VLANs added dynamically due to GVRP operation.

7. Return to the VLAN Menu to assign ports to the new VLAN(s) as described in the next section, "Adding or Changing a VLAN Port Assignment".

Adding or Changing a VLAN Port Assignment

Use this procedure to add ports to a VLAN or to change the VLAN assignment(s) for any port. (Ports not specifically assigned to a VLAN are automatically in the default VLAN.)

1. From the Main Menu select:

   2. Switch Configuration

   8. VLAN Menu ...

   3. VLAN Port Assignment

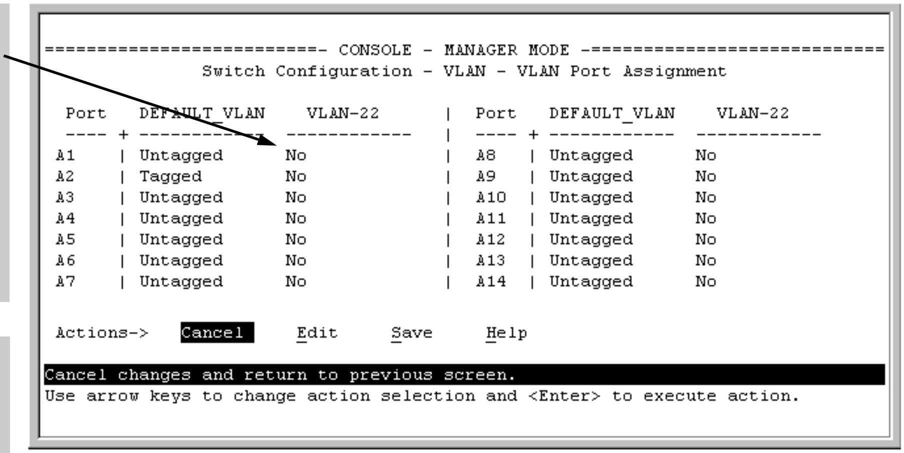
   You will then see a VLAN Port Assignment screen similar to the following:

Note

The "VLAN Port Assignment" screen displays up to 32 static, port-based VLANs in ascending order, by VID. If the switch configuration includes more than 32 such VLANs, use the CLI **show vlans [VID | ports < *port-list* >]** command to list data on VLANs having VIDs numbered sequentially higher than the first 32.

Default: In this example, the “VLAN-22” has been defined, but no ports have yet been assigned to it. (“No” means the port is not assigned to that VLAN.)
Using GVRP? If you plan on using GVRP, any ports you don’t want to join should be changed to “Forbid”.

A port can be assigned to several VLANs, but only one of those assignments can be “Untagged”.



```
=========================- CONSOLE - MANAGER MODE -============================
               Switch Configuration - VLAN - VLAN Port Assignment

  Port   DEFAULT_VLAN    VLAN-22       |  Port   DEFAULT_VLAN    VLAN-22
  ---- + ------------  ------------    |  ---- + ------------  ------------
  A1   | Untagged      No              |  A8   | Untagged      No
  A2   | Tagged        No              |  A9   | Untagged      No
  A3   | Untagged      No              |  A10  | Untagged      No
  A4   | Untagged      No              |  A11  | Untagged      No
  A5   | Untagged      No              |  A12  | Untagged      No
  A6   | Untagged      No              |  A13  | Untagged      No
  A7   | Untagged      No              |  A14  | Untagged      No

 Actions->  Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 2-16. Example of the Port-Based VLAN Port Assignment Screen in the Menu Interface

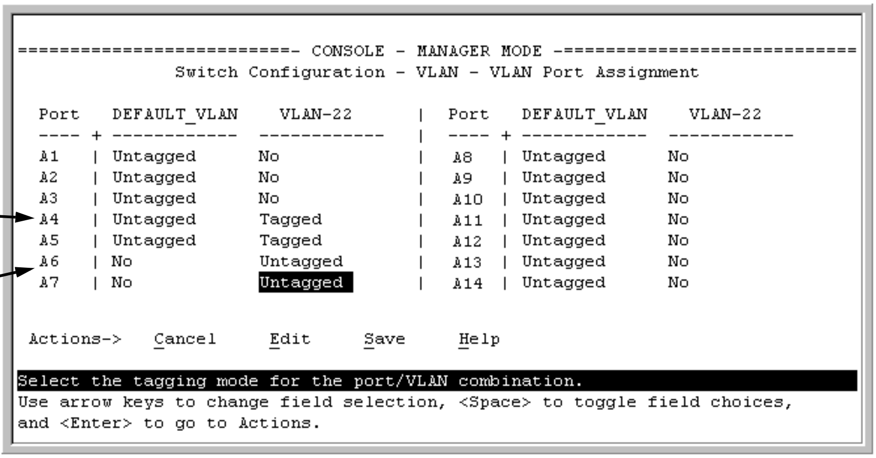
2. To change a port’s VLAN assignment(s):
   a. Press **[E]** (for **Edit**).
   b. Use the arrow keys to select a VLAN assignment you want to change.
   c. Press the Space bar to make your assignment selection (**No**, **Tagged**, **Untagged**, or **Forbid**).

Note

For GVRP Operation: If you enable GVRP on the switch, “**No**” converts to “**Auto**”, which allows the VLAN to dynamically join an advertised VLAN that has the same VID. See “Per-Port Options for Dynamic VLAN Advertising and Joining” on page 3-9.

Untagged VLANs: Only one untagged VLAN is allowed per port. Also, there must be at least one VLAN assigned to each port. In the factory default configuration, all ports are assigned to the default VLAN (DEFAULT_VLAN).

For example, if you want ports A4 and A5 to belong to both DEFAULT_VLAN and VLAN-22, and ports A6 and A7 to belong only to VLAN-22, you would use the settings in figure page 2-28. (This example assumes the default GVRP setting—disabled—and that you do not plan to enable GVRP later.)



```
========================- CONSOLE - MANAGER MODE -===========================
               Switch Configuration - VLAN - VLAN Port Assignment

  Port   DEFAULT_VLAN    VLAN-22      |  Port   DEFAULT_VLAN    VLAN-22
  ---- + ------------  ------------   |  ---- + ------------  ------------
  A1   | Untagged      No             |   A8   | Untagged      No
  A2   | Untagged      No             |   A9   | Untagged      No
  A3   | Untagged      No             |   A10  | Untagged      No
  A4   | Untagged      Tagged         |   A11  | Untagged      No
  A5   | Untagged      Tagged         |   A12  | Untagged      No
  A6   | No            Untagged       |   A13  | Untagged      No
  A7   | No            Untagged       |   A14  | Untagged      No


 Actions->   Cancel     Edit     Save     Help

Select the tagging mode for the port/VLAN combination.
Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

Ports A4 and A5 are assigned to both VLANs.

Ports A6 and A7 are assigned only to VLAN-22.

All other ports are assigned only to the Default VLAN.

Figure 2-17. Example of Port-Based VLAN Assignments for Specific Ports

For information on VLAN tags ("Untagged" and "Tagged"), refer to "802.1Q VLAN Tagging" on page 2-40.

d. If you are finished assigning ports to VLANs, press **[Enter]** and then **[S]** (for **Save**) to activate the changes you've made and to return to the Configuration menu. (The console then returns to the VLAN menu.)

3. Return to the Main menu.

CLI: Configuring Port-Based and Protocol-Based VLAN Parameters

In the factory default state, all ports on the switch belong to the (port-based) default VLAN (DEFAULT_VLAN; VID = 1) and are in the same broadcast/multicast domain. (The default VLAN is also the Primary VLAN. For more on this topic, refer to "The Primary VLAN" on page 2-45.) You can configure up to 255 additional static VLANs by adding new VLAN names, and then assigning one or more ports to each VLAN. (The switch accepts a maximum of 2048 (vids numbered up to 4094) VLANs, including the default VLAN and any dynamic VLANs the switch creates if you enable GVRP. Refer to chapter 3, "GVRP" .) Note that each port can be assigned to multiple VLANs by using VLAN tagging. (See "802.1Q VLAN Tagging" on page 2-40.)

VLAN Commands	**Page**
show vlans	below
show vlans < *vid* >	2-32
show vlans ports <*port-list*>	
max-vlans <1-2048>	2-33
primary-vlan < *vid* >	2-34
[no] vlan < *vid* >	2-35
auto < *port-list* >	2-37 (Available if GVRP enabled.)
forbid	2-37
name < *vlan-name* >	2-37
protocol < *protocol-list* >	2-35
tagged < *port-list* >	2-37
untagged < *port-list* >	2-37
voice	2-51
static-vlan < *vlan-id* >	2-37 (Available if GVRP enabled.)

Displaying the Switch's VLAN Configuration. The **show vlans** command lists the VLANs currently running in the switch, with VID, VLAN name, and VLAN status. Dynamic VLANs appear only if the switch is running with GVRP enabled and one or more ports has dynamically joined an advertised VLAN. (In the default configuration, GVRP is disabled. (Refer to chapter 3, "GVRP" .)

Syntax: show vlans

Maximum VLANs to support: *Shows the number of VLANs the switch can currently support. (Default: 8; Maximum: 2048)*

Primary VLAN: *Refer to "The Primary VLAN" on page 2-45.*

Management VLAN: *Refer to "The Secure Management VLAN" on page 2-46.*

802.1Q VLAN ID: *The VLAN identification number, or VID. Refer to "Terminology" on page 2-6.*

Name: *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-*x*** *where "**x**" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_*x*** *where "**x**" matches the applicable VID.*

Status:

Port-Based: *Port-Based, static VLAN*

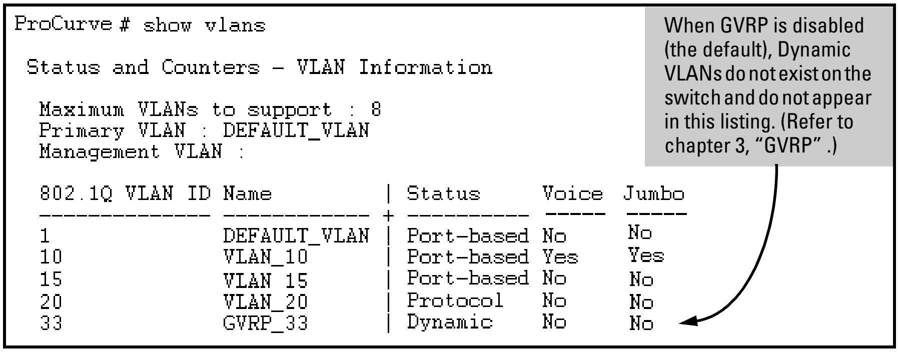
Protocol: *Protocol-Based, static VLAN*

Dynamic: *Port-Based, temporary VLAN learned through GVRP (Refer to chapter 3, "GVRP" .)*

Voice: *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-51.*

Jumbo: *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*

For example:



```
ProCurve # show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  802.1Q VLAN ID Name         | Status     Voice  Jumbo
  -------------- ------------ + ---------- -----  -----
  1              DEFAULT_VLAN | Port-based No     No
  10             VLAN_10      | Port-based Yes    Yes
  15             VLAN 15      | Port-based No     No
  20             VLAN_20      | Protocol   No     No
  33             GVRP_33      | Dynamic    No     No
```

When GVRP is disabled (the default), Dynamic VLANs do not exist on the switch and do not appear in this listing. (Refer to chapter 3, "GVRP" .)

Figure 2-18. Example of "Show VLAN" Listing (GVRP Enabled)

Displaying the VLAN Membership of One or More Ports.

This command shows to which VLAN a port belongs.

Syntax show vlan ports < *port-list* >

802.1Q VLAN ID: *The VLAN identification number, or VID. Refer to "Terminology" on page 2-6.*

Name: *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-*x*** *where "**x**" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_*x*** *where "**x**" matches the applicable VID.*

Status:

Port-Based: *Port-Based, static VLAN*

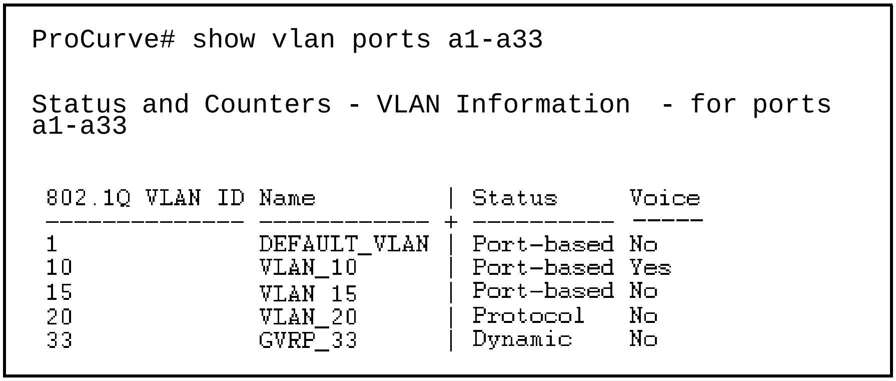
Protocol: *Protocol-Based, static VLAN*

Dynamic: *Port-Based, temporary VLAN learned through GVRP (Refer to chapter 3, "GVRP" .)*

Voice: *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-51.*

Jumbo: *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*

For example:



```
ProCurve# show vlan ports a1-a33

Status and Counters - VLAN Information  - for ports
a1-a33


 802.1Q VLAN ID Name         | Status     Voice
 -------------- ------------ + ---------- -----
 1              DEFAULT_VLAN | Port-based No
 10             VLAN_10      | Port-based Yes
 15             VLAN 15      | Port-based No
 20             VLAN_20      | Protocol   No
 33             GVRP_33      | Dynamic    No
```

Figure 2-19. Example of "Show VLAN Ports" listing

Displaying the Configuration for a Particular VLAN . This command uses the VID to identify and display the data for a specific static or dynamic VLAN.

Syntax: show vlans < *vlan-id* >

802.1Q VLAN ID: *The VLAN identification number, or VID. Refer to "Terminology" on page 2-6.*

Name: *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-*x*** *where "**x**" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_*x*** *where "**x**" matches the applicable VID.*

Status:

Port-Based: *Port-Based, static VLAN*

Protocol: *Protocol-Based, static VLAN*

Dynamic: *Port-Based, temporary VLAN learned through GVRP (Refer to chapter 3,"GVRP" in this guide.)*

Voice: *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-51.*

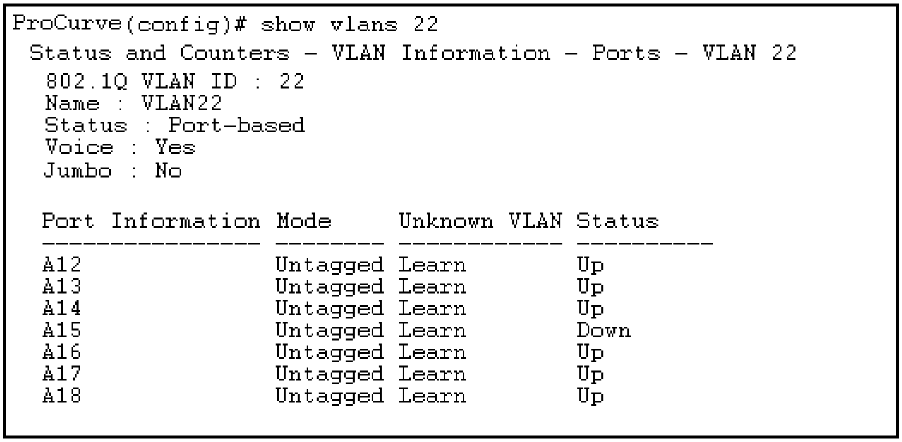
Jumbo: *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*

Port Information: *Lists the ports configured as members of the VLAN.*

DEFAULT: *Shows whether a port is a tagged or untagged member of the listed VLAN.*

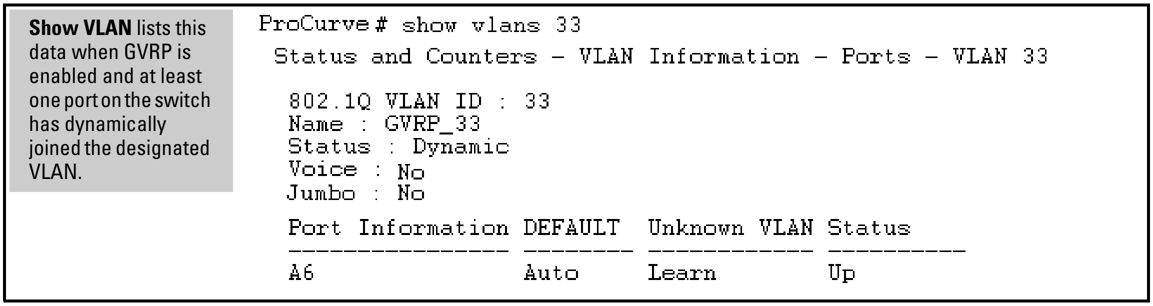
Unknown VLAN: *Shows whether the port can become a dynamic member of an unknown VLAN for which it receives an advertisement. GVRP must be enabled to allow dynamic joining to occur. Refer to table 3-1 on page 3-8.*

Status: *Shows whether the port is participating in an active link.*



```
ProCurve(config)# show vlans 22
 Status and Counters - VLAN Information - Ports - VLAN 22
  802.1Q VLAN ID : 22
  Name : VLAN22
  Status : Port-based
  Voice : Yes
  Jumbo : No

  Port Information Mode     Unknown VLAN Status
  ---------------- -------- ------------ ----------
  A12              Untagged Learn        Up
  A13              Untagged Learn        Up
  A14              Untagged Learn        Up
  A15              Untagged Learn        Down
  A16              Untagged Learn        Up
  A17              Untagged Learn        Up
  A18              Untagged Learn        Up
```

Figure 2-20. Example of "Show VLAN" for a Specific Static VLAN



Show VLAN lists this data when GVRP is enabled and at least one port on the switch has dynamically joined the designated VLAN.

```
ProCurve# show vlans 33
 Status and Counters - VLAN Information - Ports - VLAN 33

  802.1Q VLAN ID : 33
  Name : GVRP_33
  Status : Dynamic
  Voice : No
  Jumbo : No

  Port Information DEFAULT  Unknown VLAN Status
  ---------------- -------- ------------ ----------
  A6               Auto     Learn        Up
```

Figure 2-21. Example of "Show VLAN" for a Specific Dynamic VLAN

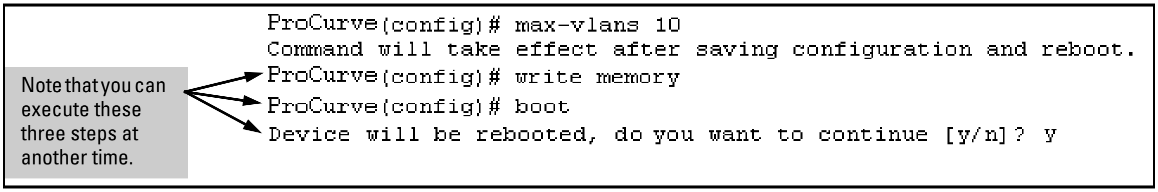
Changing the Number of VLANs Allowed on the Switch. In the default VLAN configuration, the switch allows a maximum of 8 VLANs. You can specify any value from 1 to 256.

Syntax: max-vlans < 1-256 >

Specifies the maximum number of VLANs to allow. (If GVRP is enabled, this setting includes any dynamic VLANs on the switch.) As part of implementing a new setting, you must execute a **write memory** *command (to save the new value to the startup-config file) and then reboot the switch.*

Note: *If multiple VLANs exist on the switch, you cannot reset the maximum number of VLANs to a value smaller than the current number of VLANs.*

For example, to reconfigure the switch to allow 10 VLANs:



```
ProCurve(config)# max-vlans 10
Command will take effect after saving configuration and reboot.
ProCurve(config)# write memory
ProCurve(config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Note that you can execute these three steps at another time.

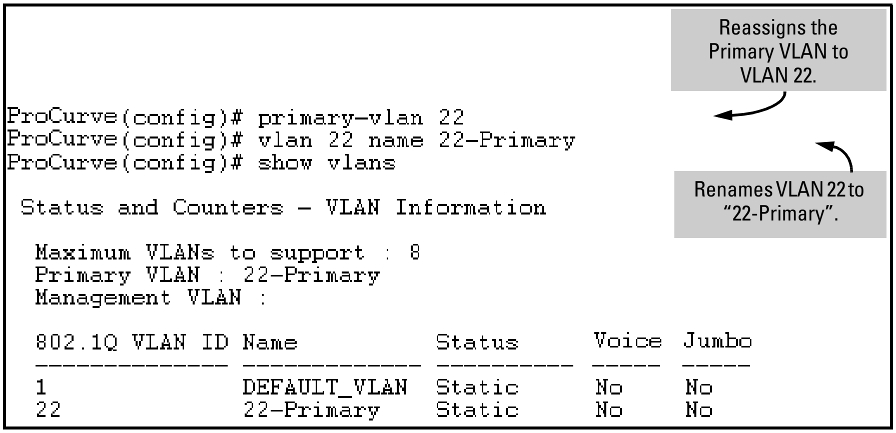
Figure 2-22. Example of Command Sequence for Changing the Number of VLANs

Changing the Primary VLAN. In the default VLAN configuration, the port-based default VLAN (**DEFAULT_VLAN**) is the Primary VLAN. However, you can reassign the Primary VLAN to any port-based, static VLAN on the switch. (For more on the Primary VLAN, refer to "The Primary VLAN" on page 2-45.) To identify the current Primary VLAN and list the available VLANs and their respective VIDs, use **show vlans**.

Syntax: primary-vlan < *vid | ascii-name-string* >

Reassigns the Primary VLAN function. Re-assignment must be to an existing, port-based, static VLAN. (The switch will not reassign the Primary VLAN function to a protocol VLAN.) If you re-assign the Primary VLAN to a non-default VLAN, you cannot later delete that VLAN from the switch until you again re-assign the Primary VLAN to another port-based, static VLAN.

For example, if you wanted to reassign the Primary VLAN to VLAN 22 and rename the VLAN with "22-Primary" and display the result:



Reassigns the Primary VLAN to VLAN 22.

Renames VLAN 22 to "22-Primary".

```
ProCurve(config)# primary-vlan 22
ProCurve(config)# vlan 22 name 22-Primary
ProCurve(config)# show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 8
  Primary VLAN : 22-Primary
  Management VLAN :

  802.1Q VLAN ID Name         Status     Voice Jumbo
  -------------- ------------ ---------- ----- -----
  1              DEFAULT_VLAN Static     No    No
  22             22-Primary   Static     No    No
```

Figure 2-23. Example of Reassigning Primary VLAN and Changing the VLAN Name

Creating a New Static VLAN (Port-Based or Protocol-Based) Changing the VLAN Context Level. The **vlan <** ***vid*** **>** command operates in the global configuration context to either configure a static VLAN and/or take the CLI to the specified VLAN's context.

Syntax: vlan < *vid | ascii-name-string* >

[no] vlan < *vid* >

If < vid > *does not exist in the switch, this command creates a port-based VLAN with the specified* < vid >. *If the command does not include options, the CLI moves to the newly created VLAN context. If you do not specify an optional name, the switch assigns a name in the default format:* **VLAN*n*** *where* ***n*** *is the* < vid > *assigned to the VLAN. If the VLAN already exists and you enter either the* ***vid*** *or the* ***ascii-name-string****, the CLI moves to the specified VLAN's context.*

The **[no]** *form of the command deletes the VLAN as follows:*

- *If one or more ports belong only to the VLAN to be deleted, the CLI notifies you that these ports will be moved to the default VLAN and prompts you to continue the deletion. For member ports that also belong to another VLAN, there is no "move" prompt.*

[protocol < ipx | ipv4 | ipv6 | arp | appletalk | sna | netbeui >]

Configures a static, protocol VLAN of the specified type. If multiple protocols are configured in the VLAN, then the **[no]** *form removes the specified protocol from the VLAN. If a protocol VLAN is configured with only one protocol type and you use the* **[no]** *form of this command to remove that protocol, the switch changes the protocol VLAN to a port-based VLAN if the VLAN does not have an untagged member port. (If an untagged member port exists on the protocol VLAN, you must either convert the port to a tagged member or remove the port from the VLAN before removing the last protocol type from the VLAN.)*

Note: *If you create an IPv4 protocol VLAN, you must also assign the* ARP *protocol option to the VLAN to provide IP address resolution. Otherwise, IP packets are not deliverable. A "Caution" message appears in the CLI if you configure IPv4 in protocol VLAN that does not already include the arp protocol option. The same message appears if you add or delete another protocol in the same VLAN.*

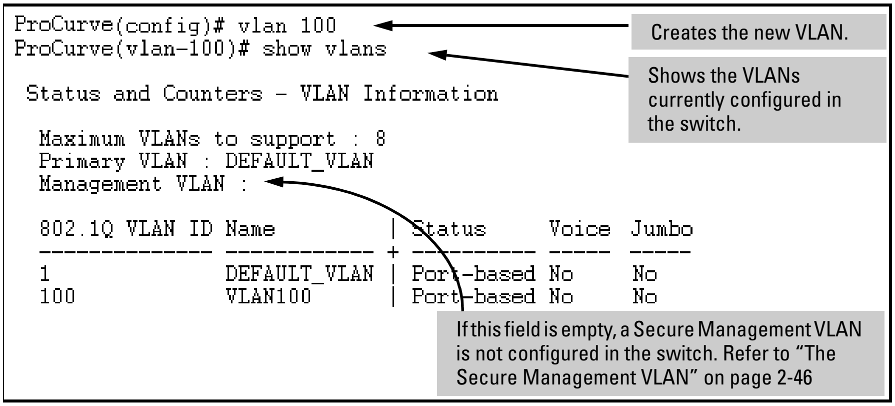
name < *ascii-name-string* >

When included in a **vlan** *command for creating a new static VLAN, specifies a non-default VLAN name. Also used to change the current name of an existing VLAN. (Avoid spaces and the following characters in the <***ascii-name-string***> entry:* **@**, **#**, **$**, **^**, **&**, *****, **(**, and **)**. *To include a blank space in a VLAN name, enclose the name in single or double quotes ('...' or "...").*

[voice]

Designates a VLAN for VoIP use. For more on this topic, refer to "Voice VLANs" on page 2-51.

For example, to create a new, port-based, static VLAN with a VID of 100:



```
ProCurve(config)# vlan 100
ProCurve(vlan-100)# show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  802.1Q VLAN ID Name         | Status     Voice  Jumbo
  -------------- ------------ + ---------- -----  -----
  1              DEFAULT_VLAN | Port-based No     No
  100            VLAN100      | Port-based No     No
```

Creates the new VLAN.

Shows the VLANs currently configured in the switch.

If this field is empty, a Secure Management VLAN is not configured in the switch. Refer to "The Secure Management VLAN" on page 2-46

Figure 2-24. Example of Creating a New, Port-Based, Static VLAN

To go to a different VLAN context level, such as to the default VLAN:

```
ProCurve(vlan-100)# vlan default_vlan
ProCurve(vlan-1) _
```

Deleting a VLAN . If ports B1-B5 belong to both VLAN 2 and VLAN 3, and ports B6-B10 belong to VLAN 3 only, then deleting VLAN 3 causes the CLI to prompt you to approve moving ports B6 - B10 to VLAN 1 (the default VLAN). (Ports B1-B5 are not moved because they still belong to another VLAN.)

```
ProCurve(config)# no vlan 3
The following ports will be moved to the default VLAN:
B6-B10
Do you want to continue? [y/n] y
ProCurve(config)#
```

Converting a Dynamic VLAN to a Static VLAN. Use this feature if you want to convert a dynamic, port-based VLAN membership to a static, port-based VLAN membership. This is necessary if you want to make the VLAN permanent on the switch.

Syntax: static-vlan < *vlan-id* >

Converts a dynamic, port-based VLAN membership to a static, port-based VLAN membership. (Allows port-based VLANs only). For this command, < *vlan-id* > *refers to the VID of the dynamic VLAN membership. (Use* **show vlan** to help identify the VID you need to use.) *This command requires that GVRP is running on the switch and a port is currently a dynamic member of the selected VLAN. After you convert a dynamic VLAN to static, you must configure the switch's per-port participation in the VLAN in the same way that you would for any static VLAN. (For GVRP and dynamic VLAN operation, refer to chapter 3, "GVRP" .)*

For example, suppose a dynamic VLAN with a VID of 125 exists on the switch. The following command converts the VLAN to a port-based, static VLAN.

```
ProCurve(config)# static-vlan 125
```

Configuring Static VLAN Per-Port Settings. The **vlan <***vlan-id***>** command, used with the options listed below, changes the name of an existing static VLAN and changes the per-port VLAN membership settings.

Note

You can use these options from the configuration level by beginning the command with **vlan <** *vid* **>**, or from the context level of the specific VLAN by just typing the command option.

Syntax: [no] vlan < *vid* >

tagged < *port-list* >

Configures the indicated port(s) as **Tagged** *for the specified VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**.

untagged < *port-list* >

Configures the indicated port(s) as **Untagged** *for the specified VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**.

forbid < *port-list* >

Used in port-based VLANs to configures **<** ***port-list*** **>** *as "forbidden" to become a member of the specified VLAN, as well as other actions. Does not operate with protocol VLANs. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto***. Refer to chapter 3, "GVRP", in this guide.*

auto < *port-list* >

Available if GVRP is enabled on the switch. Returns the per-port settings for the specified VLAN to **Auto** *operation. Note that* **Auto** *is the default per-port setting for a static VLAN if GVRP is running on the switch.* (For information on dynamic VLAN and GVRP operation, refer to *chapter 3, "GVRP", in this guide.*)

For example, suppose you have a VLAN named VLAN100 with a VID of 100, and all ports are set to **No** for this VLAN. To change the VLAN name to "**Blue_Team**" and set ports A1 - A5 to **Tagged**, you would use these commands:

```
ProCurve(config)# vlan 100 name Blue_Team
ProCurve(config)# vlan 100 tagged a1-a5
```

To move to the vlan 100 context level and execute the same commands:

```
ProCurve(config)# vlan 100
ProCurve(vlan-100)# name Blue_Team
ProCurve(vlan-100)# tagged a1-a5
```

Similarly, to change the tagged ports in the above examples to **No** (or **Auto**, if GVRP is enabled), you could use either of the following commands.

At the global config level, use:

```
ProCurve(config)# no vlan 100 tagged a1-a5
```

- or -

At the VLAN 100 context level, use:

```
ProCurve(vlan-100)# no tagged a1-a5
```

Note

You cannot use these commands with dynamic VLANs. Attempting to do so results in the message "**VLAN already exists.**" and no change occurs.

Web: Viewing and Configuring VLAN Parameters

In the web browser interface you can do the following:

- Add VLANs
- Rename VLANs
- Remove VLANs
- Configure VLAN tagging mode per-port
- Configure GVRP mode
- Select a new Primary VLAN

To configure other static VLAN port parameters, you will need to use either the CLI or the menu interface (available by Telnet from the web browser interface).

1. Click on the Configuration tab.
2. Click on **[Vlan Configuration]**.
3. Click on **[Add/Remove VLANs]**.

For web-based Help on how to use the web browser interface screen, click on the **[?]** button provided on the web browser screen.

802.1Q VLAN Tagging

General Applications:

- The switch requires VLAN tagging on a given port if more than one VLAN of the same type uses the port. When a port belongs to two or more VLANs of the same type, they remain as separate broadcast domains and cannot receive traffic from each other without routing. (If multiple, *non-routable* VLANs exist in the switch—such as NETbeui protocol VLANs— then they cannot receive traffic from each other under any circumstances.)
- The switch requires VLAN tagging on a given port if the port will be receiving inbound, tagged VLAN traffic that should be forwarded. Even if the port belongs to only one VLAN, it forwards inbound tagged traffic only if it is a tagged member of that VLAN.
- If the only authorized, inbound VLAN traffic on a port arrives untagged, then the port must be an untagged member of that VLAN. This is the case where the port is connected to a non 802.1Q-compliant device or is assigned to only one VLAN.

For example, if port 7 on an 802.1Q-compliant switch is assigned to only the Red VLAN, the assignment can remain “untagged” because the port will forward traffic only for the Red VLAN. However, if both the Red and Green VLANs are assigned to port 7, then at least one of those VLAN assignments must be “tagged” so that Red VLAN traffic can be distinguished from Green VLAN traffic. Figure 2-25 shows this concept:

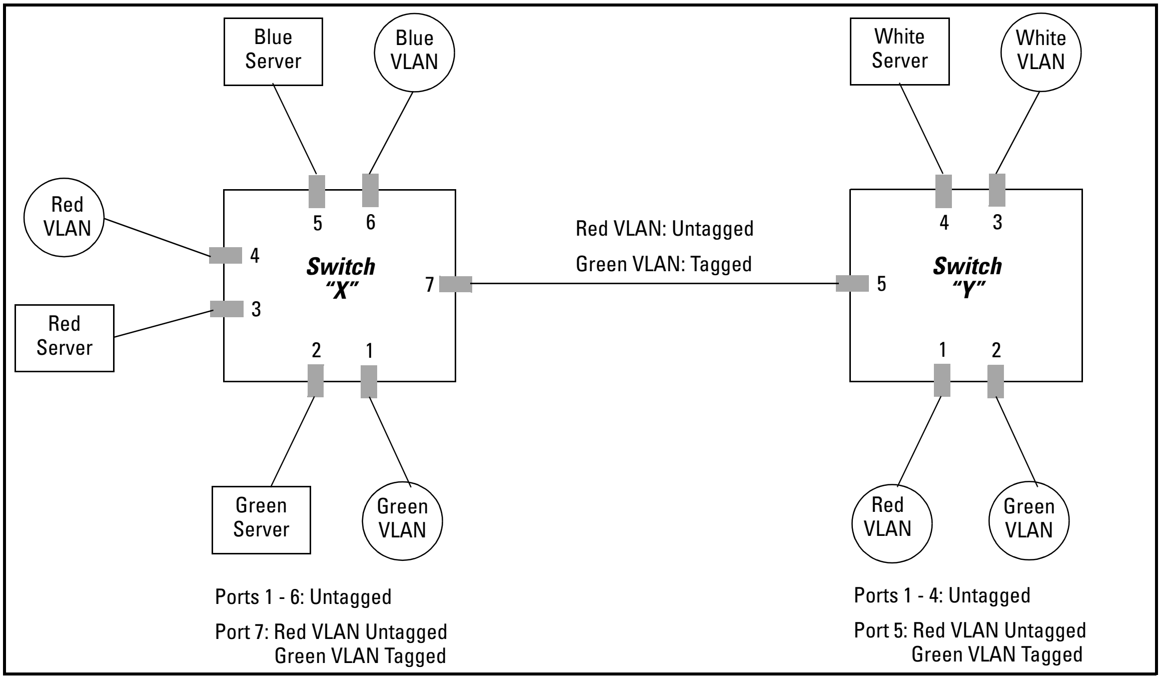


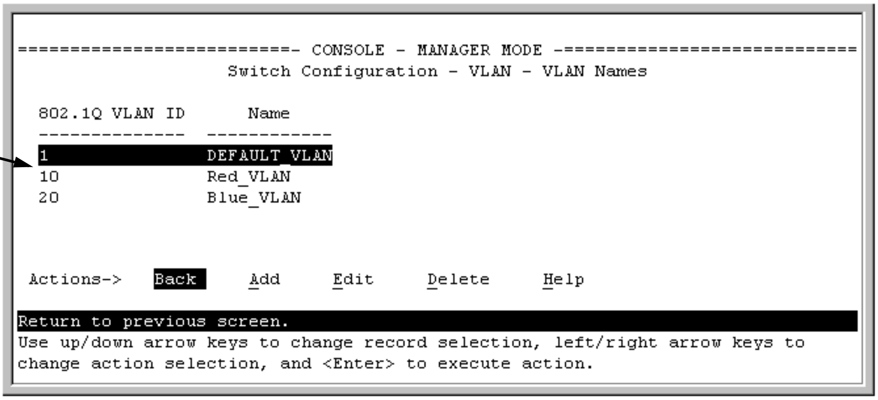
Figure 2-25. Example of Tagged and Untagged VLAN Port Assignments

- In switch X:
  - VLANs assigned to ports X1 - X6 can all be untagged because there is only one VLAN assignment per port. Red VLAN traffic will go out only the Red ports; Green VLAN traffic will go out only the Green ports, and so on. Devices connected to these ports do not have to be 802.1Q-compliant.
  - However, because both the Red VLAN and the Green VLAN are assigned to port X7, at least one of the VLANs must be tagged for this port.
- In switch Y:
  - VLANs assigned to ports Y1 - Y4 can all be untagged because there is only one VLAN assignment per port. Devices connected to these ports do not have to be 802.1Q-compliant.
  - Because both the Red VLAN and the Green VLAN are assigned to port Y5, at least one of the VLANs must be tagged for this port.
- In both switches: The ports on the link between the two switches must be configured the same. As shown in figure 2-25 (above), the Red VLAN must be untagged on port X7 and Y5 and the Green VLAN must be tagged on port X7 and Y5, or vice-versa.

Note

Each 802.1Q-compliant VLAN must have its own unique VID number, and that VLAN *must* be given the same VID in every device in which it is configured. That is, if the Red VLAN has a VID of 10 in switch X, then 10 must also be used for the Red VID in switch Y.

VID Numbers



```
==========================- CONSOLE - MANAGER MODE -============================
                       Switch Configuration - VLAN - VLAN Names

  802.1Q VLAN ID       Name
  --------------   ------------
  1                DEFAULT_VLAN
  10               Red_VLAN
  20               Blue_VLAN



 Actions->   Back     Add      Edit      Delete      Help

Return to previous screen.
Use up/down arrow keys to change record selection, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Figure 2-26. Example of VLAN ID Numbers Assigned in the VLAN Names Screen

VLAN tagging gives you several options:

- Since the purpose of VLAN tagging is to allow multiple VLANs on the same port, any port that has only one VLAN assigned to it can be configured as "Untagged" (the default) if the authorized inbound traffic for that port arrives untagged.
- Any port with two or more VLANs of the same type can have one such VLAN assigned as "Untagged". All other VLANs of the same type must be configured as "Tagged". That is:

Port-Based VLANs	Protocol VLANs
A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged.	A port can be an untagged member of one protocol-based VLAN of each protocol type. When assigning a port to multiple, protocol-based VLANs sharing the same type, the port can be an untagged member of only one such VLAN.
A port can be a tagged member of any port-based VLAN. See above.	A port can be a tagged member of any protocol-based VLAN. See above.
Note: A given VLAN ***must*** have the same VID on all 802.1Q-compliant devices in which the VLAN occurs. Also, the ports connecting two 802.1Q devices should have identical VLAN configurations.	

- If all end nodes on a port comply with the 802.1Q standard and are configured to use the correct VID, then, you can configure all VLAN assignments on a port as "Tagged" if doing so either makes it easier to manage your VLAN assignments, or if the authorized, inbound traffic for all VLANs on the port will be tagged.

For a summary and flowcharts of untagged and tagged VLAN operation on inbound traffic, refer to the following under "VLAN Operating Rules" on pages 2-14 through 2-17:

- "Inbound Tagged Packets"
- "Untagged Packet Forwarding" and figure 2-7
- "Tagged Packet Forwarding" and figure 2-8

Example. In the following network, switches X and Y and servers S1, S2, and the AppleTalk server are 802.1Q-compliant. (Server S3 could also be 802.1Q-compliant, but it makes no difference for this example.) This network includes both protocol-based (AppleTalk) VLANs and port-based VLANs.

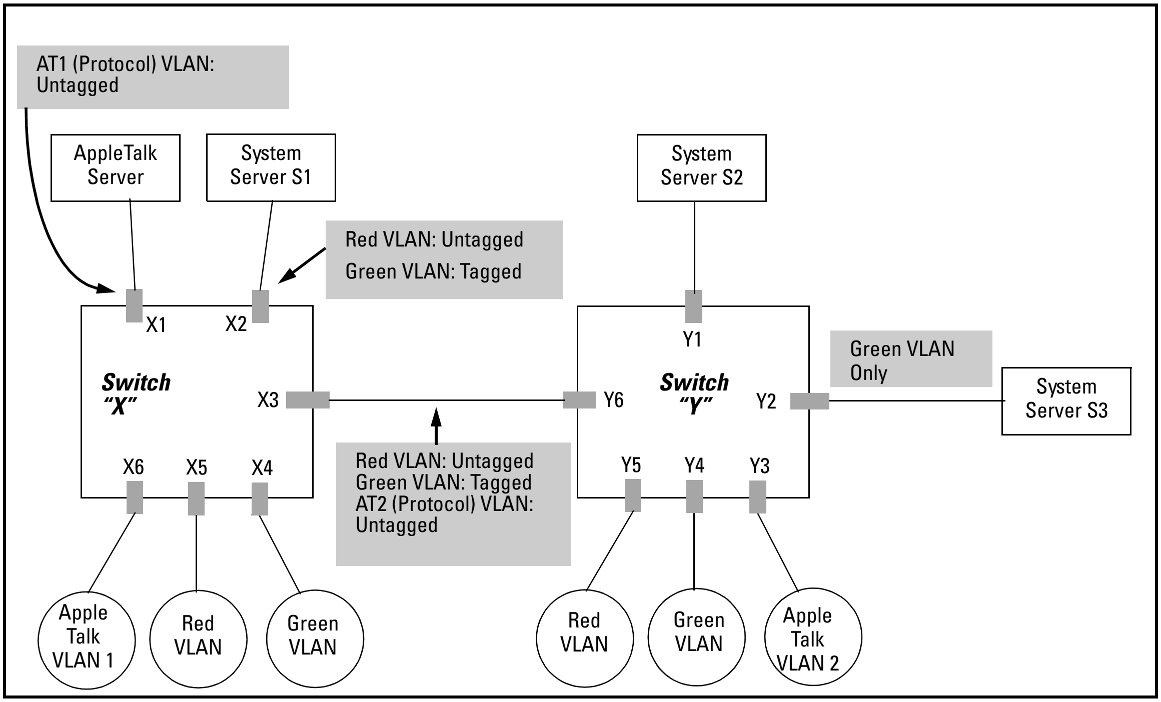


Figure 2-27. Example of Networked 802.1Q-Compliant Devices with Multiple VLANs on Some Ports

- The VLANs assigned to ports X4 - X6, Y2 - Y5 can all be untagged because there is only one VLAN assigned per port.
- Port X1 has two AppleTalk VLANs assigned, which means that one VLAN assigned to this port can be untagged and the other must be tagged.
- Ports X2 and Y1 have two port-based VLANs assigned, so one can be untagged and the other must be tagged on both ports.
- Ports X3 and Y6 have two port-based VLANs and one protocol-based VLAN assigned. Thus, one port-based VLAN assigned to this port can be untagged and the other must be tagged. Also, since these two ports share the same link, their VLAN configurations must match.

Switch X					***Switch Y***				
Port	**AT-1 VLAN**	**AT-2 VLAN**	**Red VLAN**	**Green VLAN**	**Port**	**AT-1 VLAN**	**AT-2 VLAN**	**Red VLAN**	**Green VLAN**
X1	Untagged	Tagged	No*	No*	Y1	No*	No*	Untagged	Tagged
X2	No*	No*	Untagged	Tagged	Y2	No*	No*	No*	Untagged
X3	No*	Untagged	Untagged	Tagged	Y3	No*	Untagged	No*	No*
X4	No*	No*	No*	Untagged	Y4	No*	No*	No*	Untagged
X5	No*	No*	Untagged	No*	Y5	No*	No*	Untagged	No*
X6	Untagged	No*	No*	No*	Y6	No	Untagged	Untagged	Tagged

*"No" means the port is not a member of that VLAN. For example, port X3 is not a member of the Red VLAN and does not carry Red VLAN traffic. Also, if GVRP were enabled (port-based only), "Auto" would appear instead of "No".

Note

VLAN configurations on ports connected by the same link must match. Because ports X2 and Y5 are opposite ends of the same point-to-point connection, both ports must have the same VLAN configuration; that is, both ports configure the Red VLAN as "Untagged" and the Green VLAN as "Tagged".

Special VLAN Types

VLAN Support and the Default VLAN

In the factory default configuration, VLAN support is enabled and all ports on the switch belong to the port-based, default VLAN (named DEFAULT_VLAN). This places all ports in the switch into one physical broadcast domain. In the factory-default state, the default VLAN is also the *Primary* VLAN.

You can partition the switch into multiple virtual broadcast domains by configuring one or more additional VLANs and moving ports from the default VLAN to the new VLANs. (The switch supports up to 2048 (vids numbered up to 4094) static and dynamic VLANs.) You can change the name of the default VLAN, but you cannot change the default VLAN's VID (which is always "1"). Although you can remove all ports from the default VLAN (by placing them in another port-based VLAN), this VLAN is always present; that is, you cannot delete it from the switch.

For details on port VLAN settings, refer to "Configuring Static VLAN Per-Port Settings" on page 2-37

The Primary VLAN

Because certain features and management functions run on only one VLAN in the switch, and because DHCP and Bootp can run per-VLAN, there is a need for a dedicated VLAN to manage these features and ensure that multiple instances of DHCP or Bootp on different VLANs do not result in conflicting configuration values for the switch. The *Primary* VLAN is the VLAN the switch uses to run and manage these features and data. In the factory-default configuration, the switch designates the default VLAN (DEFAULT_VLAN; VID = 1) as the Primary VLAN. However, to provide more control in your network, you can designate another static, port-based VLAN as primary. To summarize, *designating a non-default VLAN as primary* means that:

- The switch reads DHCP responses on the Primary VLAN instead of on the default VLAN. (This includes such DHCP-resolved parameters as the TimeP server address, Default TTL, and IP addressing—including the Gateway IP address—when the switch configuration specifies DHCP as the source for these values.)
- The default VLAN continues to operate as a standard VLAN (except, as noted above, you cannot delete it or change its VID).

- Any ports not specifically assigned to another VLAN will remain assigned to the Default VLAN, regardless of whether it is the Primary VLAN.

Candidates for Primary VLAN include any static, port-based VLAN currently configured on the switch. (Protocol-Based VLANs and dynamic—GVRP-learned—VLANs that have not been converted to a static VLAN cannot be the Primary VLAN.) To display the current Primary VLAN, use the CLI **show vlan** command.

Note

If you configure a non-default VLAN as the Primary VLAN, you cannot delete that VLAN unless you first select a different VLAN to serve as primary.

If you manually configure a gateway on the switch, it ignores any gateway address received via DHCP or Bootp.

To change the Primary VLAN configuration, refer to "Changing the Primary VLAN" on page 2-34.

The Secure Management VLAN

Configuring a secure Management VLAN creates an isolated network for managing the ProCurve switches that support this feature. (As of December, 2005, the Secure Management VLAN feature is available on these ProCurve switches:

- Series 6400cl switches
- Switch 6200yl
- Switch 6108
- Series 5400zl switches
- Series 5300xl switches
- Series 4200vl switches
- Series 4100gl switches
- Series 3500yl switches
- Series 3400cl switches
- Series 2800 switches
- Series 2600 switches

If you configure a Secure Management VLAN, access to the VLAN and to the switch's management functions (Menu, CLI, and web browser interface) is available only through ports configured as members.

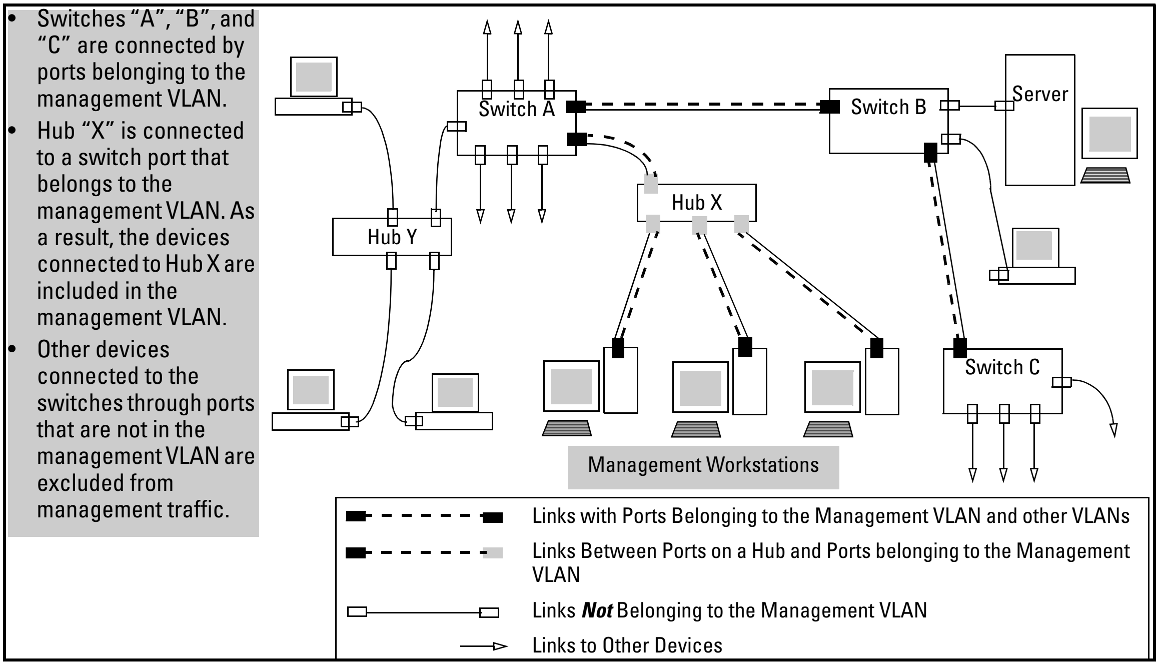
- Multiple ports on the switch can belong to the Management VLAN. This allows connections for multiple management stations you want to have access to the Management VLAN, while at the same time allowing Management VLAN links between switches configured for the same Management VLAN.

- Only traffic from the Management VLAN can manage the switch, which means that only the workstations and PCs connected to ports belonging to the Management VLAN can manage and reconfigure the switch.

Figure 2-28 illustrates use of the Management VLAN feature to support management access by a group of management workstations.

Note

The Secure Management VLAN must be a static, port-based VLAN with a manually configured IP address and subnet mask. (The switch does not allow the Management VLAN to acquire IP addressing through DHCP/Bootp.)



- Switches “A”, “B”, and “C” are connected by ports belonging to the management VLAN.
- Hub “X” is connected to a switch port that belongs to the management VLAN. As a result, the devices connected to Hub X are included in the management VLAN.
- Other devices connected to the switches through ports that are not in the management VLAN are excluded from management traffic.

Switch A
Switch B
Server
Hub X
Hub Y
Switch C
Management Workstations
Links with Ports Belonging to the Management VLAN and other VLANs
Links Between Ports on a Hub and Ports belonging to the Management VLAN
Links *Not* Belonging to the Management VLAN
Links to Other Devices

Figure 2-28. Example of Potential Security Breaches

In figure 2-29, Workstation 1 has management access to all three switches through the Management VLAN, while the PCs do not. This is because configuring a switch to recognize a Management VLAN automatically excludes attempts to send management traffic from any other VLAN.

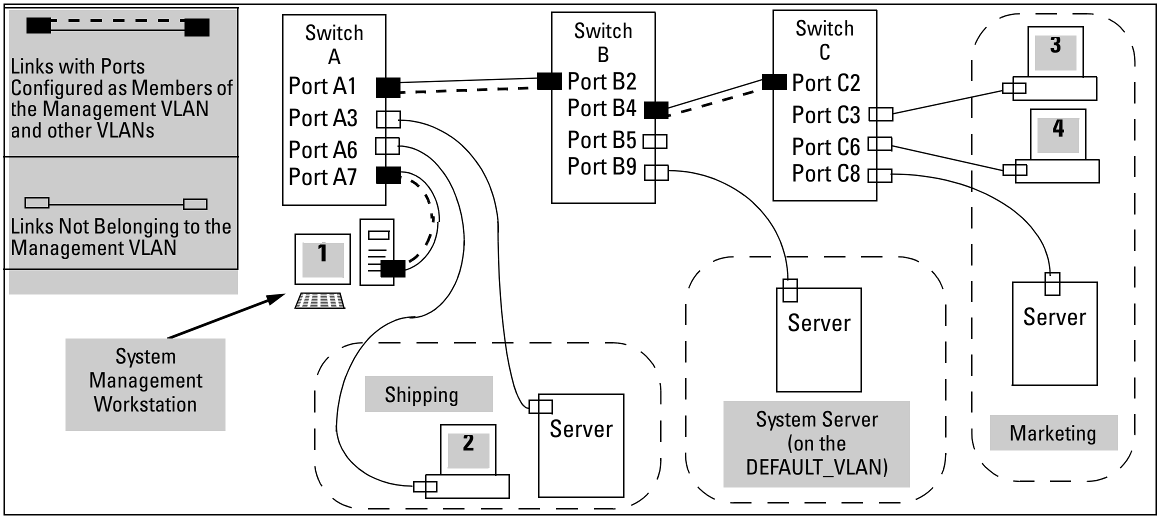

Links with Ports Configured as Members of the Management VLAN and other VLANs
Links Not Belonging to the Management VLAN
System Management Workstation
Switch A
Port A1
Port A3
Port A6
Port A7
Switch B
Port B2
Port B4
Port B5
Port B9
Switch C
Port C2
Port C3
Port C6
Port C8
1
2
3
4
Shipping
Server
Server
System Server (on the DEFAULT_VLAN)
Server
Marketing

Figure 2-29. Example of Management VLAN Control in a LAN

Table 2-7. VLAN Membership in Figure 2-29

Switch	A1	A3	A6	A7	B2	B4	B5	B9	C2	C3	C6	C8
Management VLAN (VID = 7)	**Y**	N	N	**Y**	**Y**	**Y**	N	N	**Y**	N	N	N
Marketing VLAN (VID = 12)	N	N	N	N	N	N	N	N	N	**Y**	**Y**	**Y**
Shipping Dept. VLAN (VID = 20)	N	**Y**	**Y**	N	N	N	N	N	N	N	N	N
DEFAULT-VLAN (VID = 1)	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**

Preparation

1. Determine a VID and VLAN name suitable for your Management VLAN.

   (You must manually configure the IP addressing for the Management VLAN. The switch does not allow the Management VLAN to acquire an IP address through DHCP/Bootp.)

2. Plan your Management VLAN topology to use ProCurve switches that support this feature. (Refer to page 2-46.) The ports belonging to the Management VLAN should be only the following:

   - Ports to which you will connect authorized management stations (such as Port A7 in figure 2-29.)
   - Ports on one switch that you will use to extend the Management VLAN to ports on other ProCurve switches (such as ports A1 and B2 or B4 and C2 in figure 2-29 on page 2-48.).

Hubs dedicated to connecting management stations to the Management VLAN can also be included in the above topology. Note that any device connected to a hub in the Management VLAN will also have Management VLAN access.

3. Configure the Management VLAN on the selected switch ports.
4. Test the management VLAN from all of the management stations authorized to use the Management VLAN, including any SNMP-based network management stations. Ensure that you include testing any Management VLAN links between switches.

Note

If you configure a Management VLAN on a switch by using a Telnet connection through a port that is not in the Management VLAN, then you will lose management contact with the switch if you log off your Telnet connection or execute **write memory** and **reboot** the switch.

Configuration

Syntax: [no] management-vlan < *vlan-id | vlan-name* >

Configures an existing VLAN as the management VLAN. The **no** *form disables the management VLAN and returns the switch to its default management operation. Default: Disabled. In this case, the VLAN returns to standard VLAN operation.*

For example, suppose you have already configured a VLAN named **My_VLAN** with a VID of 100. Now you want to configure the switch to do the following:

- Use **My_VLAN** as a Management VLAN (tagged, in this case) to connect port A1 on switch "A" to a management station. (The management station includes a network interface card with 802.1Q tagged VLAN capability.)
- Use port A2 to extend the Management VLAN to port B1 (which is already configured as a tagged member of **My_VLAN**) on an adjacent Procurve switch that supports the Management VLAN feature.

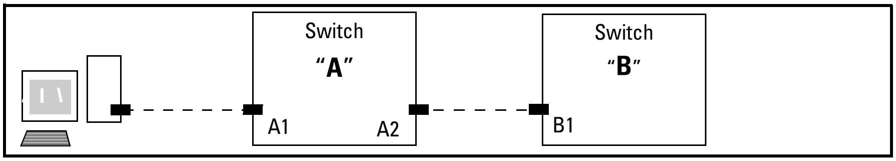


Figure 2-30. Illustration of Configuration Example

```
ProCurve (config)# management-vlan 100
ProCurve (config)# vlan 100 tagged a1
ProCurve (config)# vlan 100 tagged a2
```

Deleting the Management VLAN

You can disable the Secure Management feature without deleting the VLAN itself. For example, either of the following commands disables the Secure Management feature in the above example:

```
ProCurve (config)# no management-vlan 100
ProCurve (config)# no management-vlan my_vlan
```

Operating Notes for Management VLANs

- Use only a static, port-based VLAN for the Management VLAN.
- The Management VLAN does not support IGMP operation.
- Routing between the Management VLAN and other VLANs is not allowed.
- If there are more than 25 VLANs configured on the switch, reboot the switch after configuring the management VLAN.
- If you implement a Management VLAN in a switch mesh environment, all meshed ports on the switch will be members of the Management VLAN.
- Only one Management-VLAN can be active in the switch. If one Management-VLAN VID is saved in the startup-config file and you configure a different VID in the running-config file, the switch uses the running-config version until you either use the **write-memory** command or reboot the switch.
- During a Telnet session to the switch, if you configure the Management-VLAN to a VID that excludes the port through which you are connected to the switch, you will continue to have access only until you terminate the session by logging out or rebooting the switch.
- During a web browser session to the switch, if you configure the Management-VLAN to a VID that excludes the port through which you are connected to the switch, you will continue to have access only until you close the browser session or rebooting the switch.

Note

The Management-VLAN feature does not control management access through a direct connection to the switch's serial port.

- Enabling Spanning Tree where there are multiple links using separate VLANs, including the Management VLAN, between a pair of switches, Spanning Tree will force the blocking of one or more links. This may include the link carrying the Management VLAN, which will cause loss of management access to some devices. This can also occur where meshing is configured and the Management VLAN is configured on a separate link.

- Monitoring Shared Resources: The Management VLAN feature shares internal switch resources with several other features. The switch provides ample resources for all features. However, if the internal resources become fully subscribed, the Management VLAN feature cannot be configured until the necessary resources are released from other uses. For information on determining the current resource availability and usage, refer to the appendix titled "Monitoring Resources" in the *Management and Configuration Guide* for your switch.

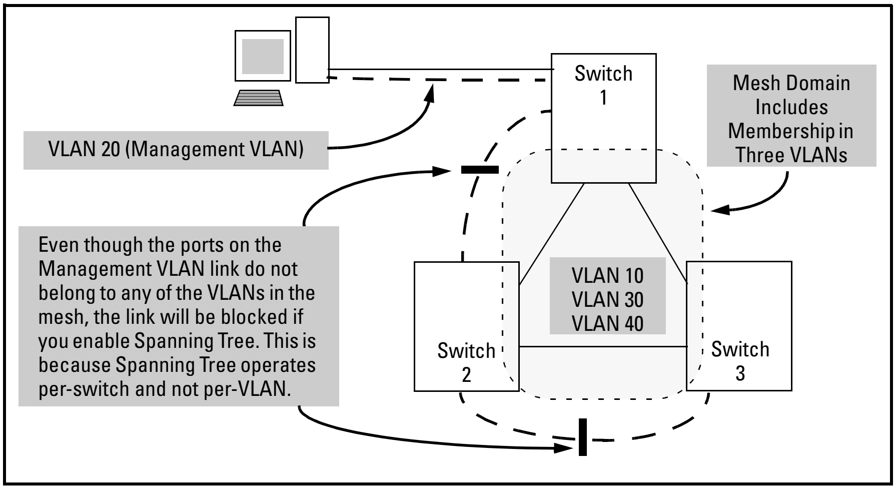

Switch 1
Mesh Domain Includes Membership in Three VLANs
VLAN 20 (Management VLAN)
Even though the ports on the Management VLAN link do not belong to any of the VLANs in the mesh, the link will be blocked if you enable Spanning Tree. This is because Spanning Tree operates per-switch and not per-VLAN.
VLAN 10
VLAN 30
VLAN 40
Switch 2
Switch 3

Figure 2-31. Example of Inadvertently Blocking a Management VLAN Link by Implementing Spanning Tree

Voice VLANs

Configuring voice VLANs separates voice traffic from data traffic and shields your voice traffic from broadcast storms. This section describes how to configure the switch for voice VLAN operation.

Operating Rules for Voice VLANs

- You must statically configure voice VLANs. GVRP and dynamic VLANs do not support voice VLAN operation.
- Configure all ports in a voice VLAN as tagged members of the VLAN. This ensures retention of the QoS (Quality of Service) priority included in voice VLAN traffic moving through your network.
- If a telephone connected to a voice VLAN includes a data port used for connecting other networked devices (such as PCs) to the network, then you must configure the port as a tagged member of the voice VLAN and a tagged or untagged member of the data VLAN you want the other networked device to use.

Components of Voice VLAN Operation

- **Voice VLAN(s):** Configure one or more voice VLANs on the switch. Some reasons for having multiple voice VLANs include:
  - Employing telephones with different VLAN requirements
  - Better control of bandwidth usage
  - Segregating telephone groups used for different, exclusive purposes

  Where multiple voice VLANs exist on the switch, you can use routing to communicate between telephones on different voice VLANs. .
- **Tagged/Untagged VLAN Membership:** If the appliances using a voice VLAN transmit tagged VLAN packets, then configure the member ports as tagged members of the VLAN. Otherwise, configure the ports as untagged members.

Voice VLAN QoS Prioritizing (Optional)

Without configuring the switch to prioritize voice VLAN traffic, one of the following conditions applies:

- If the ports in a voice VLAN are not tagged members, then the switch forwards all traffic on that VLAN at “normal” priority.
- If the ports in a voice VLAN are tagged members, then the switch forwards all traffic on that VLAN at whatever priority the traffic has when received inbound on the switch.

Using the switch’s QoS VLAN-ID (VID) Priority option, you can change the priority of voice VLAN traffic moving through the switch. If all port memberships on the voice VLAN are tagged, the priority level you set for voice VLAN traffic is carried to the next device. With all ports on the voice VLAN configured as tagged members, you can enforce a QoS priority policy moving through the switch and through your network. To set a priority on a voice VLAN, use the following command:

Syntax: vlan < *vid* > qos priority < 0 - 7 >

The qos priority default setting is 0 (normal), with 1 as the lowest priority and 7 as the highest priority.

For example, if you configured a voice VLAN with a VID of 10, and wanted the highest priority for all traffic on this VLAN, you would execute the following command:

```
ProCurve(config) # vlan 10 qos priority 7
ProCurve (config) # write memory
```

Note that you also have the option of resetting the DSCP (DiffServe Codepoint) on tagged voice VLAN traffic moving through the switch. For more on this and other QoS topics, refer to the chapter titled "Quality of Service (QoS): Managing Bandwidth More Effectively" in this guide.

Voice VLAN Access Security

You can use port security configured on an individual port or group of ports in a voice VLAN. That is, you can allow or deny access to a phone having a particular MAC address. Refer to chapter titled "Configuring and Monitoring Port Security" in the *Access Security Guide* for your switch.

Note MAC authentication is not recommended in voice VLAN applications.

Effect of VLANs on Other Switch Features

Spanning Tree Operation with VLANs

Depending on the spanning-tree option configured on the switch, the spanning-tree feature may operate as a single instance across all ports on the switch (regardless of VLAN assignments) or multiple instance on a per-VLAN basis. For single-instance operation, this means that if redundant physical links exist between the switch and another 802.1Q device, all but one link will be blocked, regardless of whether the redundant links are in separate VLANs. In this case you can use port trunking to prevent Spanning Tree from unnecessarily blocking ports (and to improve overall network performance). For multiple-instance operation, physically redundant links belonging to different VLANs can remain open. Refer to chapter 4, "Multiple Instance Spanning-Tree Operation" .

Note that Spanning Tree operates differently in different devices. For example, in the (obsolete, non-802.1Q) ProCurve Switch 2000 and the ProCurve Switch 800T, Spanning Tree operates on a per-VLAN basis, allowing redundant physical links as long as they are in separate VLANs.

IP Interfaces

There is a one-to-one relationship between a VLAN and an IP network interface. Since the VLAN is defined by a group of ports, the state (up/down) of those ports determines the state of the IP network interface associated with that VLAN. When a port-based VLAN or an IPv4 or IPv6 protocol-based VLAN comes up because one or more of its ports is up, the IP interface for that VLAN is also activated. Likewise, when a VLAN is deactivated because all of its ports are down, the corresponding IP interface is also deactivated.

VLAN MAC Address

The switches covered by this guide have one unique MAC address for all of their VLAN interfaces. You can send an 802.2 test packet to this MAC address to verify connectivity to the switch. Likewise, you can assign an IP address to the VLAN interface, and when you Ping that address, ARP will resolve the IP address to this single MAC address. In a topology where a switch has multiple VLANs and must be connected to a device having a single forwarding database, such as the Switch 4000M, some cabling restrictions apply. For more on this topic, refer to “Multiple VLAN Considerations” on page 2-18.

Port Trunks

When assigning a port trunk to a VLAN, all ports in the trunk are automatically assigned to the same VLAN. You cannot split trunk members across multiple VLANs. Also, a port trunk is tagged, untagged, or excluded from a VLAN in the same way as for individual, untrunked ports.

Port Monitoring

If you designate a port on the switch for network monitoring, this port will appear in the Port VLAN Assignment screen and can be configured as a member of any VLAN. For information on how broadcast, multicast, and unicast packets are tagged inside and outside of the VLAN to which the monitor port is assigned, refer to the section titled “VLAN-Related Problems” in the “Troubleshooting” appendix of the *Management and Configuration Guide* for your switch.

Jumbo Packet Support

Jumbo packet support is enabled per-VLAN and applies to all ports belonging to the VLAN. For more information, refer to the chapter titled “Port Traffic Controls” in the *Management and Configuration Guide* for your switch.

VLAN Restrictions

- A port must be a member of at least one VLAN. In the factory default configuration, all ports are assigned to the default VLAN (DEFAULT_VLAN; VID = 1).
- A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged. (The "Untagged" designation enables VLAN operation with non 802.1Q-compliant devices.)
- A port can be an untagged member of one protocol-based VLAN of each protocol type. When assigning a port to multiple, protocol-based VLANs sharing the same type, the port can be an untagged member of only one such VLAN.
- With routing enabled on the switch, the switch can route traffic between:
  - Multiple, port-based VLANs
  - A port-based VLAN and an IPv4 protocol-based VLAN
  - A port-based VLAN and an IPv6 protocol-based VLAN
  - An IPv4 protocol-based VLAN and an IPv6 protocol VLAN.

  Other, routable, protocol-based VLANs must use an external router to move traffic between VLANs. With routing disabled, all routing between VLANs must be through an external router.
- Prior to deleting a static VLAN, you must first re-assign all ports in the VLAN to another VLAN. You can use the **no vlan <** ***vid*** **>** command to delete a static VLAN. For more information, refer to "Creating a New Static VLAN (Port-Based or Protocol-Based) Changing the VLAN Context Level" on page 2-35.

—This page is intentionally unused—

GVRP

Contents

Overview

This chapter describes GVRP and how to configure it with the switch's built-in interfaces, and assumes an understanding of VLANs, which are described in chapter 2, "Static Virtual LANs (VLANs)" .

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

- Chapter 3, "Using the Menu Interface"
- Chapter 4, "Using the Command Line Interface (CLI)"
- Chapter 5, "Using the Web Browser Interface
- Chapter 6, "Switch Memory and Configuration"

Introduction

Feature	Default	Menu	CLI	Web
view GVRP configuration	n/a	page 3-13	page 3-14	page 3-18
list static and dynamic VLANs on a GVRP-enabled switch	n/a	—	page 3-16	page 3-18
enable or disable GVRP	disabled	page 3-13	page 3-15	page 3-18
enable or disable GVRP on individual ports	enabled	page 3-13	page 3-15	—
control how individual ports handle advertisements for new VLANs	Learn	page 3-13	page 3-15	page 3-18
convert a dynamic VLAN to a static VLAN	n/a	—	page 3-17	—
configure static VLANs	DEFAULT_VLAN (VID = 1)	page 2-22	page 2-28	page 2-39

GVRP—GARP VLAN Registration Protocol—is an application of the Generic Attribute Registration Protocol—GARP. GVRP is defined in the IEEE 802.1Q standard, and GARP is defined in the IEEE 802.1D-1998 standard.

Note

To understand and use GVRP you must have a working knowledge of 802.1Q VLAN tagging. (Refer to chapter 2, “Static Virtual LANs (VLANs)” .)

GVRP uses “GVRP Bridge Protocol Data Units” (“GVRP BPDUs”) to “advertise” static VLANs. In this manual, a GVRP BPDU is termed an *advertisement*. Advertisements are sent outbound from ports on a switch to the devices directly connected to those ports.

While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.

GVRP enables the switch to dynamically create 802.1Q-compliant VLANs on links with other devices running GVRP. This enables the switch to automatically create VLAN links between GVRP-aware devices. (A GVRP link can include intermediate devices that are not GVRP-aware.) This operation reduces the chances for errors in VLAN configuration by automatically providing VLAN ID (VID) consistency across the network. That is, you can use GVRP to propagate VLANs to other GVRP-aware devices instead of manually having to set up VLANs across your network. After the switch creates a dynamic VLAN, you can optionally use the CLI **static <*vlan-id*>** command to convert it to a static VLAN or allow it to continue as a dynamic VLAN for as long as needed. You can also use GVRP to dynamically enable port membership in static VLANs configured on a switch.

Note: On the switches covered in this guide, GVRP can be enabled only if **max vlans** is set to no more than 256 VLANs.

General Operation

When GVRP is enabled on a switch, the VID for any static VLANs configured on the switch is *advertised* (using BPDUs—Bridge Protocol Data Units) out all ports, regardless of whether a port is up or assigned to any particular VLAN. A GVRP-aware port on another device that receives the advertisements over a link can dynamically join the advertised VLAN.

A dynamic VLAN (that is, a VLAN learned through GVRP) is tagged on the port on which it was learned. Also, a GVRP-enabled port can forward an advertisement for a VLAN it learned about from other ports on the same switch (internal source), but the forwarding port will not itself join that VLAN until an advertisement for that VLAN is received through a link from another device (external source) on that specific port

Operating Note: When a GVRP-aware port on a switch learns a VID through GVRP from another device, the switch begins advertising that VID out all of its ports except the port on which the VID was learned.

Core switch with static VLANs (VID= 1, 2, & 3). Port 2 is a member of VIDs 1, 2, & 3.

1. Port 2 advertises VIDs 1, 2, & 3.

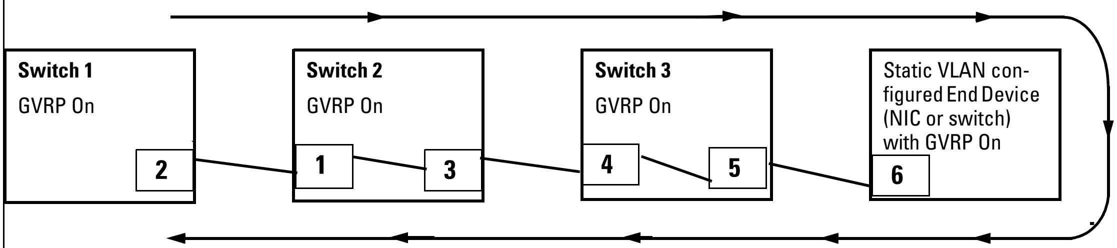
2. Port 1 receives advertisement of VIDs 1, 2, & 3 AND becomes a member of VIDs 1, 2, & 3.

3. Port 3 advertises VIDs 1, 2, & 3, but port 3 is NOT a member of VIDs 1, 2, & 3 at this point.

4. Port 4 receives advertisement of VIDs 1, 2, & 3 AND becomes a member of VIDs 1, 2, & 3.

5. Port 5 advertises VIDs 1, 2, & 3, but port 5 is NOT a member of VIDs 1, 2, & 3 at this point.

Port 6 is statically configured to be a member of VID 3.



11. Port 2 receives advertisement of VID 3. (Port 2 is already statically configured for VID 3.)

9. Port 3 receives advertisement of VID 3 AND becomes a member of VID 3. (Still not a member of VIDs 1 & 2.)

10. Port 1 advertises VID 3.

7. Port 5 receives advertisement of VID 3 AND becomes a member of VID 3. (Still not a member of VIDs 1 & 2.)

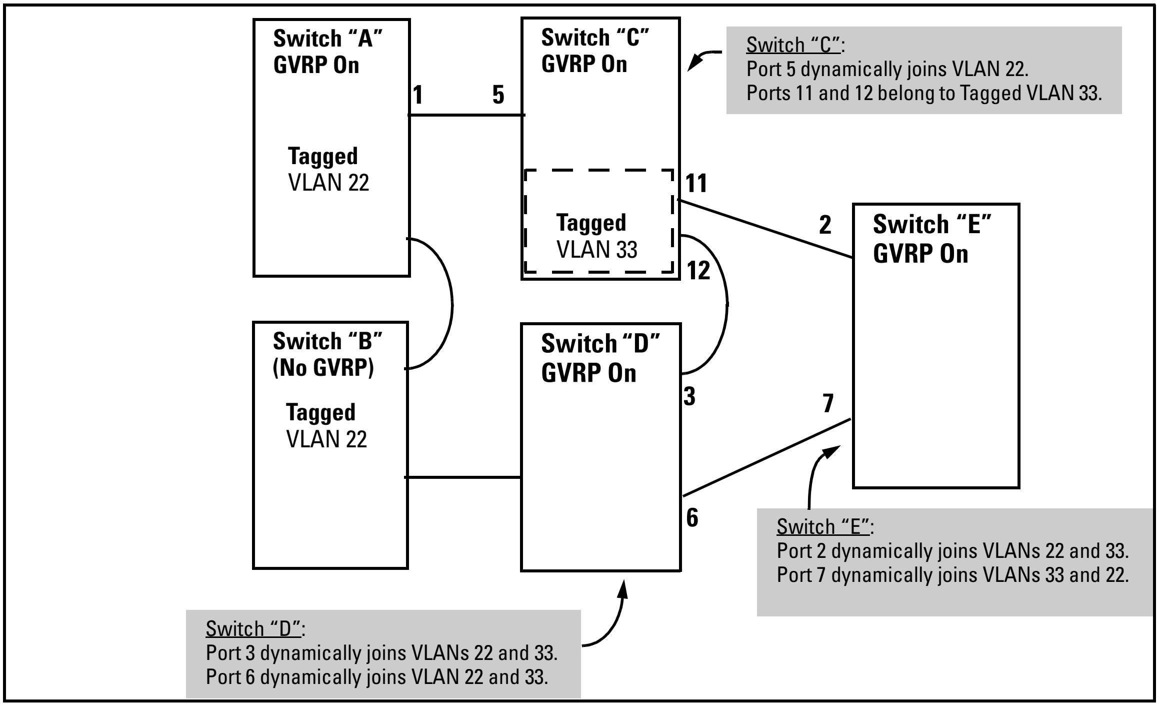
8. Port 4 advertises VID 3.

6. Port 6 advertises VID 3.

Figure 3-1. .Example of Forwarding Advertisements and Dynamic Joining

Note that if a static VLAN is configured on at least one port of a switch, and that port has established a link with another device, then all other ports of that switch will send advertisements for that VLAN.

For example, in the following figure, Tagged VLAN ports on switch “A” and switch “C” advertise VLANs 22 and 33 to ports on other GVRP-enabled switches that can dynamically join the VLANs.



Switch “C”:
Port 5 dynamically joins VLAN 22.
Ports 11 and 12 belong to Tagged VLAN 33.

Switch “E”:
Port 2 dynamically joins VLANs 22 and 33.
Port 7 dynamically joins VLANs 33 and 22.

Switch “D”:
Port 3 dynamically joins VLANs 22 and 33.
Port 6 dynamically joins VLAN 22 and 33.

Figure 3-2. Example of GVRP Operation

Note

A port can learn of a dynamic VLAN through devices that are not aware of GVRP (Switch “B”, above). VLANs must be disabled in GVRP-unaware devices to allow tagged packets to pass through.

A GVRP-aware port receiving advertisements has these options:

- If there is not already a static VLAN with the advertised VID on the receiving port, then dynamically create the VLAN and become a member.
- If the switch already has a static VLAN assignment with the same VID as in the advertisement, and the port is configured to **Auto** for that VLAN, then the port will dynamically join the VLAN and begin moving that VLAN’s traffic. (For more detail on **Auto**, see “Per-Port Options for Dynamic VLAN Advertising and Joining” on page 3-9.)
- Ignore the advertisement for that VID.
- Don’t participate in that VLAN.

Note also that a port belonging to a Tagged or Untagged static VLAN has these configurable options:

- Send VLAN advertisements, and also receive advertisements for VLANs on other ports and dynamically join those VLANs.
- Send VLAN advertisements, but ignore advertisements received from other ports.
- Avoid GVRP participation by not sending advertisements and dropping any advertisements received from other devices.

IP Addressing. A dynamic VLAN does not have an IP address, and moves traffic on the basis of port membership in VLANs. However, after GVRP creates a dynamic VLAN, you can convert it to a static VLAN. Note that it is then necessary to assign ports to the VLAN in the same way that you would for a static VLAN that you created manually. In the static state you can configure IP addressing on the VLAN and access it in the same way that you would any other static (manually created) VLAN.

Per-Port Options for Handling GVRP "Unknown VLANs"

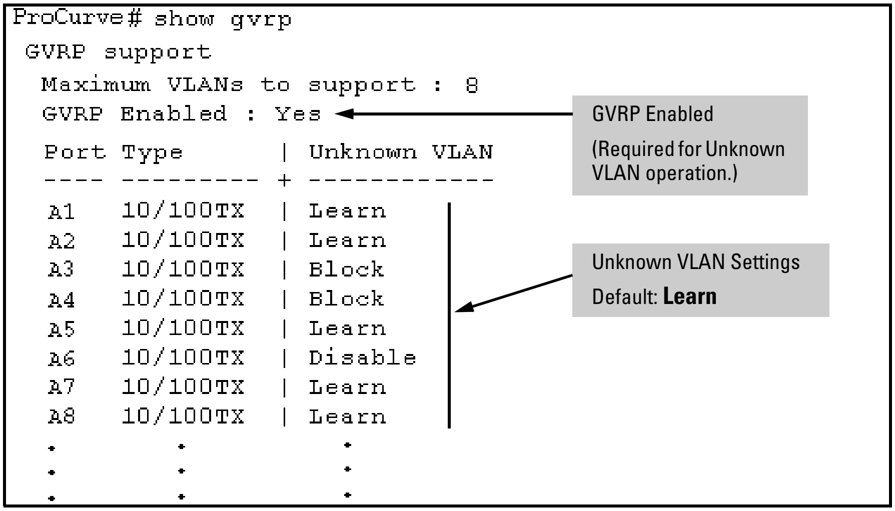
An "unknown VLAN" is a VLAN that the switch learns of by receiving an advertisement for that VLAN on a port that is not already a member of that VLAN. If the port is configured to learn unknown VLANs, then the VLAN is dynamically created and the port becomes a tagged member of the VLAN. For example, suppose that in figure 3-2 (page 3-6), port 1 on switch "A" is connected to port 5 on switch "C". Because switch "A" has VLAN 22 statically configured, while switch "C" does not have this VLAN statically configured (and does not "Forbid" VLAN 22 on port 5), VLAN 22 is handled as an "Unknown VLAN" on port 5 in switch "C". Conversely, if VLAN 22 was statically configured on switch C, but port 5 was not a member, port 5 would become a member when advertisements for VLAN 22 were received from switch "A".

When you enable GVRP on a switch, you have the per-port join-request options listed in table 3-1:

Table 3-1. Options for Handling "Unknown VLAN" Advertisements:

UnknownVLAN Mode	**Operation**
Learn (the Default)	Enables the port to become a member of any unknown VLAN for which it receives an advertisement. Allows the port to advertise other VLANs that have at least one other port on the same switch as a member.
Block	Prevents the port from joining any new dynamic VLANs for which it receives an advertisement. Allows the port to advertise other VLANs that have at least one other port as a member.
Disable	Causes the port to ignore and drop all GVRP advertisements it receives and also prevents the port from sending any GVRP advertisements.

The CLI **show gvrp** command and the menu interface VLAN Support screen show a switch's current GVRP configuration, including the Unknown VLAN settings.



```
ProCurve# show gvrp
 GVRP support

  Maximum VLANs to support : 8
  GVRP Enabled : Yes

  Port Type      | Unknown VLAN
  ---- --------- + ------------
  A1   10/100TX  | Learn
  A2   10/100TX  | Learn
  A3   10/100TX  | Block
  A4   10/100TX  | Block
  A5   10/100TX  | Learn
  A6   10/100TX  | Disable
  A7   10/100TX  | Learn
  A8   10/100TX  | Learn
  .        .          .
  .        .          .
  .        .          .
```

GVRP Enabled

(Required for Unknown VLAN operation.)

Unknown VLAN Settings

Default: **Learn**

Figure 3-3. Example of GVRP Unknown VLAN Settings

Per-Port Options for Dynamic VLAN Advertising and Joining

Initiating Advertisements. As described in the preceding section, to enable dynamic joins, GVRP must be enabled and a port must be configured to Learn (the default). However, to send advertisements in your network, one or more static (**Tagged**, **Untagged**, or **Auto**) VLANs must be configured on one or more switches (with GVRP enabled), depending on your topology.

Enabling a Port for Dynamic Joins. You can configure a port to dynamically join a static VLAN. The join will then occur if that port subsequently receives an advertisement for the static VLAN. (This is done by using the **Auto** and **Learn** options described in table 3-2, on the next page.

Parameters for Controlling VLAN Propagation Behavior. You can configure an individual port to actively or passively participate in dynamic VLAN propagation or to ignore dynamic VLAN (GVRP) operation. These options are controlled by the GVRP "Unknown VLAN" and the static VLAN configuration parameters, as described in the following table:

Table 3-2. Controlling VLAN Behavior on Ports with Static VLANs

Per-Port "Unknown VLAN" (GVRP) Configuration	Static VLAN Options—Per VLAN Specified on Each Port [1]		
	Port Activity: Tagged or Untagged (Per VLAN)[2]	**Port Activity: Auto[2] (Per VLAN)**	**Port Activity: Forbid (Per VLAN)[2]**
Learn (the Default)	The port: • Belongs to specified VLAN. • Advertises specified VLAN. • Can become a member of dynamic VLANs for which it receives advertisements. • Advertises dynamic VLANs that have at least one other port (on the same switch) as a member.	The port: • Will become a member of specified VLAN if it receives advertisements for specified VLAN from another device. • Will advertise specified VLAN. • Can become a member of other, dynamic VLANs for which it receives advertisements. • Will advertise a dynamic VLAN that has at least one other port (on the same switch) as a member.	The port: 1. Will not become a member of the specified VLAN. 2. Will not advertise specified VLAN. 3. Can become a member of other dynamic VLANs for which it receives advertisements. 4. Will advertise a dynamic VLAN that has at least one other port on the same switch as a member.
Block	The port: • Belongs to the specified VLAN. • Advertises this VLAN. • Will not become a member of new dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port as a member.	The port: • Will become a member of specified VLAN if it receives advertisements for this VLAN. • Will advertise this VLAN. • Will not become a member of new dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port (on the same switch) as a member.	The port: • Will not become a member of the specified VLAN. • Will not advertise this VLAN. • Will not become a member of dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port (on the same switch) as a member.
Disable	The port: • Is a member of the specified VLAN. • Will ignore GVRP PDUs. • Will not join any advertised VLANs. • Will not advertise VLANs.	The port: • Will not become a member of the specified VLAN. • Will ignore GVRP PDUs. • Will not join any dynamic VLANs. • Will not advertise VLANs.	The port: • Will not become a member of this VLAN. • Will ignore GVRP PDUs. • Will not join any dynamic VLANs. • Will not advertise VLANs.

[1] Each port of the switch must be a Tagged or Untagged member of at least one VLAN. Thus, any port configured for GVRP to Learn or Block will generate and forward advertisements for static VLAN(s) configured on the switch and also for dynamic VLANs the switch learns on other ports.

[2] To configure tagging, **Auto**, or **Forbid**, see "Configuring Static VLAN Per-Port Settings" on page 2-37 (for the CLI) or "Adding or Changing a VLAN Port Assignment" on page 2-26 (for the menu).

As the preceding table indicates, when you enable GVRP, a port that has a Tagged or Untagged static VLAN has the option for both generating advertisements and dynamically joining other VLANs.

Note

In table 3-2, above, the Unknown VLAN parameters are configured on a per-port basis using the CLI. The Tagged, Untagged, Auto, and Forbid options are configured per static VLAN on every port, using either the menu interface or the CLI.

Because dynamic VLANs operate as Tagged VLANs, and because a tagged port on one device cannot communicate with an untagged port on another device, ProCurve recommends that you use Tagged VLANs for the static VLANs you will use to generate advertisements.

GVRP and VLAN Access Control

Advertisements and Dynamic Joins

When you enable GVRP on a switch, the default GVRP parameter settings allow all of the switch's ports to transmit and receive dynamic VLAN advertisements (GVRP advertisements) and to dynamically join VLANs. The two preceding sections describe the per-port features you can use to control and limit VLAN propagation. To summarize, you can:

- Allow a port to advertise and/or join dynamic VLANs (Learn mode—the default).
- Allow a port to send VLAN advertisements, but not receive them from other devices; that is, the port cannot dynamically join a VLAN but other devices can dynamically join the VLANs it advertises (Block mode).
- Prevent a port from participating in GVRP operation (Disable mode).

Port-Leave From a Dynamic VLAN

A dynamic VLAN continues to exist on a port for as long as the port continues to receive advertisements of that VLAN from another device connected to that port or until you:

- Convert the VLAN to a static VLAN (See "Converting a Dynamic VLAN to a Static VLAN" on page 3-17.)
- Reconfigure the port to **Block** or **Disable**

- Disable GVRP
- Reboot the switch

The time-to-live for dynamic VLANs is 10 seconds. That is, if a port has not received an advertisement for an existing dynamic VLAN during the last 10 seconds, the port removes itself from that dynamic VLAN.

Planning for GVRP Operation

These steps outline the procedure for setting up dynamic VLANs for a segment.

1. Determine the VLAN topology you want for each segment (broadcast domain) on your network.
2. Determine the VLANs that must be static and the VLANs that can be dynamically propagated.
3. Determine the device or devices on which you must manually create static VLANs in order to propagate VLANs throughout the segment.
4. Determine security boundaries and how the individual ports in the segment will handle dynamic VLAN advertisements. (See table 3-1 on page 3-8 and table 3-2 on page 3-10.)
5. Enable GVRP on all devices you want to use with dynamic VLANs and configure the appropriate "Unknown VLAN" parameter (**Learn**, **Block**, or **Disable**) for each port.
6. Configure the static VLANs on the switch(es) where they are needed, along with the per-VLAN parameters (**Tagged**, **Untagged**, **Auto**, and **Forbid**—see table 3-2 on page 3-10) on each port.
7. Dynamic VLANs will then appear automatically, according to the configuration options you have chosen.
8. Convert dynamic VLANs to static VLANs where you want dynamic VLANs to become permanent.

Configuring GVRP On a Switch

The procedures in this section describe how to:

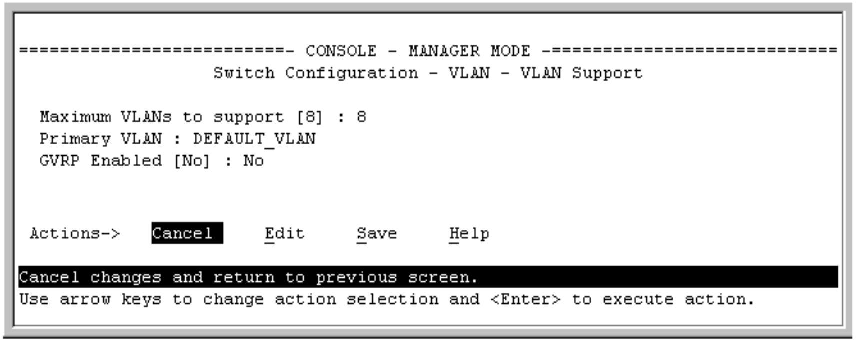
- View the GVRP configuration on a switch
- Enable and disable GVRP on a switch
- Specify how individual ports will handle advertisements

To view or configure static VLANs for GVRP operation, refer to "Per-Port Static VLAN Configuration Options" on page 2-12.

Menu: Viewing and Configuring GVRP

1. From the Main Menu, select:

   2. Switch Configuration …
   8. VLAN Menu …
   1. VLAN Support



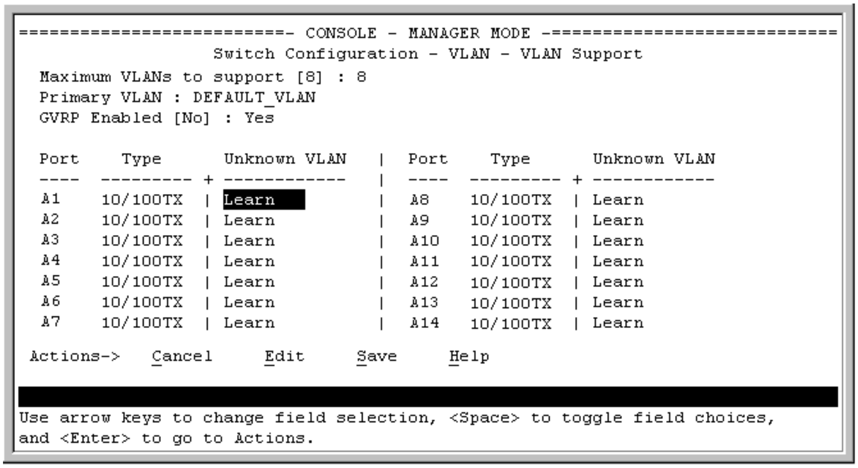
```
==========================- CONSOLE - MANAGER MODE -============================
                   Switch Configuration - VLAN - VLAN Support

  Maximum VLANs to support [8] : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : No


 Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 3-4. The VLAN Support Screen (Default Configuration)

2. Do the following to enable GVRP and display the Unknown VLAN fields:
   a. Press **[E]** (for **Edit**).
   b. Use [↓] to move the cursor to the **GVRP Enabled** field.
   c. Press the Space bar to select **Yes**.
   d. Press [↓] again to display the **Unknown VLAN** fields.

The Unknown VLAN fields enable you to configure each port to:

- Learn - Dynamically join any advertised VLAN and advertise all VLANs learned through other ports.
- Block - Do not dynamically join any VLAN, but still advertise all VLANs learned through other ports.
- Disable - Ignore and drop all incoming advertisements and do not transmit any advertisements.



```
=========================- CONSOLE - MANAGER MODE -============================
                   Switch Configuration - VLAN - VLAN Support
  Maximum VLANs to support [8] : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : Yes

  Port    Type      Unknown VLAN   |  Port    Type      Unknown VLAN
  ----  --------- + ------------   |  ----  --------- + ------------
  A1    10/100TX  | Learn          |  A8    10/100TX  | Learn
  A2    10/100TX  | Learn          |  A9    10/100TX  | Learn
  A3    10/100TX  | Learn          |  A10   10/100TX  | Learn
  A4    10/100TX  | Learn          |  A11   10/100TX  | Learn
  A5    10/100TX  | Learn          |  A12   10/100TX  | Learn
  A6    10/100TX  | Learn          |  A13   10/100TX  | Learn
  A7    10/100TX  | Learn          |  A14   10/100TX  | Learn

 Actions->   Cancel     Edit     Save     Help


Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

Figure 3-5. Example Showing Default Settings for Handling Advertisements

3. Use the arrow keys to select the port you want, and the Space bar to select Unknown VLAN option for any ports you want to change.

4. When you finish making configuration changes, press **[Enter]**, then **[S]** (for **Save**) to save your changes to the Startup-Config file.

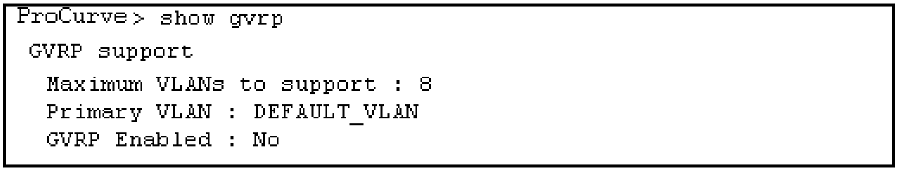
CLI: Viewing and Configuring GVRP

GVRP Commands Used in This Section

show gvrp	below
gvrp	page 3-15
unknown-vlans	page 3-15

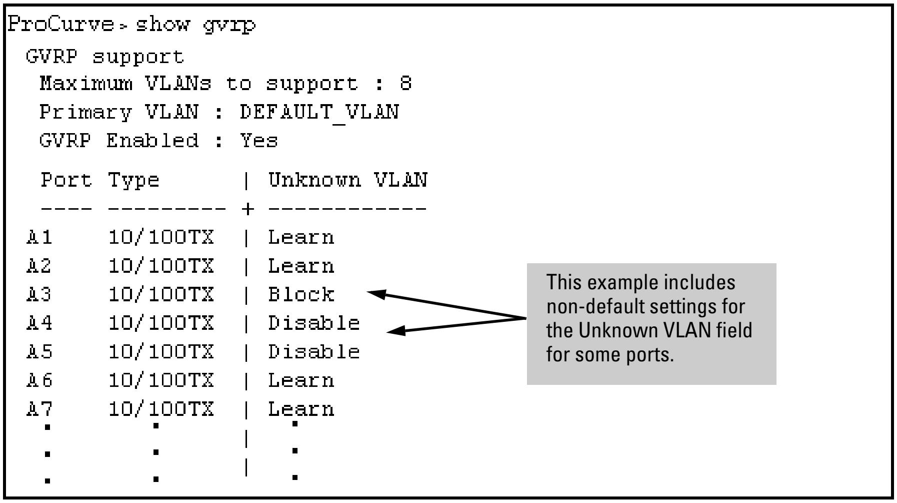
Displaying the Switch's Current GVRP Configuration. This command shows whether GVRP is disabled, along with the current settings for the maximum number of VLANs and the current Primary VLAN. (For more on the last two parameters, see chapter 2, "Static Virtual LANs (VLANs)" .)

Syntax: show gvrp *Shows the current settings.*



```
ProCurve> show gvrp
 GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : No
```

Figure 3-6. Example of "Show GVRP" Listing with GVRP Disabled



```
ProCurve> show gvrp
 GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : Yes

  Port Type      | Unknown VLAN
  ---- --------- + ------------
 A1    10/100TX  | Learn
 A2    10/100TX  | Learn
 A3    10/100TX  | Block
 A4    10/100TX  | Disable
 A5    10/100TX  | Disable
 A6    10/100TX  | Learn
 A7    10/100TX  | Learn
  .       .      |   .
  .       .      |   .
  .       .      |   .
```

This example includes non-default settings for the Unknown VLAN field for some ports.

Figure 3-7. Example of Show GVRP Listing with GVRP Enabled

Enabling and Disabling GVRP on the Switch. This command enables GVRP on the switch.

Syntax: gvrp

This example enables GVRP:

```
ProCurve(config)# gvrp
```

This example disables GVRP operation on the switch:

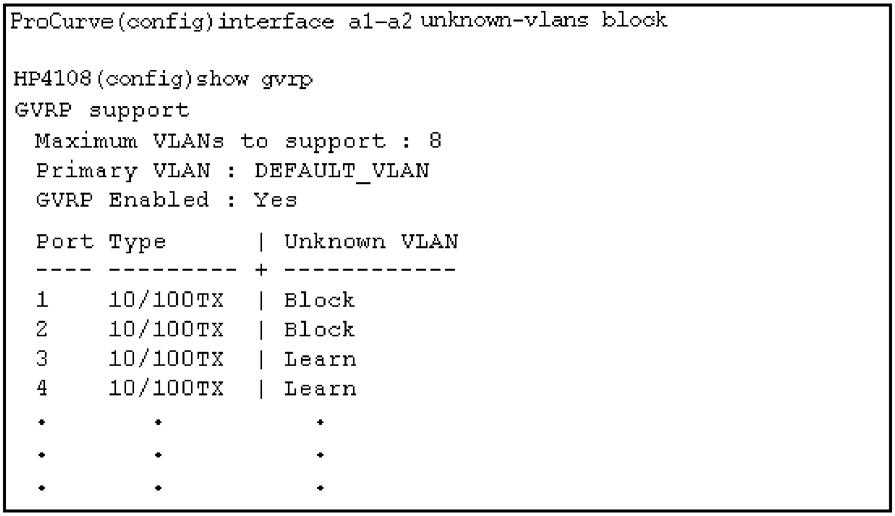
```
ProCurve(config)# no gvrp
```

Enabling and Disabling GVRP On Individual Ports. When GVRP is enabled on the switch, use the **unknown-vlans** command to change the Unknown VLAN field for one or more ports. You can use this command at either the Manager level or the interface context level for the desired port(s).

Syntax: interface < *port-list* > unknown-vlans < learn | block | disable >

Changes the Unknown VLAN field setting for the specified port(s).

For example, to change and view the configuration for ports A1-A2 to **Block**:



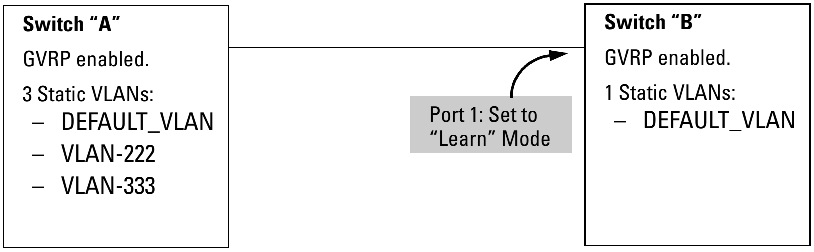
```
ProCurve(config)interface a1-a2 unknown-vlans block

HP4108(config)show gvrp
GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : Yes

  Port Type      | Unknown VLAN
  ---- --------- + ------------
  1    10/100TX  | Block
  2    10/100TX  | Block
  3    10/100TX  | Learn
  4    10/100TX  | Learn
  .       .          .
  .       .          .
  .       .          .
```

Figure 3-8. Displaying the Static and Dynamic VLANs Active on the Switch

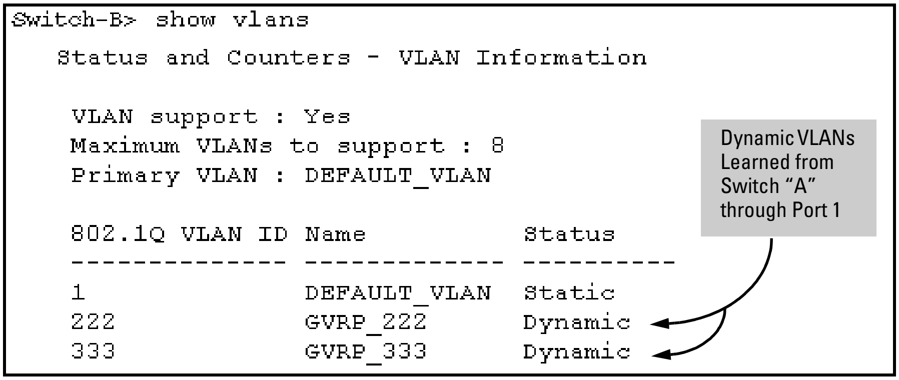
Syntax: show vlans

The **show vlans** *command lists all VLANs present in the switch.*

For example, in the following illustration, switch “B” has one static VLAN (the default VLAN), with GVRP enabled and port 1 configured to **Learn** for Unknown VLANs. Switch “A” has GVRP enabled and has three static VLANs: the default VLAN, VLAN-222, and VLAN-333. In this scenario, switch B will dynamically join VLAN-222 and VLAN-333:



The **show vlans** command lists the dynamic (and static) VLANs in switch "B" after it has learned and joined VLAN-222 and VLAN-333.



```
Switch-B> show vlans

   Status and Counters - VLAN Information

    VLAN support : Yes
    Maximum VLANs to support : 8
    Primary VLAN : DEFAULT_VLAN

    802.1Q VLAN ID Name          Status
    -------------- ------------- ----------
    1              DEFAULT_VLAN  Static
    222            GVRP_222      Dynamic
    333            GVRP_333      Dynamic
```

Dynamic VLANs Learned from Switch "A" through Port 1

Figure 3-9. Example of Listing Showing Dynamic VLANs

Converting a Dynamic VLAN to a Static VLAN. If a port on the switch has joined a dynamic VLAN, you can use the following command to convert that dynamic VLAN to a static VLAN:

Syntax: static < *dynamic-vlan-id* >

Converts the a dynamic VLAN to a static VLAN.

For example, to convert dynamic VLAN 333 (from the previous example) to a static VLAN:

```
ProCurve(config)# static 333
```

When you convert a dynamic VLAN to a static VLAN, all ports on the switch are assigned to the VLAN in Auto mode.

Web: Viewing and Configuring GVRP

To view, enable, disable, or reconfigure GVRP:

1. Click on the **Configuration** tab.
2. Click on **[VLAN Configuration]** and do the following:
   - To enable or disable GVRP, click on **GVRP Enabled**.
   - To change the Unknown VLAN field for any port:
     i. Click on **[GVRP Security]** and make the desired changes.
     ii. Click on **[Apply]** to save and implement your changes to the Unknown VLAN fields.

For web-based Help on how to use the web browser interface screen, click on the **[?]** button provided on the web browser screen.

GVRP Operating Notes

- A dynamic VLAN must be converted to a static VLAN before it can have an IP address.
- On the switches covered in this guide, GVRP can be enabled only if **max vlans** is set to no more than 256 VLANs.
- The total number of VLANs on the switch (static and dynamic combined) cannot exceed the current Maximum VLANs setting. For example, in the factory default state, the switch supports eight VLANs. Thus, in a case where four static VLANs are configured on the switch, the switch can accept up to four additional VLANs in any combination of static and dynamic. Any additional VLANs advertised to the switch will not be added unless you first increase the Maximum VLANs setting. In the Menu interface, click on **2. Switch Configuration ... | 8. VLAN Menu | 1. VLAN Support**. In the global config level of the CLI, use **max-vlans**.
- Converting a dynamic VLAN to a static VLAN and then executing the **write memory** command saves the VLAN in the startup-config file and makes it a permanent part of the switch's VLAN configuration.
- Within the same broadcast domain, a dynamic VLAN can pass through a device that is not GVRP-aware. This is because a hub or a switch that is not GVRP-ware will flood the GVRP (multicast) advertisement packets out all ports.
- GVRP assigns dynamic VLANs as Tagged VLANs. To configure the VLAN as Untagged, you must first convert it to a static VLAN.

- Rebooting a switch on which a dynamic VLAN exists deletes that VLAN. However, the dynamic VLAN re-appears after the reboot if GVRP is enabled and the switch again receives advertisements for that VLAN through a port configured to add dynamic VLANs.
- By receiving advertisements from other devices running GVRP, the switch learns of static VLANs on those other devices and dynamically (automatically) creates tagged VLANs on the links to the advertising devices. Similarly, the switch advertises its static VLANs to other GVRP-aware devices, as well as the dynamic VLANs the switch has learned.
- A GVRP-enabled switch does not advertise any GVRP-learned VLANs out of the port(s) on which it originally learned of those VLANs.
- While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.
- A VLAN enabled for jumbo traffic cannot be used to create a dynamic VLAN. A port belonging to a statically configured, jumbo-enabled VLAN cannot join a dynamic VLAN.

— This page intentionally unused. —

Multiple Instance Spanning-Tree Operation

Contents

Overview

MSTP Features

802.1s Spanning Tree Protocol	Default	Menu	CLI	Web
Viewing the MSTP Status and Configuration	n/a	—	page 4-33	—
Enable/Disable MSTP and Configure Global Parameters	Disabled	—	page 4-19	—
Configuring Basic Port Connectivity Parameters	edge-port: No mcheck: Yes hello-time: 2 path-cost: auto point-to-point MAC: Force-True priority: 128 (multiplier: 8)	—	page 4-22 and following	—
Configuring MSTP Instance Parameters	instance (MSTPI): none priority: 32768 (multiplier: 8)	—	page 4-25	—
Configuring MSTP Instance Per-Port Parameters	Auto	—	page 4-28	—
Enabling/Disabling MSTP Spanning Tree Operation	Disabled	—	page 4-31	—
Enabling an Entire MST Region at Once	n/a	—	page 4-31	—

Without spanning tree, having more than one active path between a pair of nodes causes loops in the network, which can result in duplication of messages, leading to a "broadcast storm" that can bring down the network.

Multiple-Instance spanning tree operation (802.1s) ensures that only one active path exists between any two nodes in a spanning-tree *instance.* A spanning-tree instance comprises a unique set of VLANs, and belongs to a specific spanning-tree *region.* A region can comprise multiple spanning-tree instances (each with a different set of VLANs), and allows one active path among regions in a network. Applying VLAN tagging to the ports in a multiple-instance spanning-tree network enables blocking of redundant links in one instance while allowing forwarding over the same links for non-redundant use by another instance. For example, suppose you have three switches in a region configured with VLANs grouped into two instances, as follows:

VLANs	Instance 1	Instance 2
10, 11, 12	Yes	No
20, 21, 22	No	Yes

The logical and physical topologies resulting from these VLAN/Instance groupings result in blocking on different links for different VLANs:

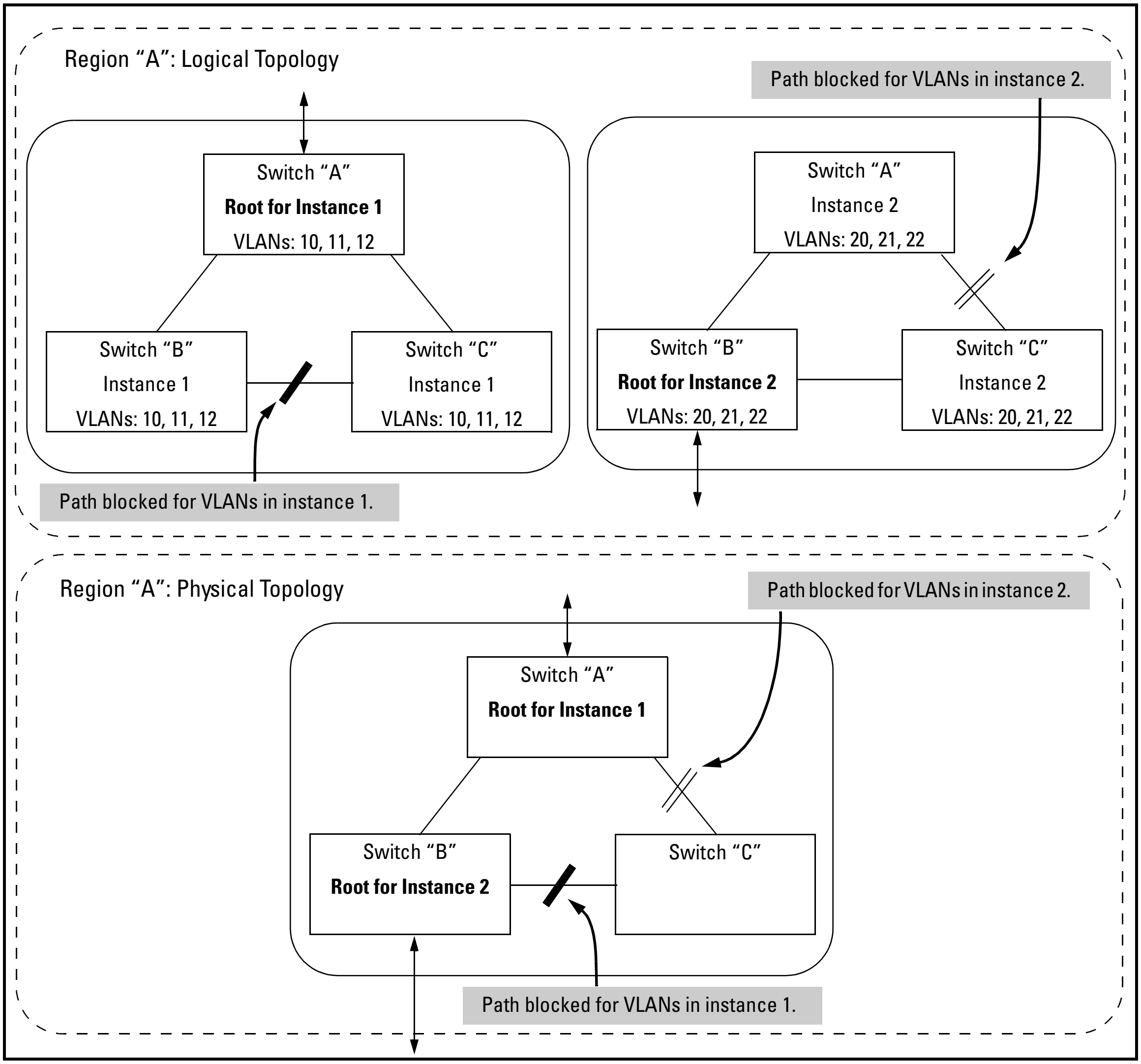


Figure 4-1. Example of a Multiple Spanning-Tree Application

Note on Path Cost

RSTP and MSTP implements a greater range of path costs and new default path cost values to account for higher network speeds. These values are different than the values defined by 802.1D STP as shown below.

Port Type	802.1D STP Path Cost	RSTP and MSTP Path Cost
10 Mbps	100	2 000 000
100 Mbps	10	200 000
1 Gbps	5	20 000

Because the maximum value for the path cost allowed by 802.1D STP is 65535, devices running that version of spanning tree cannot be configured to match the values defined by MSTP, at least for 10 Mbps and 100 Mbps ports. In LANs where there is a mix of devices running 802.1D STP, RSTP, and/or MSTP, you should reconfigure the devices so the path costs match for ports with the same network speeds.

802.1s Multiple Spanning Tree Protocol (MSTP)

The 802.1D and 802.1w spanning tree protocols operate without regard to a network's VLAN configuration, and maintain one common spanning tree throughout a bridged network. Thus, these protocols map one loop-free, logical topology on a given physical topology. The 802.1s Multiple Spanning Tree protocol (MSTP) uses VLANs to create multiple spanning trees in a network, which significantly improves network resource utilization while maintaining a loop-free environment.

While the per-VLAN spanning tree approach adopted by some vendors overcomes the network utilization problems inherent in using STP or RSTP, using a per-VLAN technology with multiple VLANs can overload the switch's CPU. MSTP on the switches covered in this guide complies with the IEEE 802.1s standard, and extends STP and RSTP functionality to map multiple independent spanning tree instances onto a physical topology. With MSTP, each spanning tree instance can include one or more VLANs and applies a separate, per-instance forwarding topology. Thus, where a port belongs to multiple VLANs, it may be dynamically blocked in one spanning tree instance, but forwarding in another instance. This achieves load-balancing across the network while keeping the switch's CPU load at a moderate level (by aggregating multiple VLANs in a single spanning tree instance). MSTP provides fault tolerance through rapid, automatic reconfiguration if there is a failure in a network's physical topology.

With MSTP-capable switches, you can create a number of MST regions containing multiple spanning tree instances. This requires the configuration of a number of MSTP-capable switches. However, it is NOT necessary to do this. You can just enable MSTP on an MSTP-capable switch and a spanning tree instance is created automatically. This instance always exists by default when spanning tree is enabled, and is the spanning tree instance that communicates with STP and RSTP environments. The MSTP configuration commands operate exactly like RSTP commands and MSTP is backward-compatible with the RSTP-enabled and STP-enabled switches in your network.

Caution

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default MSTP parameter settings are usually adequate for spanning tree operation. Also, because incorrect MSTP settings can adversely affect network performance, you should not change the MSTP settings from their default values unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default MSTP timer settings (**Hello Time** and **Forward Delay**) are usually adequate for MSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the MSTP **Hello Time** and **Forward Delay** timers can cause unnecessary topology changes and end-node connectivity problems.

For MSTP information beyond what is provided in this manual, refer to the IEEE 802.1s standard.

MSTP Structure

MSTP maps active, separate paths through separate spanning tree instances and between MST regions. Each MST region comprises one or more MSTP switches. Note that MSTP recognizes an STP or RSTP LAN as a distinct spanning-tree region.

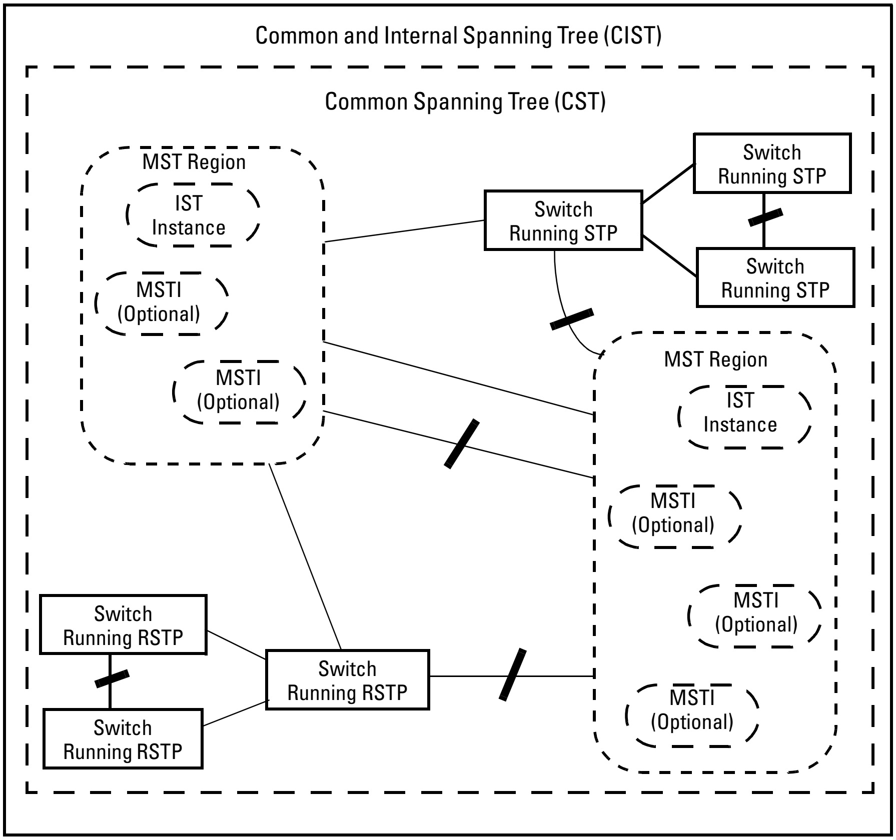


Figure 4-2. Example of MSTP Network with Legacy STP and RSTP Devices Connected

Common and Internal Spanning Tree (CIST): The CIST identifies the regions in a network and administers the CIST root bridge for the network, the root bridge for each region, and the root bridge for each spanning-tree instance in each region.

Common Spanning Tree (CST): The CST administers the connectivity among the MST regions, STP LANs, and RSTP LANs in a bridged network.

MST Region: An MST region comprises the VLANs configured on physically connected MSTP switches. All switches in a given region must be configured with the same VLANs and Multiple Spanning Tree Instances (MSTIs).

Internal Spanning Tree (IST): The IST administers the topology within a given MST region. When you configure a switch for MSTP operation, the switch automatically includes all of the static VLANs configured on the switch in a single, active spanning tree topology (instance) within the IST. This is termed the "IST instance". Any VLANs you subsequently configure on the switch are added to this IST instance. To create separate forwarding paths within a region, group specific VLANs into different Multiple Spanning Tree Instances (MSTIs). (Refer to "Multiple Spanning Tree Instance", below.)

Types of Multiple Spanning Tree Instances: A multiple spanning tree network comprises separate spanning-tree instances existing in an MST region. (There can be multiple regions in a network.) Each instance defines a single forwarding topology for an exclusive set of VLANs. By contrast, an STP or RSTP network has only one spanning tree instance for the entire network, and includes all VLANs in the network. (An STP or RSTP network operates as a single-instance network.) A region can include two types of STP instances:

- **Internal Spanning-Tree Instance (IST Instance):** This is the default spanning tree instance in any MST region. It provides the root switch for the region and comprises all VLANs configured on the switches in the region that are not specifically assigned to Multiple Spanning Tree Instances (MSTIs, described below). All VLANs in the IST instance of a region are part of the same, single spanning tree topology, which allows only one forwarding path between any two nodes belonging to any of the VLANs included in the IST instance. All switches in the region must belong to the set of VLANs that comprise the IST instance. Note that the switch automatically places dynamic VLANs (resulting from GVRP operation) in the IST instance. Dynamic VLANs cannot exist in an MSTI (described below).
- **MSTI (Multiple Spanning Tree Instance):** This type of configurable spanning tree instance comprises all static VLANs you specifically assign to it, and must include at least one VLAN. The VLAN(s) you assign to an MSTI must initially exist in the IST instance of the same MST region. When you assign a static VLAN to an MSTI, the switch removes the VLAN from the IST instance. (Thus, you can assign a VLAN to only one MSTI in a given region.) All VLANs in an MSTI operate as part of the same single spanning tree topology. (The switch does not allow dynamic VLANs in an MSTI.)

Caution

When you enable MSTP on the switch, the default MSTP spanning tree configuration settings comply with the values recommended in the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) standard. Note that inappropriate changes to these settings can result in severely degraded network performance. For this reason, *ProCurve strongly recommends that changing these default settings be reserved only for experienced network administrators who have a strong understanding of the IEEE 802.1D/w/s standards and operation.*

How MSTP Operates

In the factory default configuration, spanning tree operation is off. Also, the switch retains its currently configured spanning tree parameter settings when disabled. Thus, if you disable spanning tree, then later re-enable it, the parameter settings will be the same as before spanning tree was disabled. The switch also includes a “Pending” feature that enables you to exchange MSTP configurations with a single command. (Refer to “Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another” on page 4-31.)

Note

The switch automatically senses port identity and type, and automatically defines spanning-tree parameters for each type, as well as parameters that apply across the switch. Although these parameters can be adjusted, *ProCurve strongly recommends leaving these settings in their default configurations unless the proposed changes have been supplied by an experienced network administrator who has a strong understanding of the IEEE 802.1D/w/s standards and operation.*

MST Regions

All MSTP switches in a given region must be configured with the same VLANs. Also, each MSTP switch within the same region must have the same VLAN-to-instance assignments. (A VLAN can belong to only one instance within any region.) Within a region:

- All of the VLANs belonging to a given instance compose a single, active spanning-tree topology for that instance.
- Each instance operates independently of other regions.

Between regions there is a single, active spanning-tree topology.

How Separate Instances Affect MSTP Operation. Assigning different groups of VLANs to different instances ensures that those VLAN groups use independent forwarding paths. For example, in figure 4-3 each instance has a different forwarding path.

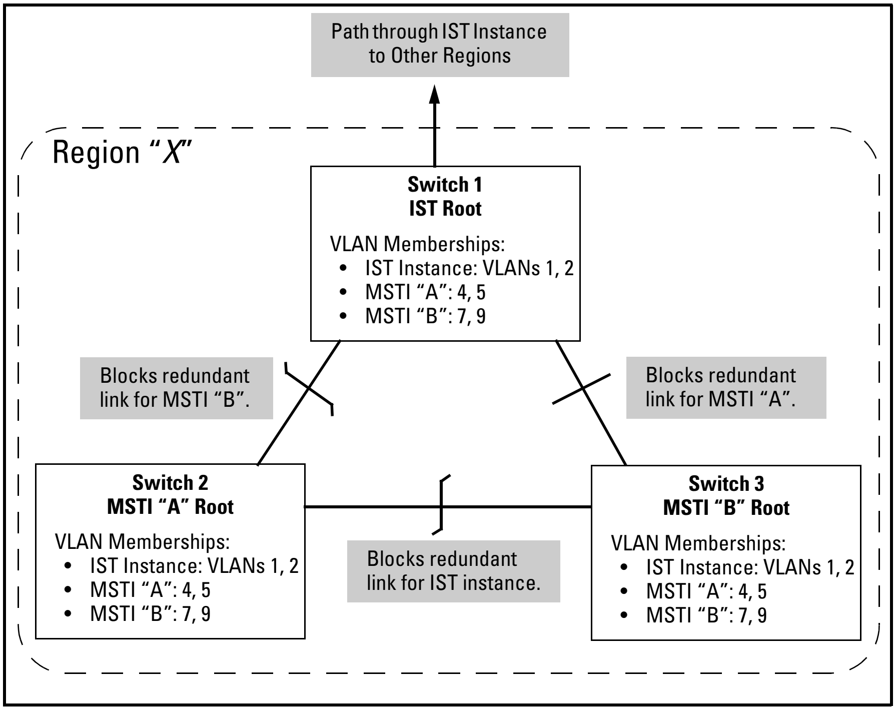

Path through IST Instance to Other Regions
Region "X"
Switch 1
IST Root
VLAN Memberships:
• IST Instance: VLANs 1, 2
• MSTI "A": 4, 5
• MSTI "B": 7, 9
Blocks redundant link for MSTI "B".
Blocks redundant link for MSTI "A".
Switch 2
MSTI "A" Root
VLAN Memberships:
• IST Instance: VLANs 1, 2
• MSTI "A": 4, 5
• MSTI "B": 7, 9
Blocks redundant link for IST instance.
Switch 3
MSTI "B" Root
VLAN Memberships:
• IST Instance: VLANs 1, 2
• MSTI "A": 4, 5
• MSTI "B": 7, 9

Figure 4-3. Active Topologies Built by Three Independent MST Instances

While allowing only one active path through a given instance, MSTP retains any redundant physical paths in the instance to serve as backups (blocked) paths in case the existing active path fails. Thus, if an active path in an instance fails, MSTP automatically activates (unblocks) an available backup to serve as the new active path through the instance for as long as the original active path is down. Note also that a given port may simultaneously operate in different states (forwarding or blocking) for different spanning-tree instances within the same region. This depends on the VLAN memberships to which the port is assigned. For example, if a port belongs to VLAN 1 in the IST instance of a region and also belongs to VLAN 4 in MSTI "*x*" in the same region, the port may apply different states to traffic for these two different instances.

Within a region, traffic routed between VLANs in separate instances can take only one physical path. To ensure that traffic in all VLANs within a region can travel between regions, all of the boundary ports for each region should belong to all VLANs configured in the region. Otherwise, traffic from some areas within a region could be blocked from moving to other regions.

All MSTP switches (as well as STP and RSTP switches) in a network use BPDUs (Bridge Protocol Data Units) to exchange information from which to build multiple, active topologies in the individual instances within a region and between regions. From this information:

- The MSTP switches in each LAN segment determine a designated bridge and designated port or trunk for the segment.
- The MSTP switches belonging to a particular instance determine the root bridge and root port or trunk for the instance.
- For the IST instance within a region, the MSTP switches linking that region to other regions (or to STP or RSTP switches) determine the IST root bridge and IST root port or trunk for the region. (For any Multiple Spanning-Tree instance—MSTI—in a region, the regional root may be a different switch that is not necessarily connected to another region.)
- The MSTP switches block redundant links within each LAN segment, across all instances, and between regions, to prevent any traffic loops.

As a result, each individual instance (spanning tree) within a region determines its regional root bridge, designated bridges, and designated ports or trunks.

Regions, Legacy STP and RSTP Switches, and the Common Spanning Tree (CST)

The IST instance and any MST instances in a region exist only within that region. Where a link crosses a boundary between regions (or between a region and a legacy STP or RSTP switch), traffic is forwarded or blocked as determined by the Common Spanning Tree (CST). The CST ensures that there is only one active path between any two regions, or between a region and a switch running STP and RSTP. (Refer to figure 4-2 on page 4-7.)

MSTP Operation with 802.1Q VLANs

As indicated in the preceding sections, within a given MST instance, a single spanning tree is configured for all VLANs included in that instance. This means that if redundant physical links exist in separate VLANs within the same instance, MSTP blocks all but one of those links. However, you can prevent the bandwidth loss caused by blocked redundant links for different VLANs in an instance by using a port trunk. The following example shows how you can use a port trunk with 802.1Q (tagged) VLANs and MSTP without unnecessarily blocking any links or losing any bandwidth.

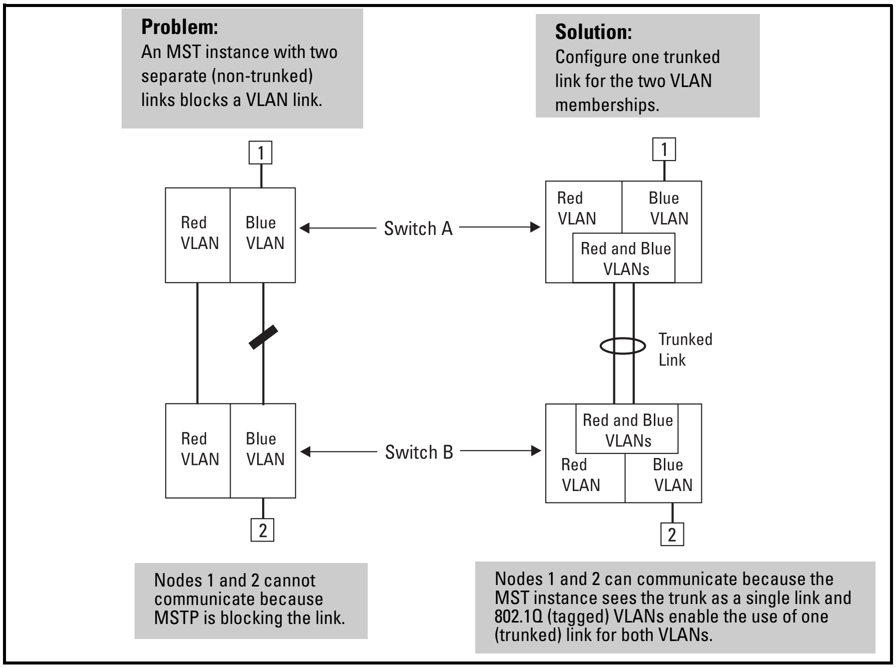

Problem:
An MST instance with two separate (non-trunked) links blocks a VLAN link.
Solution:
Configure one trunked link for the two VLAN memberships.
1
Red VLAN
Blue VLAN
Switch A
Red and Blue VLANs
Trunked Link
Switch B
2
Nodes 1 and 2 cannot communicate because MSTP is blocking the link.
Nodes 1 and 2 can communicate because the MST instance sees the trunk as a single link and 802.1Q (tagged) VLANs enable the use of one (trunked) link for both VLANs.

Figure 4-4. Example of Using a Trunked Link To Support Multiple VLAN Connectivity within the Same MST Instance

Note

All switches in a region should be configured with the VLANs used in that region, and all ports linking MSTP switches together should be members of all VLANs in the region. Otherwise, the path to the root for a given VLAN will be broken if MSTP selects a spanning tree through a link that does not include that VLAN.

Terminology

Bridge: See “MSTP Bridge”.

Common and Internal Spanning Tree (CIST): Comprises all LANs, STP, and RSTP bridges and MSTP regions in a network. The CIST automatically determines the MST regions in a network and defines the root bridge (switch) and designated port for each region. The CIST includes the Common Spanning Tree (CST), the Internal Spanning Tree (IST) within each region, and any multiple spanning-tree instances (MSTIs) in a region.

Common Spanning Tree (CST): Refers to the single forwarding path the switch calculates for STP (802.1D) and RSTP (802.1w) topologies, and for inter-regional paths in MSTP (802.1s) topologies. Note that all three types of spanning tree can interoperate in the same network. Also, the MSTP switch interprets a device running 802.1D STP or 802.1w RSTP as a separate region. (Refer to figure 4-2 on page 4-7.)

Internal Spanning Tree (IST): Comprises all VLANs within a region that are not assigned to a multiple spanning-tree instance configured within the region. All MST switches in a region should belong to the IST. In a given region "X", the IST root switch is the regional root switch and provides information on region "X" to other regions.

MSTP (Multiple Spanning Tree Protocol): A network supporting MSTP allows multiple spanning tree instances within configured regions, and a single spanning tree among regions, STP bridges, and RSTP bridges.

MSTP BPDU (MSTP Bridge Protocol Data Unit): These BPDUs carry region-specific information, such as the region identifier (region name and revision number). If a switch receives an MSTP BPDU with a region identifier that differs from its own, then the port on which that BPDU was received is on the boundary of the region in which the switch resides.

MSTP Bridge: In this manual, an MSTP bridge is a switch (or another 802.1s-compatible device) configured for MSTP operation.

MST Region: An MST region forms a multiple spanning tree domain and is a component of a single spanning-tree domain within a network. For switches internal to the MST region:

- All switches have identical MST configuration identifiers (region name and revision number).
- All switches have identical VLAN assignments to the region's IST and (optional) MST instances.
- One switch functions as the designated bridge (IST root) for the region.
- No switch has a point-to-point connection to a bridging device that cannot process RSTP BPDUs.

Operating Rules

- All switches in a region must be configured with the same set of VLANs, as well as the same MST configuration name and MST configuration number.
- Within a region, a VLAN can be allocated to either a single MSTI or to the region’s IST instance.
- All switches in a region must have the same VID-to-MST instance and VID-to-IST instance assignments.
- There is one root MST switch per configured MST instance.
- Within any region, the root switch for the IST instance is also the root switch for the region. Because boundary ports provide the VLAN connectivity between regions, all boundary ports on a region's root switch should be configured as members of all static VLANs defined in the region.
- There is one root switch for the Common and Internal Spanning Tree (CIST). Note that the per-port **hello-time** parameter assignments on the CIST root switch propagate to the ports on downstream switches in the network and override the **hello-time** configured on the downstream switch ports.
- Where multiple MST regions exist in a network, there is only one active, physical communication path between any two regions, or between an MST region and an STP or RSTP switch. MSTP blocks any other physical paths as long as the currently active path remains in service.
- Within a network, an MST region appears as a virtual RSTP bridge to other spanning tree entities (other MST regions, and any switches running 802.1D or 802.1w spanning-tree protocols).
- Within an MSTI, there is one spanning tree (one physical, communication path) between any two nodes. That is, within an MSTI, there is one instance of spanning tree, regardless of how many VLANs belong to the MSTI. Within an IST instance, there is also one spanning tree across all VLANs belonging to the IST instance.
- An MSTI comprises a unique set of VLANs and forms a single spanning-tree instance within the region to which it belongs.
- Communication between MST regions uses a single spanning tree.
- If a port on a switch configured for MSTP receives a legacy (STP/802.1D or RSTP/802.1w) BPDU, it automatically operates as a legacy port. In this case, the MSTP switch interoperates with the connected STP or RSTP switch as a separate MST region.
- Within an MST region, there is one logical forwarding topology per instance, and each instance comprises a unique set of VLANs. Where multiple paths exist between a pair of nodes using VLANs belonging to

the same instance, all but one of those paths will be blocked for that instance. However, if there are different paths in different instances, all such paths are available for traffic. Separate forwarding paths exist through separate spanning tree instances.

- A port can have different states (forwarding or blocking) for different instances (which represent different forwarding paths).
- MSTP interprets a switch mesh as a single link.
- A dynamic VLAN learned by GVRP will always be placed in the IST instance and cannot be moved to any configured MST instance.

Transitioning from STP or RSTP to MSTP

IEEE 802.1s MSTP includes RSTP functionality and is designed to be compatible with both IEEE 802.1D and 802.1w spanning-tree protocols. Even if all the other devices in your network are using STP, you can enable MSTP on the switches covered in this guide. Also, using the default configuration values, your switches will interoperate effectively with STP and RSTP devices. MSTP automatically detects when the switch ports are connected to non-MSTP devices in the spanning tree and communicates with those devices using 802.1D or 802.1w STP BPDU packets, as appropriate.

Because MSTP is so efficient at establishing the network path, ProCurve highly recommends that you update all of the switches covered in this guide to support 802.1s/MSTP. (For switches that do not support 802.1s/MSTP, ProCurve recommends that you update to RSTP to benefit from the convergence times of less than one second under optimal circumstances.) To make the best use of MSTP and achieve the fastest possible convergence times, there are some changes that you should make to the MSTP default configuration.

Note

Under some circumstances, it is possible for the rapid state transitions employed by MSTP and RSTP to result in an increase in the rates of frame duplication and misordering in the switched LAN. In order to allow MSTP and RSTP switches to support applications and protocols that may be sensitive to frame duplication and misordering, setting the Force Protocol Version parameter to **STP-compatible** allows MSTP and RSTP to operate with the rapid transitions disabled. The value of this parameter applies to all ports on the switch. See information on **force version** on page 4-21.

As indicated above, one of the benefits of MSTP and RSTP is the implementation of a larger range of port path costs, which accommodates higher network speeds. New default values have also been implemented for the path costs associated with the different network speeds. This can create some incompatibility between devices running the older 802.1D STP and your switch running MSTP or RSTP. Please see the "Note on Path Cost" on page 4-4 for more information on adjusting to this incompatibility.

Tips for Planning an MSTP Application

- Ensure that the VLAN configuration in your network supports all of the forwarding paths necessary for the desired connectivity. All ports connecting one switch to another within a region and one switch to another between regions should be configured as members of all VLANs configured in the region.
- All ports or trunks connecting one switch to another within a region should be configured as members of all VLANs in the region. Otherwise, some VLANs could be blocked from access to the spanning-tree root for an instance or for the region.
- Plan individual regions based on VLAN groupings. That is, plan on all MSTP switches in a given region supporting the same set of VLANs. Within each region, determine the VLAN membership for each spanning-tree instance. (Each instance represents a single forwarding path for all VLANs in that instance.)
- There is one logical spanning-tree path through the following:
  - Any inter-regional links
  - Any IST or MST instance within a region
  - Any legacy (802.1D or 802.1w) switch or group of switches. (Where multiple paths exist between an MST region and a legacy switch, expect the CST to block all but one such path.)
- Determine the root bridge and root port for each instance.
- Determine the designated bridge and designated port for each LAN segment.
- Determine which VLANs to assign to each instance, and use port trunks with 802.1Q VLAN tagging where separate links for separate VLANs would result in a blocked link preventing communication between nodes on the same VLAN. (Refer to "MSTP Operation with 802.1Q VLANs" on page 4-11.)
- Identify the edge ports connected to end nodes and enable the edge-port setting for these ports. Leave the edge-port setting disabled for ports connected to another switch, a bridge, or a hub.

Note on MSTP Rapid State Transitions

Under some circumstances the rapid state transitions employed by MSTP can increase the rates of frame duplication and misordering in the switched LAN. To allow MSTP switches to support applications and protocols that may be sensitive to frame duplication and misordering, setting the Force Protocol Version (**force-version**) parameter to **stp-compatible** allows MSTP to operate with rapid transitions disabled. The value of this parameter applies to all ports on the switch. See the information on **force-version** on page 4-21.

Steps for Configuring MSTP

This section outlines the general steps for configuring MSTP operation in your network, and assumes you have already planned and configured the VLANs you want MSTP to use. The actual MSTP parameter descriptions are in the following sections.

Note

The switch supports MSTP configuration through the CLI.

1. Configure MSTP global parameters. This step involves configuring the following:
   - Required parameters for MST region identity:

     Region Name: **spanning-tree config-name**

     Region Revision Number: **spanning-tree config-revision**
   - Optional MSTP parameter changes for region settings:

     ProCurve recommends that you leave these parameters at their default settings for most networks. Refer to the "Caution" on page 4-9.
     - The maximum number of hops before the MSTP BPDU is discarded (default: 20)
       spanning-tree max-hops

     - Force-Version operation
       spanning-tree force-version

     - Forward Delay
       spanning-tree forward-delay

     - Hello Time (used if the switch operates as the root device.)
       spanning-tree hello-time

  - Maximum age to allow for STP packets before discarding
    spanning-tree maximum-age

  - Device spanning-tree priority. Specifies the priority value used along with the switch MAC address to determine which device is root. The lower a priority value, the higher the priority.
    spanning-tree priority

2. Configure MST instances.

   - Configure one instance for each VLAN group that you want to operate as an active topology within the region to which the switch belongs. When you create the instance, you must include a minimum of one VID. You can add more VIDs later if desired.
     spanning-tree instance

     To move a VLAN from one instance to another, first use **no spanning-tree instance <** ***n*** **> vlan <** ***vid*** **>** to unmap the VLAN from the current instance, then add the VLAN to the other instance. (While the VLAN is unmapped from an MSTI, it is associated with the region's IST instance.)

   - Configure the priority for each instance.
     spanning-tree instance

3. Configure MST instance port parameters. Enable **edge-port** for ports connected to end nodes (page 4-22), but leave it disabled (the default) for connections to another switch, a bridge, or a hub. Set the path cost value for the port(s) used by a specific MST instance. Leaving this setting at the default auto allows the switch to calculate the path-cost from the link speed.
   spanning-tree instance

4. Enable spanning-tree operation on the switch.
   spanning-tree

Configuring MSTP Operation Mode and Global Parameters

Command	**Page**
spanning-tree config-name < *ascii-string* >	4-19
spanning-tree config-revision < *revision-number* >	4-20
spanning-tree max-hops < *hop-count* >	4-20
spanning-tree force-version < stp-compatible | rstp-operation | mstp-operation>	4-21
spanning-tree hello-time < 1..10 >	4-21

The commands in this section apply on the switch level, and do not affect individual port configurations.

Syntax: [no] spanning-tree config-name < *ascii-string* >

This command resets the configuration name of the MST region in which the switch resides. This name can include up to 32 nonblank characters and is case-sensitive. On all switches within a given MST region, the configuration names must be identical. Thus, if you want more than one MSTP switch in the same MST region, you must configure the identical region name on all such switches. If you retain the default configuration name on a switch, it cannot exist in the same MST region with another switch. (Default Name: A text string using the hexadecimal representation of the switch's MAC address)

The **no** *form of the command overwrites the currently configured name with the default name.*

Note: *This option is available only when the switch is configured for MSTP operation. Also, there is no defined limit on the number of regions you can configure.*

Syntax: spanning-tree config-revision < *revision-number* >

This command configures the revision number you designate for the MST region in which you want the switch to reside. This setting must be the same for all switches residing in the same region. Use this setting to differentiate between region configurations in situations such as the following:

- *Changing configuration settings within a region where you want to track the configuration versions you use*
- *Creating a new region from a subset of switches in a current region and want to maintain the same region name.*
- *Using the* **pending** *option to maintain two different configuration options for the same physical region.*

Note that this setting must be the same for all MSTP switches in the same MST region. (Range: **0 - 65535***; Default:* **0***)*

> ***Note:*** *This option is available only when the switch is configured for MSTP operation.*

Syntax: spanning-tree max-hops < *hop-count* >

This command resets the number of hops allowed for BPDUs in an MST region. When an MSTP switch receives a BPDU, it decrements the hop-count setting the BPDU carries. If the hop-count reaches zero, the receiving switch drops the BPDU. Note that the switch does not change the message-age and maximum-age data carried in the BPDU as it moves through the MST region and is propagated to other regions. (Range: 1 - 40; Default: 20)

Syntax: spanning-tree force-version < stp-compatible | rstp-operation | mstp-operation >

Sets the spanning-tree compatibility mode. When the switch is configured with MSTP mode, this command forces the switch to emulate behavior of earlier versions of spanning tree protocol or return to MSTP behavior. The command is useful in test or debug applications, and removes the need to reconfigure the switch for temporary changes in spanning-tree operation.

stp-compatible: *The switch applies 802.1D STP operation on all ports.*

rstp-operation: *The switch applies 802.1w operation on all ports except those ports where it detects a system using 802.1D Spanning Tree.*

mstp-operation: *The switch applies 802.1s MSTP operation on all ports where compatibility with 802.1D or 802.1w spanning tree protocols is not required.*

This command is available when the protocol version is set to **mstp** *(see 'protocol-version' above).*

Note that even when mstp-operation is selected, if the switch detects an 802.1D BPDU or an 802.1w BPDU on a port, it communicates with the device linked to that port using STP or RSTP BPDU packets. Also, if errors are encountered as described in the "Note on MSTP Rapid State Transitions" on page 4-17, setting **force-version** *to* **stp-compatible** *forces the MSTP switch to communicate out all ports using operations that are compatible with IEEE 802.1D STP.*

Syntax: spanning-tree hello-time < 1..10 >

If MSTP is running and the switch is operating as the CIST root for your network, this command specifies the time in seconds between transmissions of BPDUs for all ports on the switch configured with the **Global** option. *(the default). This parameter applies in MSTP, RSTP and STP modes. During MSTP operation, you can override this global setting on a per-port basis with this command:* **spanning-tree <** ***port-list*** **> hello-time < 1..10 >** (page 4-22). (*Default:* **2**.)

Configuring Basic Port Connectivity Parameters

Command	Page
spanning-tree < *port-list* >	
edge-port	below
mcheck	below
hello-time < global | 1..10 >	4-23
spanning-tree path-cost < auto | 200000000 >	4-26
spanning-tree point-to-point-mac < force-true | force-false | auto>	4-27
spanning-tree priority <priority-multiplier>	4-27

The basic port connectivity parameters affect spanning-tree links at the global level. In most cases, ProCurve recommends that you use the default settings for these parameters and apply changes on a per-port basis only where a non-default setting is clearly indicated by the circumstances of individual links.

Syntax: [no] spanning-tree < *port-list* > < edge-port | mcheck >

[edge-port]

Enable **edge-port** *on ports connected to end nodes. During spanning tree establishment, ports with* **edge-port** *enabled transition immediately to the forwarding state. Disable this feature on any switch port that is connected to another switch, bridge, or hub. (Default:* **No** *- disabled)*

The **no spanning-tree <** ***port-list*** **> edge-port** *command disables edge-port operation on the specified ports.*

[mcheck]

Forces a port to send RSTP BPDUs for 3 seconds. This allows for another switch connected to the port and running RSTP to establish its connection quickly and for identifying switches running 802.1D STP. If the whole-switch force-version parameter is set to stp-compatible, the switch ignores the mcheck setting and sends 802.1D STP BPDUs out all ports. Disable this feature on all ports that are known to be connected to devices that are running 802.1D STP. (Default: **Yes** *- enabled)*

The **no spanning-tree <** ***port-list*** **> mcheck** *command disables mcheck.*

Syntax: spanning-tree < *port-list* > < hello-time | path-cost | point-to-point-mac | priority >

[hello-time < global | 1 - 10 >]

When the switch is the CIST root, this parameter specifies the interval (in seconds) between periodic BPDU transmissions by the designated ports. This interval also applies to all ports in all switches downstream from each port in the < *port-list*>. *A setting of* **global** *indicates that the ports in* < *port-list*> *on the CIST root are using the value set by the global spanning-tree* **hello-time** *value (page 4-21). When a given switch "X" is not the CIST root, the per-port* **hello-time** *for all active ports on switch "X" is propagated from the CIST root, and is the same as the* **hello-time** *in use on the CIST root port in the currently active path from switch "X" to the CIST root. (That is, when switch "X" is not the CIST root, then the upstream CIST root's port* **hello-time** *setting overrides the* **hello-time** *setting configured on switch "X". (Default Per-Port setting:* **Use Global**. *Default Global Hello-Time:* **2**.*)*

[path-cost < auto | 1..200000000 >]

Assigns an individual port cost that the switch uses to determine which ports are forwarding ports in a given spanning tree. In the default configuration (auto) the switch determines a port's path cost by the port's type:

- *10 Mbps:* **2000000**
- *100 Mbps:* **200000**
- *1 Gbps:* **20000**

Refer to "Note on Path Cost" on page 4-4 for information on compatibility with devices running 802.1D STP for the path cost values (Default: Auto.).

[point-to-point-mac < force-true | force-false | auto >]

This parameter informs the switch of the type of device to which a specific port connects.

Force-True ***(default)****: Indicates a point-to-point link to a device such as a switch, bridge, or end-node.*

Force-False*: Indicates a connection to a hub (which is a shared LAN segment).*

Auto: *Causes the switch to set Force-False on the port if it is not running at full duplex. (Connections to hubs are half-duplex.)*

[priority < priority-multiplier>]

MSTP uses this parameter to determine the port(s) to use for forwarding. The port with the lowest priority number has the highest priority. The range is 0 to 240, and is configured by specifying a multiplier in the range of 0 - 15. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

(priority-multiplier) x 16

*For example, if you configure "***2***" as the priority multiplier on a given port, then the actual* **Priority** *setting is 32. Thus, after you specify the port priority multiplier, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree** *or* **show spanning-tree <** ***port-list*** **>** *displays.*

You can view the actual multiplier setting for ports by executing **show running** *and looking for an entry in this format:*

spanning-tree < *port-list* > priority < *priority-multiplier* >

For example, configuring port A2 with a priority multiplier of "3" results in this line in the **show running** *output:*

```
spanning-tree A2 priority 3
```

Configuring MST Instance Parameters

Command	Page
[no] spanning-tree instance < 1..16 > vlan < *vid*> [*vid..vid*] no spanning-tree instance < 1..16 >	4-22
spanning-tree instance < 1..16 > priority < 0..15 >	4-25
spanning-tree priority < 0..15 >	4-26

Syntax: [no] spanning-tree instance < 1..16 > vlan < *vid* [*vid..vid*] >
no spanning-tree instance < 1..16 >

Configuring MSTP on the switch automatically configures the IST instance and places all statically configured VLANs on the switch into the IST instance. This command creates a new MST instance (MSTI) and moves the VLANs you specify from the IST to the MSTI. At least one VLAN must be mapped to a MSTI when you create it. (A VLAN cannot be mapped to more than one instance at a time.) You can create up to 16 MSTIs in a region. The **no** *form of the command deletes the specified VLAN or if no VLANs are specified, the* **no** *form of the command deletes the specified MSTI. (Removing a VLAN from an MSTI returns the VLAN to the IST instance, where it can either remain or be re-assigned to another MSTI configured in the region.)*

The **no** *form of the command deletes the specified VLAN, or if no VLANs are specified, the* **no** *form of the command deletes the specified MSTI.*

Syntax: spanning-tree instance < 1..16 > priority < priority-multiplier >

This command sets the switch (bridge) priority for the designated instance. This priority is compared with the priorities of other switches in the same instance to determine the root switch for the instance. The lower the priority value, the higher the priority. (If there is only one switch in the instance, then that switch is the root switch for the instance.) The root bridge in a given instance provides the path to connected instances in other regions that share one or more of the same VLAN(s). (Traffic in VLANs assigned to a numbered STP instance in a given region moves to other regions through the root switch for that instance.)

The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch for the specified MST instance is:

(*priority-multiplier*) x 4096

*For example, if you configure "***5***" as the priority-multiplier for MST Instance 1 on a given MSTP switch, then the* **Switch Priority** *setting is 20,480 for that instance in that switch.*

Note: *If multiple switches in the same MST instance have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that instance.*

Syntax: spanning-tree priority < priority-multiplier >

Every switch running an instance of MSTP has a Bridge Identifier, which is a unique identifier that helps distinguish this switch from all others. The switch with the lowest Bridge Identifier is elected as the root for the tree.

The Bridge Identifier is composed of a configurable Priority component (2 bytes) and the bridge's MAC address (6 bytes). The ability to change the Priority component provides flexibility in determining which switch will be the root for the tree, regardless of its MAC address.

This command sets the switch (bridge) priority for the designated region in which the switch resides. The switch compares this priority with the priorities of other switches in the same region to determine the root switch for the region. The lower the priority value, the higher the priority. (If there is only one switch in the region, then that switch is the root switch for the region.) The root bridge in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance. (Traffic in VLANs assigned to a numbered STP instance in a given region moves to other regions through the root switch for that instance.)

The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch is:

(*priority-multiplier*) x 4096

*For example, if you configure "***2***" as the priority-multiplier on a given MSTP switch, then the* **Switch Priority** *setting is 8,192.*

Note: *If multiple switches in the same MST region have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that region.*

Configuring MST Instance Per-Port Parameters

Command	**Page**
spanning-tree instance < 1..16 > < port-list > path-cost < auto | 1..200000000 >	4-28
spanning-tree instance < 1..16 > < *port-list* > priority < *priority-multiplier* >	4-29
spanning-tree < *port-list* > priority < *priority-multiplier* >	4-30

Syntax: spanning-tree instance < 1..16 > < *port-list* > path-cost < auto | 1..200000000 >

This command assigns an individual port cost for the specified MST instance. (For a given port, the path cost setting can be different for different MST instances to which the port may belong.) The switch uses the path cost to determine which ports are the forwarding ports in the instance; that is which links to use for the active topology of the instance and which ports to block. The settings are either **auto** *or in a range from 1 to 200,000,000. With the* **auto** *setting, the switch calculates the path cost from the link speed:*

10 Mbps — 2000000
100 Mbps — 200000
1 Gbps — 20000

(Default: **Auto***)*

Syntax: spanning-tree instance < 1..16 >< *port-list* > priority <*priority-multiplier*>

This command sets the priority for the specified port(s) in the specified MST instance. (For a given port, the priority setting can be different for different MST instances to which the port may belong.) The priority range for a port in a given MST instance is 0-255. However, this command specifies the priority as a multiplier (0 - 15) of 16. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

(*priority-multiplier*) *x* 16

*For example, if you configure "***2***" as the priority multiplier on a given port in an MST instance, then the actual* **Priority** *setting is 32. Thus, after you specify the port priority multiplier in an instance, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree instance < 1..16 >** *or* **show spanning-tree <** ***port-list*** **> instance < 1..16 >** *displays.*

You can view the actual multiplier setting for ports in the specified instance by executing **show running** *and looking for an entry in this format:*

spanning-tree instance < 1..15 > < *port-list* > priority < *priority-multiplier* >

For example, configuring port A2 with a priority multiplier of "3" in instance 1, results in this line in the **show running** *output:*

```
spanning-tree instance 1 A2 priority 3
```

Syntax: spanning-tree < *port-list* > priority < *priority-multiplier* >

This command sets the priority for the specified port(s) for the IST (that is, Instance 0) of the region in which the switch resides. The "priority" component of the port's "Port Identifier" is set. The Port Identifier is a unique identifier that helps distinguish this switch's ports from all others. It consists of the Priority value with the port number extension—PRIORITY:PORT_NUMBER. A port with a lower value of Port Identifier is more likely to be included in the active topology.

This priority is compared with the priorities of other ports in the IST to determine which port is the root port for the IST instance. The lower the priority value, the higher the priority. The IST root port (or trunk) in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance.

The priority range for a port in a given MST instance is 0-240. However, this command specifies the priority as a multiplier (0 - 15) of 16. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

(*priority-multiplier*) x 16

*For example, configuring "***5***" as the priority multiplier on a given port in the IST instance for a region creates an actual* **Priority** *setting of* **80**. *Thus, after you specify the port priority multiplier for the IST instance, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree instance ist** *or* **show spanning-tree <** ***port-list*** **> instance ist** *displays. You can view the actual multiplier setting for ports in the IST instance by executing* **show running** *and looking for an entry in this format:*

spanning-tree < *port-list* > priority < *priority-multiplier* >

For example, configuring port A2 with a priority multiplier of "2" in the IST instance, results in this line in the **show running** *output:*

```
spanning-tree A2 priority 2
```

Enabling or Disabling Spanning Tree Operation

This command enables or disables spanning tree operation for any spanning tree protocol enabled on the switch. Before using this command to enable spanning tree, ensure that the version you want to use is active on the switch.

Syntax: [no] spanning-tree

Enabling spanning tree with MSTP configured implements MSTP for all physical ports on the switch, according to the VLAN groupings for the IST instance and any other configured instances. Disabling MSTP removes protection against redundant loops that can significantly slow or halt a network.

This command simply turns spanning tree on or off. It does not change the existing spanning tree configuration.

Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another

Command	Page
spanning-tree pending < apply | config-name | config-revision | instance | reset >	4-32

This operation exchanges the currently active MSTP configuration with the currently pending MSTP configuration. It enables you to implement a new MSTP configuration with minimal network disruption or to exchange MSTP configurations for testing or troubleshooting purposes.

When you configure or reconfigure MSTP, the switch re-calculates the corresponding network paths. This can have a ripple effect throughout your network as adjacent MSTP switches recalculate network paths to support the configuration changes invoked in a single switch. Although MSTP employs RSTP operation, the convergence time for implementing MSTP changes can be disruptive to your network. However, by using the spanning-tree **pending** feature, you can set up an MSTP on the switch and then invoke all instances of the new configuration at the same time, instead of one at a time.

To Create a Pending MSTP Configuration. This procedure creates a pending MSTP configuration and exchanges it with the active MSTP configuration.

1. Configure the VLANs you want included in any instances in the new region. When you create the pending region, all VLANs configured on the switch will be assigned to the pending IST instance unless assigned to other, pending MST instances.
2. Configure MSTP as the spanning-tree protocol, then execute **write mem** and reboot. (The pending option is available only with MSTP enabled.)
3. Configure the pending region name to assign to the switch.
4. Configure the pending **config-revision** number for the region name.
5. If you want an MST instance other than the IST instance, configure the instance number and assign the appropriate VLANs (VIDs). (The **pending** command creates the region's IST instance automatically.)
6. Repeat step 5 for each additional MST instance you want to configure.
7. Use the **show spanning-tree pending** command to review your pending configuration (page 4-39).
8. Use the **spanning-tree pending apply** command to exchange the currently active MSTP configuration with the pending MSTP configuration.

Syntax: spanning-tree pending < apply | *config-name* | *config-revision* | instance | reset >

apply

Exchanges the currently active MSTP configuration with the pending MSTP configuration.

config-name

Specifies the pending MST region name. Must be the same for all MSTP switches in the region. (Default: The switch's MAC address.)

config-revision

Specifies the pending MST region configuration revision number. Must be the same for all MSTP switches in the region. (Default: **0***).*

instance < 1..16 > vlan [< *vid* | *vid-range* >

Creates the pending instance and assigns one or more VLANs to the instance.

reset

Copies the switch's currently active MSTP configuration to the pending configuration. This is useful when you want to experiment with the current MSTP configuration while maintaining an unchanged version.

9. To view the current pending MSTP configuration, use the **show spanning-tree pending** command (page 4-39).

Displaying MSTP Statistics and Configuration

Command	Page
MSTP Statistics:	
show spanning-tree [< *port-list* >]	below
show spanning-tree instance < ist | 1..16 >	4-35
MSTP Configuration	
show spanning-tree [*port-list*] config	4-36
show spanning-tree [port-list] config instance < ist | 1..16 >	4-37
show spanning-tree mst-config	4-38
show spanning-tree pending< < instance | ist > | mst-config >	4-39

Displaying MSTP Statistics

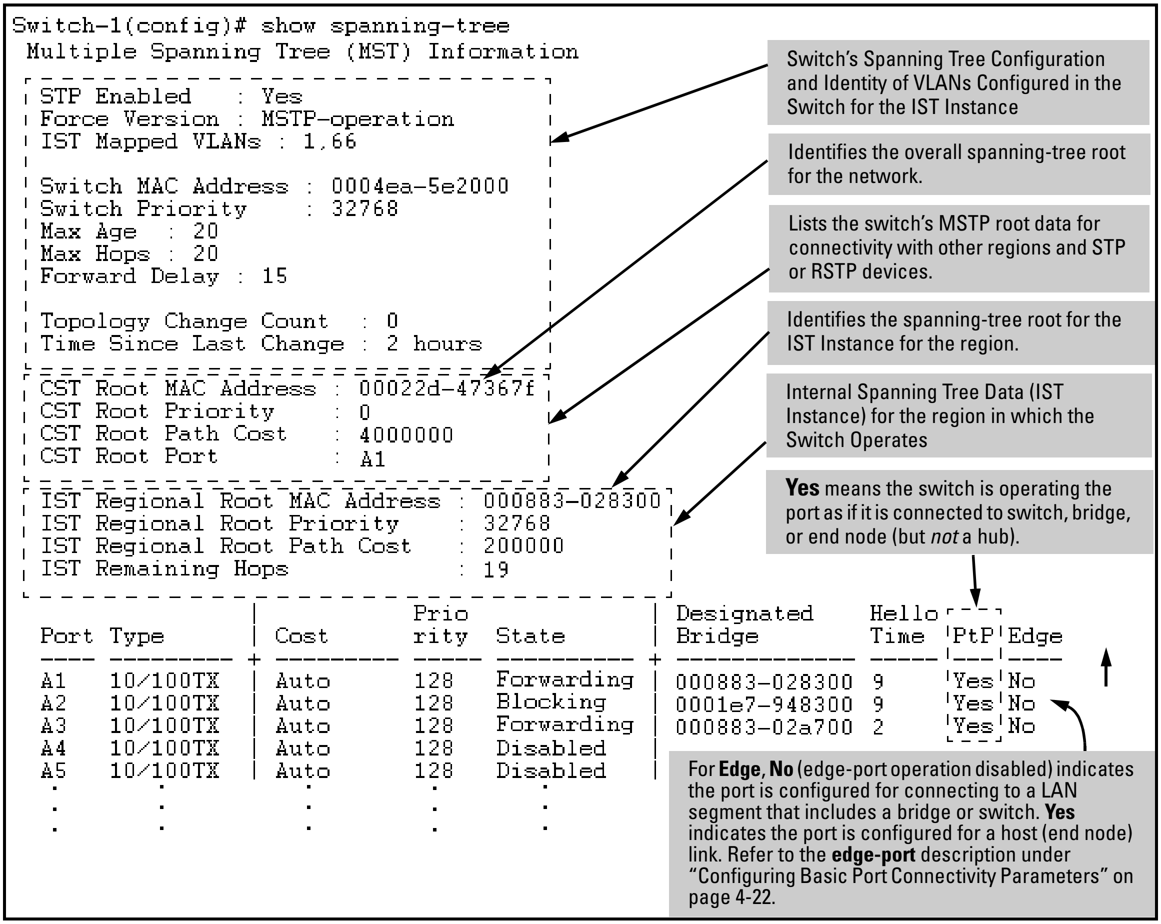
Displaying Switch Statistics for the Common Spanning Tree. This command displays the MSTP statistics for the connections between MST regions in a network.

Syntax: show spanning-tree

This command displays the switch's global and regional spanning-tree status, plus the per-port spanning-tree operation at the regional level. Note that values for the following parameters appear only for ports connected to active devices: **Designated Bridge**, **Hello Time**, **PtP**, *and* **Edge**.

Syntax: show spanning-tree < *port-list* >

This command displays the spanning-tree status for the designated port(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, you would use this command: **show spanning-tree a20-a42,trk1**



```
Switch-1(config)# show spanning-tree
 Multiple Spanning Tree (MST) Information

  STP Enabled   : Yes
  Force Version : MSTP-operation
  IST Mapped VLANs : 1,66

  Switch MAC Address : 0004ea-5e2000
  Switch Priority    : 32768
  Max Age  : 20
  Max Hops : 20
  Forward Delay : 15

  Topology Change Count  : 0
  Time Since Last Change : 2 hours

  CST Root MAC Address : 00022d-47367f
  CST Root Priority    : 0
  CST Root Path Cost   : 4000000
  CST Root Port        : A1

  IST Regional Root MAC Address : 000883-028300
  IST Regional Root Priority    : 32768
  IST Regional Root Path Cost   : 200000
  IST Remaining Hops            : 19

                 |            Prio               | Designated     Hello
  Port  Type     | Cost       rity  State        | Bridge         Time  PtP  Edge
  ----  -------- + ---------- ----- ------------ + -------------  ----- ---  ----
  A1    10/100TX | Auto       128   Forwarding   | 000883-028300  9     Yes  No
  A2    10/100TX | Auto       128   Blocking     | 0001e7-948300  9     Yes  No
  A3    10/100TX | Auto       128   Forwarding   | 000883-02a700  2     Yes  No
  A4    10/100TX | Auto       128   Disabled     |
  A5    10/100TX | Auto       128   Disabled     |
   .       .        .          .        .
   .       .        .          .        .
   .       .        .          .        .
```

Switch's Spanning Tree Configuration and Identity of VLANs Configured in the Switch for the IST Instance

Identifies the overall spanning-tree root for the network.

Lists the switch's MSTP root data for connectivity with other regions and STP or RSTP devices.

Identifies the spanning-tree root for the IST Instance for the region.

Internal Spanning Tree Data (IST Instance) for the region in which the Switch Operates

Yes means the switch is operating the port as if it is connected to switch, bridge, or end node (but *not* a hub).

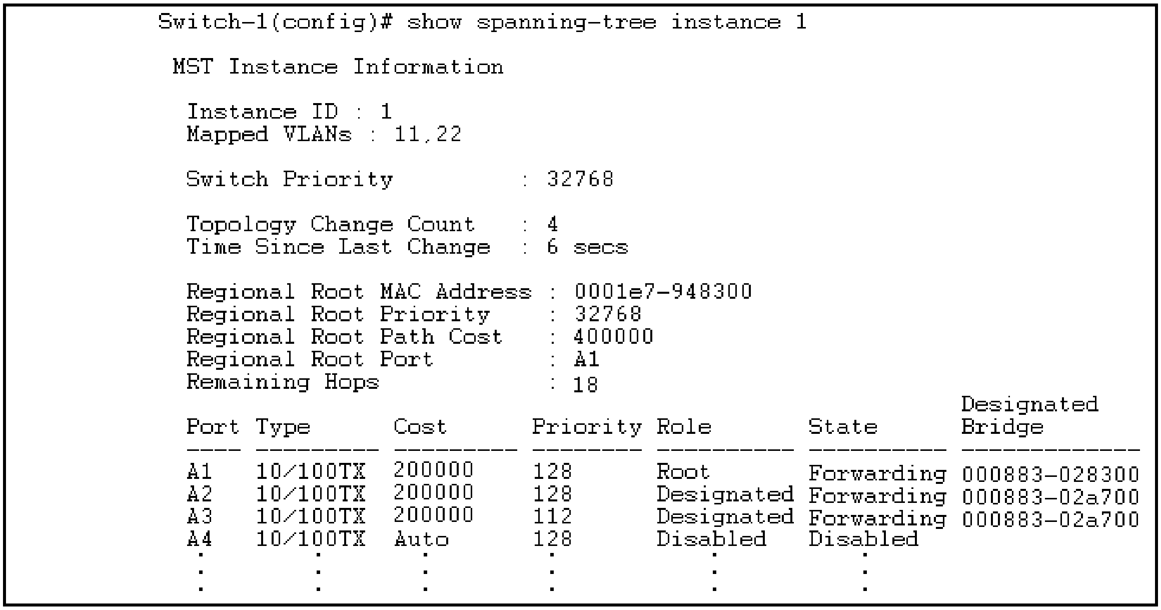
For **Edge**, **No** (edge-port operation disabled) indicates the port is configured for connecting to a LAN segment that includes a bridge or switch. **Yes** indicates the port is configured for a host (end node) link. Refer to the **edge-port** description under "Configuring Basic Port Connectivity Parameters" on page 4-22.

Figure 4-5. Example of Common Spanning Tree Status on an MSTP Switch

Displaying Switch Statistics for a Specific MST Instance.

Syntax: show spanning-tree instance < ist | 1..16 >

This command displays the MSTP statistics for either the IST instance or a numbered MST instance running on the switch.



```
Switch-1(config)# show spanning-tree instance 1

 MST Instance Information

  Instance ID : 1
  Mapped VLANs : 11,22

  Switch Priority          : 32768

  Topology Change Count    : 4
  Time Since Last Change   : 6 secs

  Regional Root MAC Address : 0001e7-948300
  Regional Root Priority    : 32768
  Regional Root Path Cost   : 400000
  Regional Root Port        : A1
  Remaining Hops            : 18
                                                                Designated
  Port Type      Cost      Priority Role       State      Bridge
  ---- --------- --------- -------- ---------- ---------- -------------
  A1   10/100TX  200000    128      Root       Forwarding 000883-028300
  A2   10/100TX  200000    128      Designated Forwarding 000883-02a700
  A3   10/100TX  200000    112      Designated Forwarding 000883-02a700
  A4   10/100TX  Auto      128      Disabled   Disabled
   .       .       .         .          .          .
   .       .       .         .          .          .
   .       .       .         .          .          .
```

Figure 4-6. Example of MSTP Statistics for a Specific Instance on an MSTP Switch

Displaying the MSTP Configuration

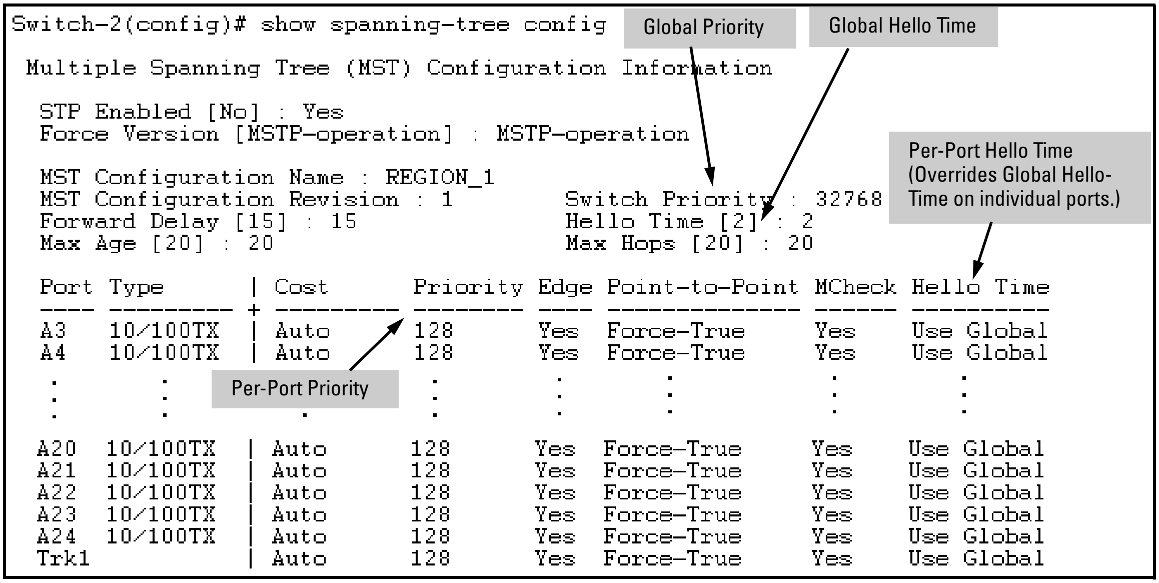
Displaying the Global MSTP Configuration. This command displays the switch's basic and MST region spanning-tree configuration, including basic port connectivity settings.

Syntax: show spanning-tree config

The upper part of this output shows the switch's global spanning-tree configuration that applies to the MST region. The port listing shows the spanning-tree port parameter settings for the spanning-tree region operation (configured by the **spanning-tree <** ***port-list*** **>** *command). For information on these parameters, refer to "Configuring Basic Port Connectivity Parameters" on page 4-22.*

Syntax: show spanning-tree < *port-list* > config

This command shows the same data as the above command, but lists the spanning-tree port parameter settings for only the specified port(s) and/or trunk(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, use this command: **show spanning-tree a20-a24,trk1 config**



```
Switch-2(config)# show spanning-tree config

 Multiple Spanning Tree (MST) Configuration Information

  STP Enabled [No] : Yes
  Force Version [MSTP-operation] : MSTP-operation

  MST Configuration Name : REGION_1
  MST Configuration Revision : 1        Switch Priority : 32768
  Forward Delay [15] : 15               Hello Time [2] : 2
  Max Age [20] : 20                     Max Hops [20] : 20

  Port Type      | Cost      Priority Edge Point-to-Point MCheck Hello Time
  ---- --------- + --------- -------- ---- -------------- ------ ----------
  A3   10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A4   10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
   .      .            .      .        .       .           .        .
   .      .            .      .        .       .           .        .
   .      .            .      .        .       .           .        .

  A20  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A21  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A22  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A23  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A24  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  Trk1           | Auto      128      Yes  Force-True     Yes    Use Global
```

Global Priority

Global Hello Time

Per-Port Hello Time (Overrides Global Hello-Time on individual ports.)

Per-Port Priority

Figure 4-7. Example of Displaying the Switch's Global Spanning-Tree Configuration

Displaying Per-Instance MSTP Configurations. These commands displays the per-instance port configuration and current state, along with instance identifiers and regional root data.

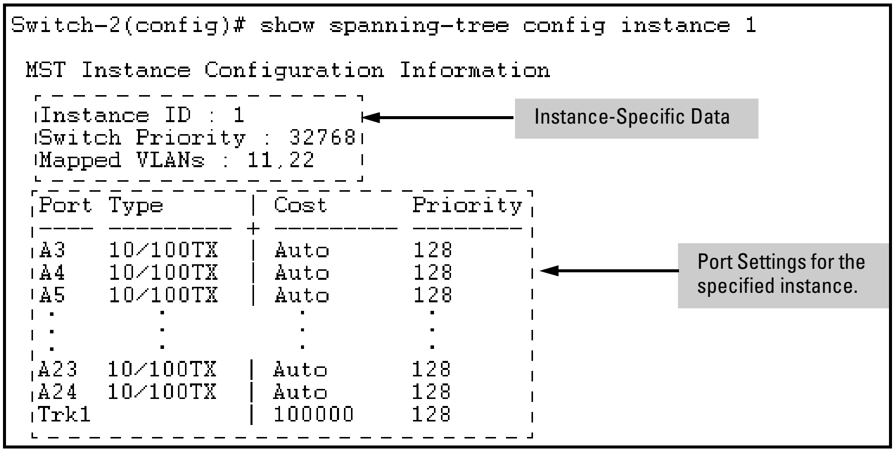
Syntax: show spanning-tree config instance < ist | 1..16 >

The upper part of this output shows the instance data for the specified instance. The lower part of the output lists the spanning-tree port settings for the specified instance.

Syntax: show spanning-tree < *port-list* > config instance < ist | 1..16 >

This command shows the same data as the above command, but lists the spanning-tree port parameter settings for only the specified port(s) and/or trunk(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, use this command:

show spanning-tree a20-a24,trk1 config instance 1



```
Switch-2(config)# show spanning-tree config instance 1

 MST Instance Configuration Information

  Instance ID : 1
  Switch Priority : 32768
  Mapped VLANs : 11,22

  Port Type      | Cost      Priority
  ---- --------- + --------- --------
  A3   10/100TX  | Auto      128
  A4   10/100TX  | Auto      128
  A5   10/100TX  | Auto      128
  .       .          .         .
  .       .          .         .
  .       .          .         .
  A23  10/100TX  | Auto      128
  A24  10/100TX  | Auto      128
  Trk1           | 100000    128
```

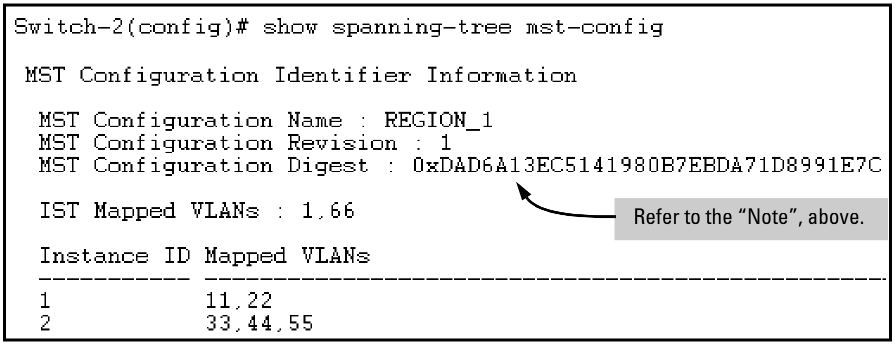
Figure 4-8. Example of the Configuration Listing for a Specific Instance

Displaying the Region-Level Configuration in Brief. This command output is useful for quickly verifying the allocation of VLANs in the switch's MSTP configuration and for viewing the configured region identifiers.

Syntax: show spanning-tree mst-config

This command displays the switch's regional configuration.

***Note:** The switch computes the* **MSTP Configuration Digest** *from the VID to MSTI configuration mappings on the switch itself. As required by the 802.1s standard, all MSTP switches within the same region must have the same VID to MSTI assignments, and any given VID can be assigned to either the IST or one of the MSTIs within the region. Thus, the MSTP Configuration Digest must be identical for all MSTP switches intended to belong to the same region. When comparing two MSTP switches, if their Digest identifiers do not match, then they cannot be members of the same region.*



```
Switch-2(config)# show spanning-tree mst-config

 MST Configuration Identifier Information

  MST Configuration Name : REGION_1
  MST Configuration Revision : 1
  MST Configuration Digest : 0xDAD6A13EC5141980B7EBDA71D8991E7C

  IST Mapped VLANs : 1,66

  Instance ID Mapped VLANs
  ----------- ----------------------------------------------------
  1           11,22
  2           33,44,55
```

Refer to the "Note", above.

Figure 4-9. Example of a Region-Level Configuration Display

Displaying the Pending MSTP Configuration. This command displays the MSTP configuration the switch will implement if you execute the spanning-tree pending apply command (Refer to "Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another" on page 4-31.)

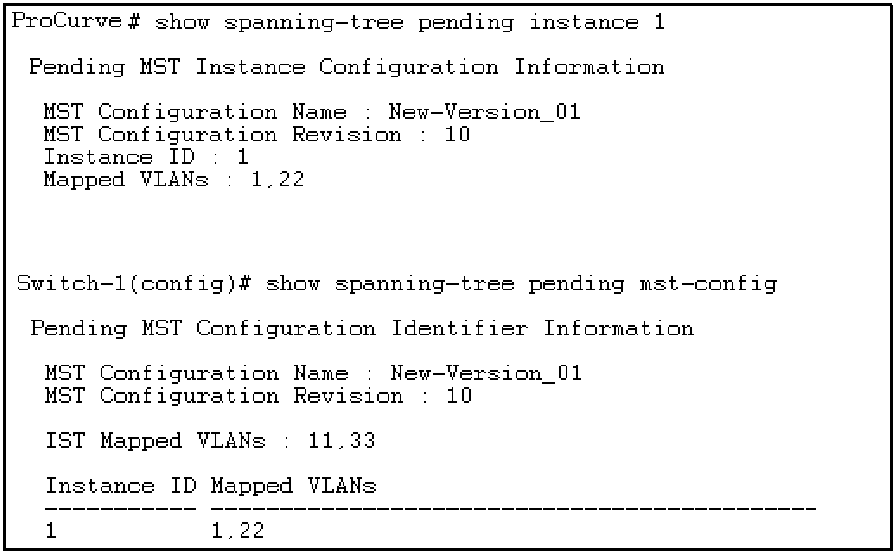
Syntax: show spanning-tree pending < instance | mst-config >

instance < 1..16 | ist >

Lists region, instance I.D. and VLAN information for the specified, pending instance.

mst-config

Lists region, IST instance VLAN(s), numbered instances, and assigned VLAN information for the pending MSTP configuration.



```
ProCurve # show spanning-tree pending instance 1

 Pending MST Instance Configuration Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 10
  Instance ID : 1
  Mapped VLANs : 1,22



Switch-1(config)# show spanning-tree pending mst-config

 Pending MST Configuration Identifier Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 10

  IST Mapped VLANs : 11,33

  Instance ID Mapped VLANs
  ----------- --------------------------------------------
  1           1,22
```

Figure 4-10. Example of Displaying a Pending Configuration

Operating Notes

SNMP MIB Support for MSTP. MSTP is a superset of the STP/802.1D and RSTP/802.1w protocols and uses the MIB objects defined for these two protocols.

Troubleshooting

Duplicate packets on a VLAN, or packets not arriving on a LAN at all. The allocation of VLANs to MSTIs may not be identical among all switches in a region.

A Switch Intended To Operate Within a Region Does Not Receive Traffic from Other Switches in the Region. An MSTP switch intended for a particular region may not have the same configuration name or region revision number as the other switches intended for the same region. The MSTP Configuration Name and MSTP Configuration Revision number must be identical on all MSTP switches intended for the same region. Another possibility is that the set of VLANs configured on the switch may not match the set of VLANs configured on other switches in the intended region.

5

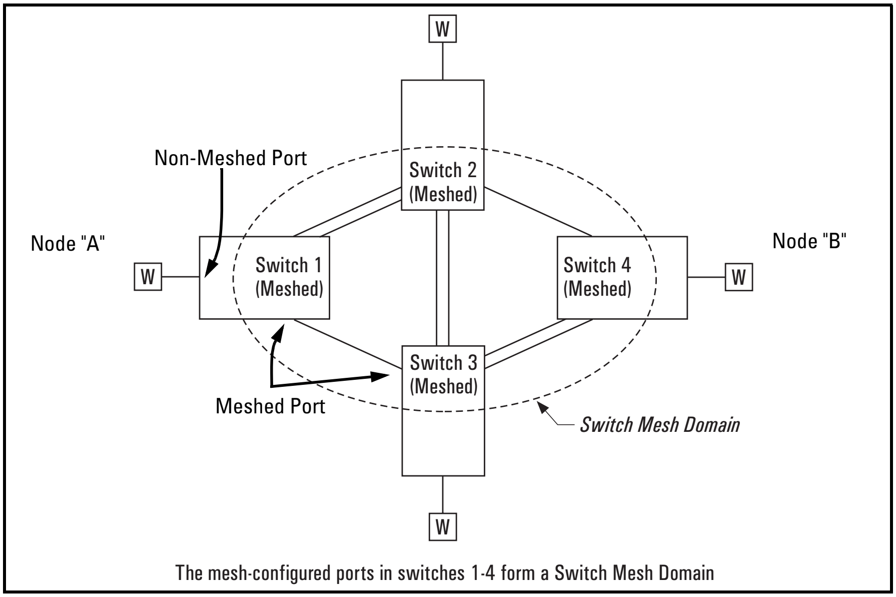
Switch Meshing

Contents

Introduction

Switch meshing is a load-balancing technology that enhances reliability and performance in these ways:

- Provides significantly better bandwidth utilization than either Spanning Tree Protocol (MSTP) or standard port trunking.
- Uses redundant links that remain open to carry traffic, removing any single point of failure for disabling the network, and allowing quick responses to individual link failures. This also helps to maximize investments in ports and cabling.
- Unlike trunked ports, the ports in a switch mesh can be of different types and speeds (10 and 100 Mbps, gigabit, and 10 gigabit). For example, a 10Base-FL port and a 1GB port can be included in the same switch mesh.



The mesh-configured ports in switches 1-4 form a Switch Mesh Domain

Figure 5-1. Example of Switch Meshing

Finding the Fastest Path. Using multiple switches redundantly linked together to form a *meshed switch domain*, switch meshing dynamically distributes traffic across load-balanced switch paths by seeking the fastest paths for new traffic between nodes. In actual operation, the switch mesh periodically determines the best (lowest latency) paths, then assigns these paths as the need arises. The path assignment remains until the related MAC address entry times out. The mesh sees later traffic between the same nodes as new traffic, and may assign a different path, depending on conditions at the time. For example, at one time the best path from node A to node B is through switch 2. However, if traffic between node A and node B ceases long enough for the path assignment to age out, then the next time node A has traffic for node B, the assigned path between these nodes may be through switch 3 if network conditions have changed significantly.

Note

The **mac-age-time** parameter determines how long an inactive path assignment remains in memory. Refer to "System Information" in the chapter titled "Interface Access and System Information" in the *Management and Configuration Guide* for your switch.

Because Redundant Paths Are Active, Meshing Adjusts Quickly to Link Failures. If a link in the mesh fails, the fast convergence time designed into meshing typically has an alternate route selected in less than a second for traffic that was destined for the failed link.

Meshing Allows Scalable Responses to Increasing Bandwidth Demand. As more bandwidth is needed in a LAN backbone, another switch and another set of links can be added. This means that bandwidth is not limited by the number of trunk ports allowed in a single switch.

Meshing Features

Feature	Default	Menu	CLI	Web
Viewing a mesh configuration	n/a	5-9	5-12	n/a
Configuring a Switch Mesh	n/a	5-9	5-14	n/a

Switch Meshing Fundamentals

Terminology

Switch Mesh Domain. This is a group of meshed switch ports exchanging meshing protocol packets. Paths between these ports can have multiple redundant links without creating broadcast storms.

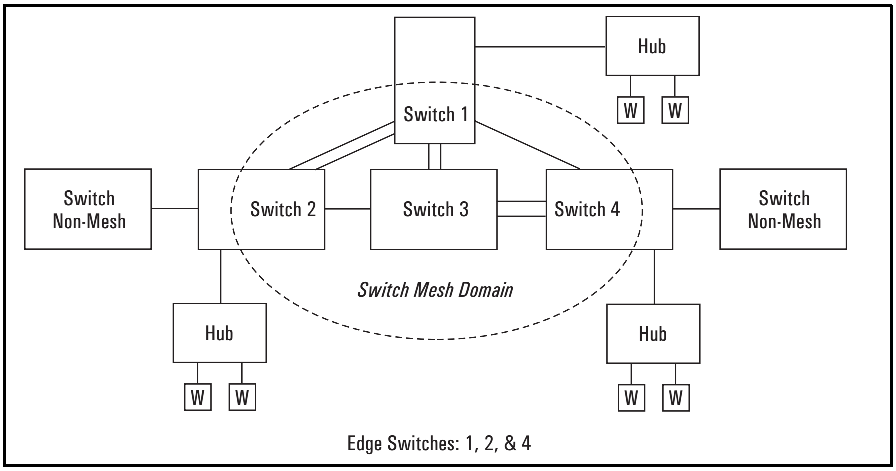


Figure 5-2. Example of a Switch Mesh Domain in a Network

Edge Switch. This is a switch that has some ports in the switch meshing domain and some ports outside of the domain. (See figure 5-2, above.)

Operating Rules

(See also "Mesh Design Optimization" on page 5-22.)

- A meshed switch can have some ports in the meshed domain and other ports outside the meshed domain. That is, ports within the meshed domain must be configured for meshing, while ports outside the meshed domain must not be configured for meshing.
- Meshed links must be point-to-point switch links.
- On any switch, all meshed ports belong to the same mesh domain.
- A switch can have up to 24 meshed ports.
- A mesh domain can include up to 12 switches.
- Up to five inter-switch, meshed hops are allowed in the path connecting two nodes through a switch mesh domain. A path of six or more meshed hops between two nodes is unusable. However, in most mesh topologies, there would normally be a shorter path available, and paths of five hops or fewer through the same mesh will continue to operate.
- Hub links between meshed switch links are not allowed.
- If the switch has multiple static VLANs and you configure a port for meshing, the port becomes a tagged member of all such VLANs . If you remove a port from meshing, it becomes an untagged member of only the default VLAN.
- A port configured as a member of a *static* trunk (LACP or Trunk) cannot also be configured for meshing.
- If a port belongs to a *dynamic* LACP trunk and you impose meshing on the port, it automatically ceases to be a member of the dynamic trunk.
- Meshing is not supported on ports configured with 802.1X access control.
- On a port configured for meshing, if you subsequently remove meshing from the port's configuration and reboot the switch, the port returns to its default configuration. (It *does not* revert to any non-default configuration it had before being configured for meshing).
- In a given mesh domain, switches in the same product family must run the same switch software version. For example, if you update the software version on one Series 5400zl switch, then you must update the software version on any other Series 5400zl switch in the mesh. ProCurve recommends that you always use the most recent software version available for the switches in your network.
- If meshing is configured on the switch, the routing features (IP routing, RIP, and OSPF) must be disabled. *That is, the switch's meshing and routing features cannot be enabled at the same time.*

- The spanning-tree configuration must be the same for all switches in the mesh (enabled or disabled). If spanning tree is enabled in the mesh, it must be the same version on all switches in the mesh: 802.1D, 802.1w, or 802.1s.
- If a switch in the mesh has GVRP enabled, then all switches in the mesh must have GVRP enabled. Otherwise, traffic on a dynamic VLAN may not pass through the mesh.
- If a switch in the mesh has a particular static vlan configured, then all switches in the mesh must have that static vlan configured.
- If a switch in the mesh has IGMP enabled, then all switches in the mesh must have IGMP enabled.
- If a switch in the mesh has LLDP enabled, then all switches in the mesh must have LLDP enabled.
- After adding or removing a port from the mesh, you must save the current configuration and reboot the switch in order for the change to take effect.
- Multiple meshed domains require separation by either a non-meshed switch or a non-meshed link. For example:

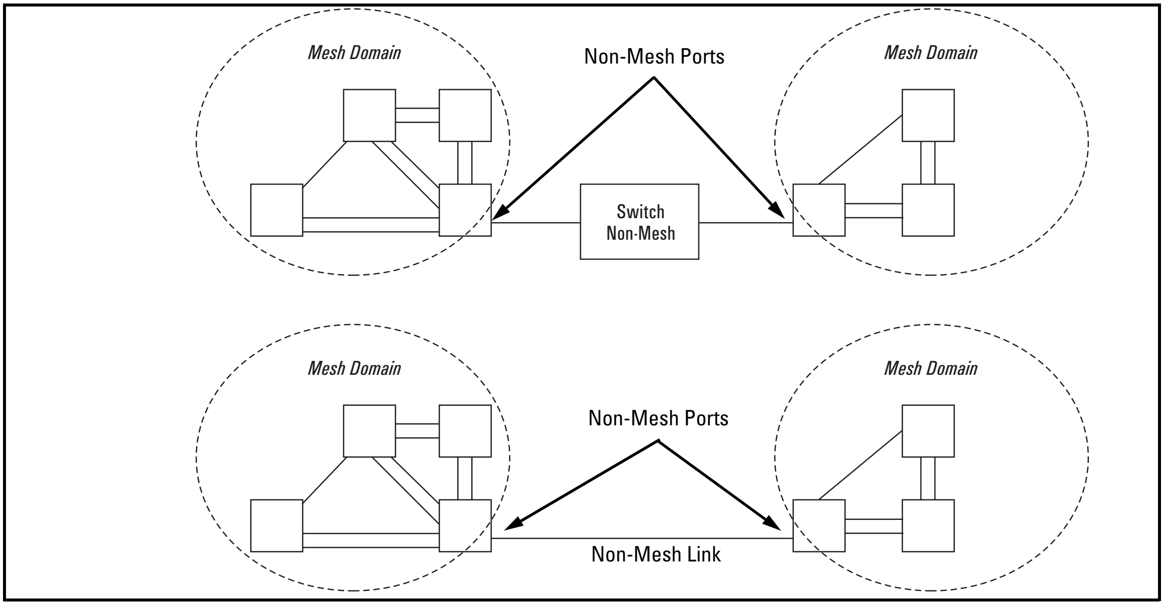


Figure 5-3. Example of Multiple Meshed Domains Separated by a Non-Mesh Switch or a Non-Mesh Link

- If GVRP is enabled, meshed ports in a switch become members of any dynamic VLANs created in the switch in the same way that they would if meshing was not configured in the switch. (For more on GVRP, refer to chapter 3, “GVRP”.)

Note

- A switch mesh domain (figure 5-1 on page 5-2) cannot include either a switch that is not configured for meshing, or a hub.
- Where a given pair of switches are linked with meshed ports, you must not also link the pair together through non-meshed ports unless you have also enabled STP, RSTP, or MSTP to prevent a loop from forming.

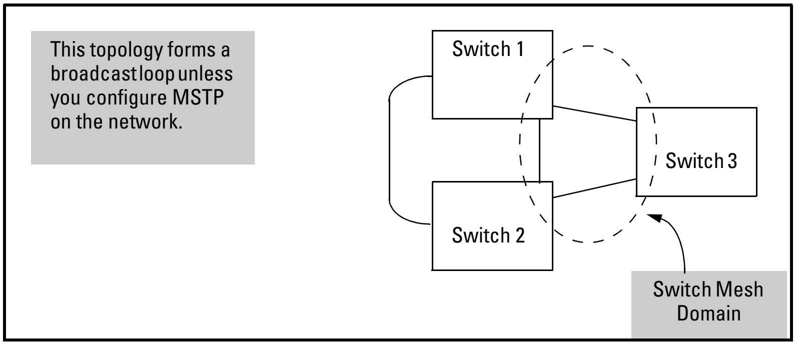

This topology forms a broadcast loop unless you configure MSTP on the network.
Switch 1
Switch 2
Switch 3
Switch Mesh Domain

Figure 5-4. Example of an Unsupported Topology

- The switch blocks traffic on a meshed port connected to a non-meshed port on another switch.
- Switch meshing does not allow trunked links (LACP or Trunk) between meshed ports.

Linking a non-mesh device or port into the mesh causes the meshed switch port(s) connected to that device to shut down.

Using a Heterogeneous Switch Mesh

You can use the switches covered in this guide with the ProCurve Series 5300xl switches in normal mode.

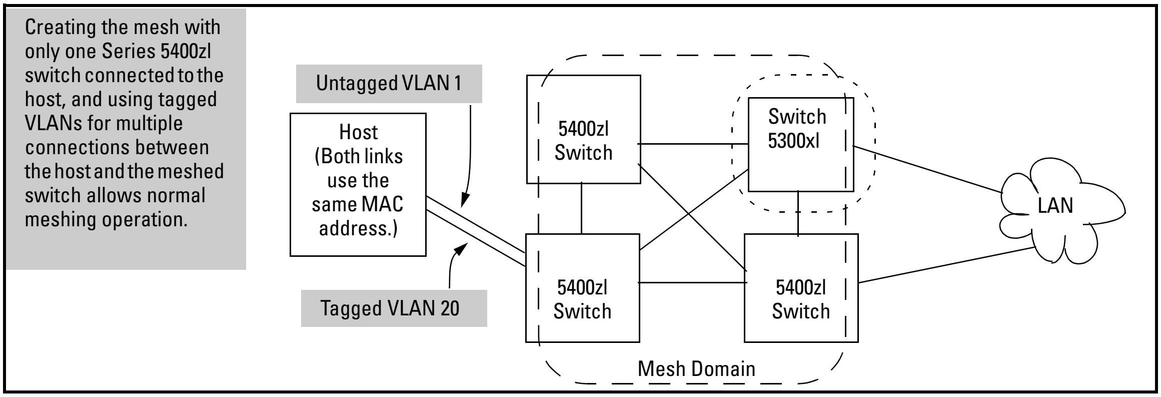

Creating the mesh with only one Series 5400zl switch connected to the host, and using tagged VLANs for multiple connections between the host and the meshed switch allows normal meshing operation.
Untagged VLAN 1
Host (Both links use the same MAC address.)
Tagged VLAN 20
5400zl Switch
Switch 5300xl
5400zl Switch
5400zl Switch
LAN
Mesh Domain

Figure 5-5. Example of a Supported Heterogeneous Topology in Normal Mode

Bringing Up a Switch Mesh Domain

When a meshed port detects a non-meshed port on the opposite end of a point-to-point connection, the link will be blocked. Thus, as you bring up switch meshing on various switches, you may temporarily experience blocked ports where meshed links should be running. These conditions should clear themselves after all switches in the mesh have been configured for meshing and their switches rebooted. To reduce the effect of blocked ports during bring-up, configure meshing and reboot the switches before installing the meshed switches in the network. Also, since adding (or removing) a meshed port requires a switch reboot to implement, you can avoid repeated system disruptions by waiting to implement the mesh until you have finished configuring meshing on all ports in your intended mesh domain.

Further Operating Information

Refer to “Operating Notes for Switch Meshing” on page 5-15.

Configuring Switch Meshing

Preparation

Before configuring switch meshing:

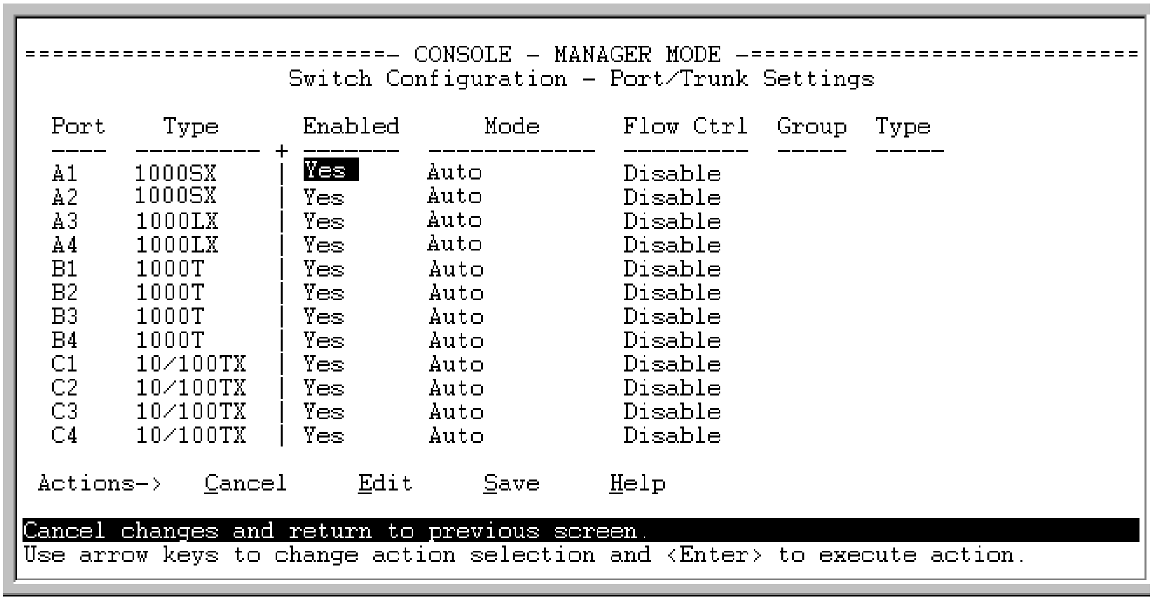
- Review the Operating Rules (page 5-5), and particularly the restrictions and requirements for using switch meshing in environments that include static trunks, multiple static VLANs, GVRP, IGMP, and MSTP.
- To avoid unnecessary system disruption, plan the mesh bring-up to minimize temporary port-blocking. (Refer to "Bringing Up a Switch Mesh Domain" on page 5-8.)
- To view the current switch mesh status on the switch, use the CLI **show mesh** command (page 5-12).

Menu: To Configure Switch Meshing

1. From the Main Menu, select:

   2. Switch Configuration
   　2. Port/Trunk Settings

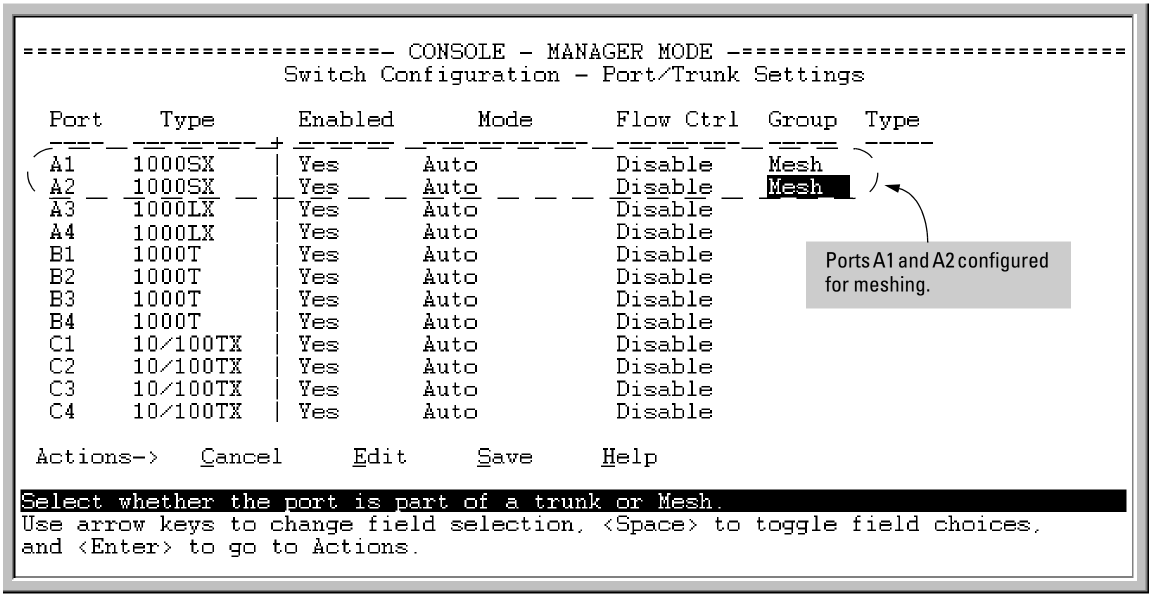
2. Press **[E]** (for **Edit**) to access the load balancing parameters.



```
==========================- CONSOLE - MANAGER MODE -===========================
                       Switch Configuration - Port/Trunk Settings

  Port    Type        Enabled      Mode        Flow Ctrl  Group  Type
  ----  --------- +  -------  ------------  ---------  -----  -----
  A1    1000SX    |  Yes      Auto          Disable
  A2    1000SX    |  Yes      Auto          Disable
  A3    1000LX    |  Yes      Auto          Disable
  A4    1000LX    |  Yes      Auto          Disable
  B1    1000T     |  Yes      Auto          Disable
  B2    1000T     |  Yes      Auto          Disable
  B3    1000T     |  Yes      Auto          Disable
  B4    1000T     |  Yes      Auto          Disable
  C1    10/100TX  |  Yes      Auto          Disable
  C2    10/100TX  |  Yes      Auto          Disable
  C3    10/100TX  |  Yes      Auto          Disable
  C4    10/100TX  |  Yes      Auto          Disable

 Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 5-6. Example of the Screen for Configuring Ports for Meshing

3. In the Group column, move the cursor to the port you want to assign to the switch mesh.
4. Press **[M]** to choose **Mesh** for the selected port.
5. Use the **up-arrow or down-arrow** key to select the next port you want to include in your mesh domain, then press **[M]** again. For example, if you were adding ports A1 and A2 to your mesh domain, the screen would appear similar to figure 5-7:



```
=========================- CONSOLE - MANAGER MODE -============================
                      Switch Configuration - Port/Trunk Settings

  Port    Type      Enabled      Mode       Flow Ctrl  Group  Type
  ---- --------- + -------  ------------  ---------  -----  -----
  A1    1000SX    | Yes      Auto         Disable    Mesh
  A2    1000SX    | Yes      Auto         Disable    Mesh
  A3    1000LX    | Yes      Auto         Disable
  A4    1000LX    | Yes      Auto         Disable
  B1    1000T     | Yes      Auto         Disable
  B2    1000T     | Yes      Auto         Disable
  B3    1000T     | Yes      Auto         Disable
  B4    1000T     | Yes      Auto         Disable
  C1    10/100TX  | Yes      Auto         Disable
  C2    10/100TX  | Yes      Auto         Disable
  C3    10/100TX  | Yes      Auto         Disable
  C4    10/100TX  | Yes      Auto         Disable

 Actions->   Cancel     Edit     Save     Help

Select whether the port is part of a trunk or Mesh.
Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

Ports A1 and A2 configured for meshing.

Figure 5-7. Example of Mesh Group Assignments for Several Ports

6. Repeat step 5 for all ports you want in the mesh domain.

Notes

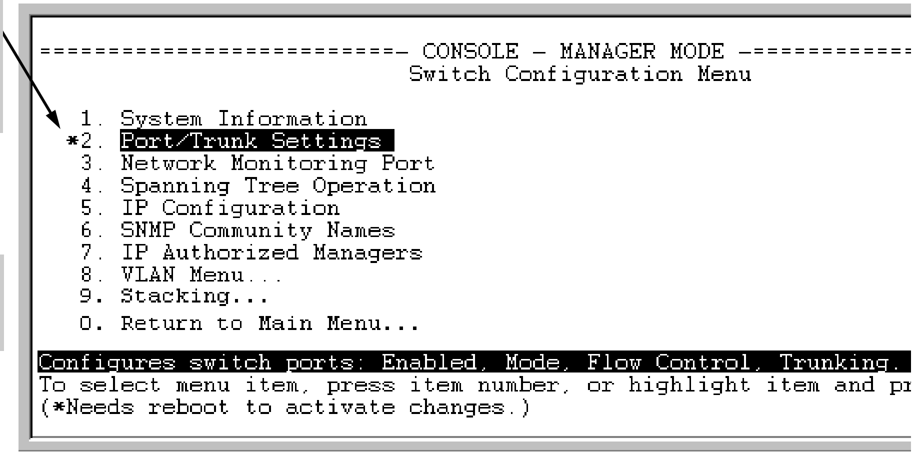
For meshed ports, leave the **Type** setting blank. (Meshed ports do not accept a **Type** setting.)

All meshed ports in the switch automatically belong to the same mesh domain. (See figure 5-2 on page 5-4.)

7. When you finish assigning ports to the switch mesh, press **[Enter],** then **[S]** (for **Save**). You will then see the following screen.

The asterisk indicates that you must reboot the switch to cause the **Mesh** configuration change to take effect.

Stacking is supported on the 3500yl and the 6200yl switches.



```
=========================- CONSOLE - MANAGER MODE -============
                           Switch Configuration Menu

   1. System Information
  *2. Port/Trunk Settings
   3. Network Monitoring Port
   4. Spanning Tree Operation
   5. IP Configuration
   6. SNMP Community Names
   7. IP Authorized Managers
   8. VLAN Menu...
   9. Stacking...
   0. Return to Main Menu...

Configures switch ports: Enabled, Mode, Flow Control, Trunking.
To select menu item, press item number, or highlight item and pr
(*Needs reboot to activate changes.)
```

Figure 5-8. After Saving a Mesh Configuration Change, Reboot the Switch

8. Press **[0]** to return to the Main menu.
9. To activate the mesh assignment(s) from the Main menu, reboot the switch by pressing the following keys:
   a. **[6]** (for **Reboot Switch**)
   b. Space bar (to select **Yes**).
   c. **13** (to start the reboot process).

(The switch cannot dynamically reconfigure ports to enable or disable meshing, so it is always necessary to reboot the switch after adding or deleting a port in the switch mesh.)

CLI: To View and Configure Switch Meshing

Port Status and Configuration Features

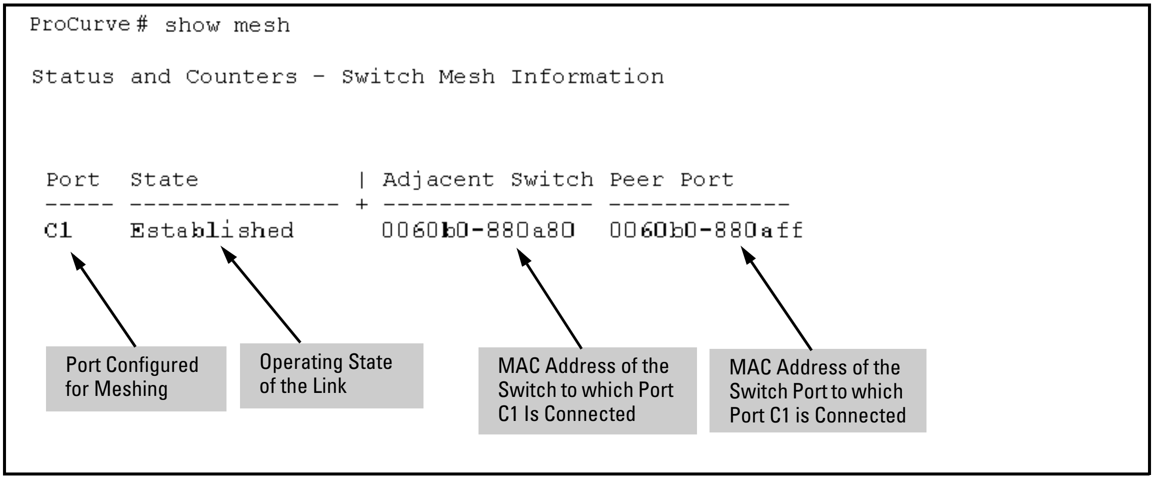
Feature	Default	Menu	CLI	Web
viewing switch mesh status	n/a	n/a	below	n/a
configuring switch meshing	Disabled	n/a		n/a

Viewing Switch Mesh Status

Syntax: show mesh

Lists the switch ports configured for meshing, along with the **State** *of each mesh-configured connection, the MAC address of the switch on the opposite end of the link (***Adjacent Switch***), and the MAC address of the port on the opposite end of the link (***Peer Port***).*

Reading the Show Mesh Output. For each port configured for meshing, the State column indicates whether the port has an active link to the mesh or is experiencing a problem.



```
ProCurve# show mesh

Status and Counters - Switch Mesh Information



 Port  State           | Adjacent Switch Peer Port
 ----- --------------- + --------------- -------------
 C1    Established       0060b0-880a80   0060b0-880aff
```

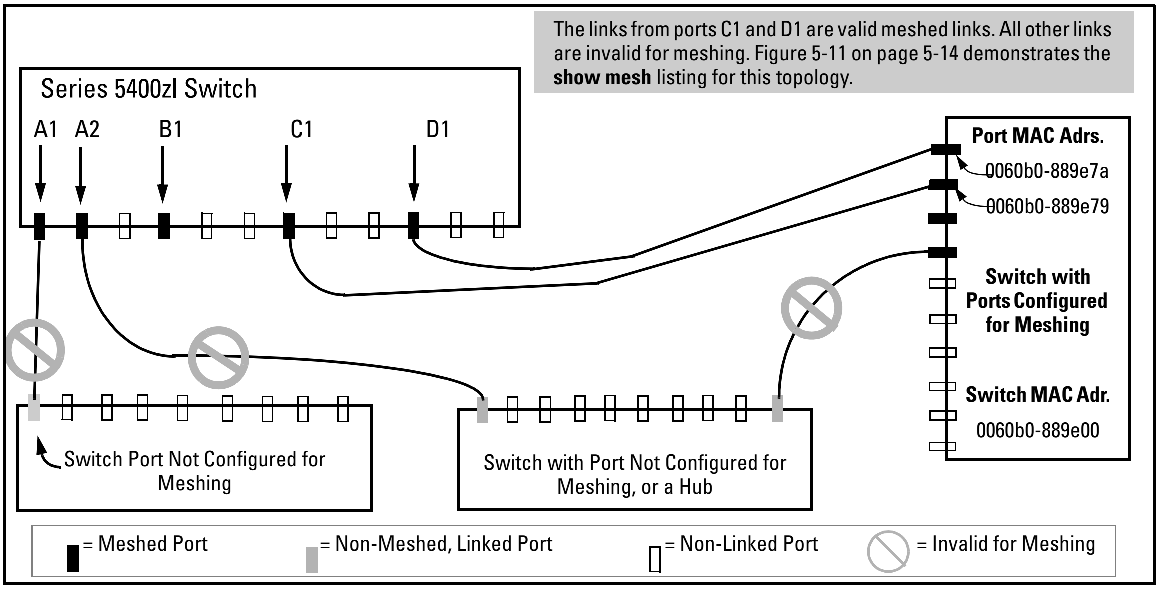
Port Configured for Meshing
Operating State of the Link
MAC Address of the Switch to which Port C1 Is Connected
MAC Address of the Switch Port to which Port C1 is Connected

Figure 5-9. Example of the Show Mesh Report

Table 5-1. State Descriptions for Show Mesh Output

State	Meaning
Established	The port is linked to a meshed port on another switch and meshing traffic is flowing across the link. The **show mesh** listing includes the MAC addresses of the adjacent switch and direct connection port on the adjacent switch.
Not Established	The port may be linked to a switch on a port that is not configured for meshing or has gone down.
Initial	The port has just come up as a meshed port and is trying to negotiate meshing.
Disabled	The port is configured for meshing but is not connected to another device.
Error	Indicates a multiple MAC-address error. This occurs when you have two or more mesh ports from the same switch linked together through a hub.
Topology Error	Two meshed switches are connected via a hub, and traffic from other, non-meshed devices, is flowing into the hub. The **show mesh** listing includes the MAC addresses of the adjacent switch and direct connection port on the adjacent switch.

Topology Example with Show Mesh. Suppose that you have the following topology:


The links from ports C1 and D1 are valid meshed links. All other links are invalid for meshing. Figure 5-11 on page 5-14 demonstrates the **show mesh** listing for this topology.

Series 5400zl Switch
A1
A2
B1
C1
D1
Port MAC Adrs.
0060b0-889e7a
0060b0-889e79
Switch with Ports Configured for Meshing
Switch MAC Adr.
0060b0-889e00
Switch Port Not Configured for Meshing
Switch with Port Not Configured for Meshing, or a Hub
= Meshed Port
= Non-Meshed, Linked Port
= Non-Linked Port
= Invalid for Meshing

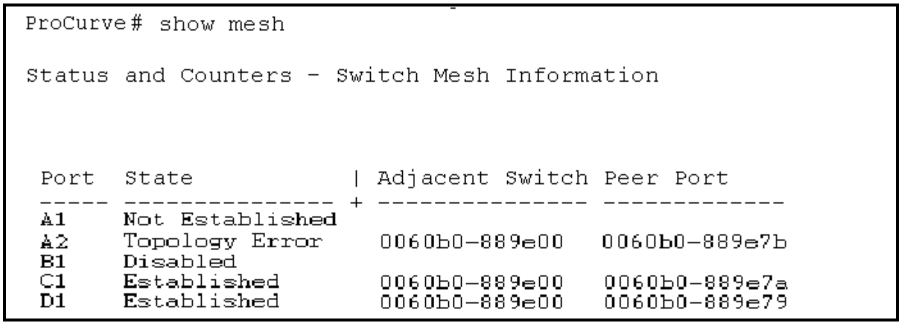
Figure 5-10. Example of a Meshed Topology with Some Mesh Ports Incorrectly Linked

Table 5-2 on page 5-14 describes the meshing operation in the above topology.

Table 5-2. Operating Details for Figure 5-10

Port	Meshing?	Connection
A1	Yes	Connected to a port that may not be configured for meshing
A2	Yes	Connected to a switch port on a device that is not configured for meshing (another switch, or a hub). In this case, the **Topology Error** message indicates that the switch detects a meshed port on another, non-adjacent device that is also connected to the non-meshed switch or hub. ***However, meshing will not operate properly through this connection.***
B1	Yes	Not connected to another device.
C1	Yes	Connected to a meshed port on the same adjacent switch as D1 with meshing operating properly.
D1	Yes	Connected to a meshed port on the same adjacent switch as C1 with meshing operating properly.

Figure 5-11 lists the show mesh display for the topology and meshing configuration in figure 5-10:



```
ProCurve# show mesh

Status and Counters - Switch Mesh Information



  Port  State            | Adjacent Switch Peer Port
  ----- ---------------- + --------------- -------------
  A1    Not Established
  A2    Topology Error     0060b0-889e00   0060b0-889e7b
  B1    Disabled
  C1    Established        0060b0-889e00   0060b0-889e7a
  D1    Established        0060b0-889e00   0060b0-889e79
```

Figure 5-11. Example of the Show Mesh Listing for the Topology in Figure 5-10

CLI: Configuring Switch Meshing

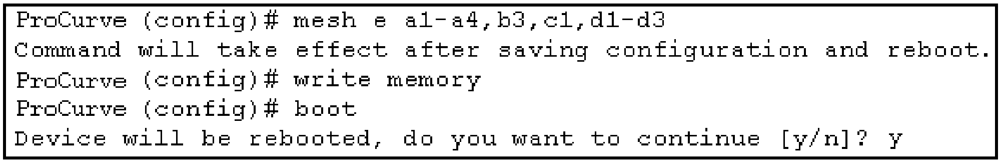
Syntax: [no] mesh [e] < *port-list* >

Enables or disables meshing operation on the specified ports.

All meshed ports on a switch belong to the same mesh domain. Thus, to configure multiple meshed ports on a switch, you need to:

1. Specify the ports you want to operate in the mesh domain.
2. Use **write memory** to save the configuration to the startup-config file.
3. Reboot the switch

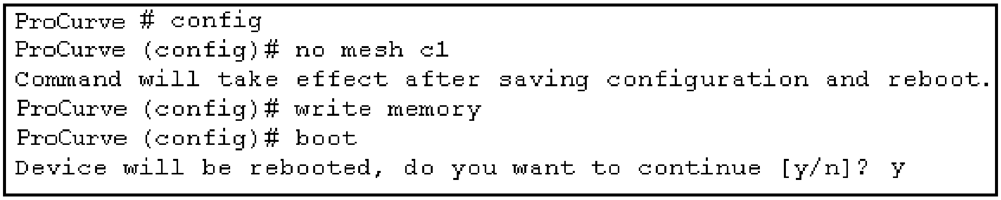
For example, to configure meshing on ports A1-A4, B3, C1, and D1-D3:



```
ProCurve (config)# mesh e a1-a4,b3,c1,d1-d3
Command will take effect after saving configuration and reboot.
ProCurve (config)# write memory
ProCurve (config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 5-12. Example of How To Configure Ports for Meshing

To remove a port from meshing, use the "**no**" version of **mesh**, followed by **write memory** and rebooting the switch. For example, to remove port C1 from the mesh:



```
ProCurve # config
ProCurve (config)# no mesh c1
Command will take effect after saving configuration and reboot.
ProCurve (config)# write memory
ProCurve (config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 5-13. Example of Removing a Port from the Mesh

Operating Notes for Switch Meshing

In a switch mesh domain traffic is distributed across the available paths with an effort to keep latency the same from path to path. The path selected at any time for a connection between a source node and a destination node is based on these latency and throughput cost factors:

- Outbound queue depth, or the current outbound load factor for any given outbound port in a possible path
- Port speed, such as 10Mbps versus 100Mbps; full-duplex or half-duplex
- Inbound queue depth, or how busy a destination switch is in a possible path
- Increased packet drops, indicating an overloaded port or switch

Paths having a lower cost will have more traffic added than those having a higher cost. Alternate paths and cost information is discovered periodically and communicated to the switches in the mesh domain. This information is used to assign traffic paths between devices that are newly active on the mesh.

This means that after an assigned path between two devices has timed out, new traffic between the same two devices may take a different path than previously used.

To display information on the operating states of meshed ports and the identities of adjacent meshed ports and switches, see “Viewing Switch Mesh Status” on page 5-12.

Flooded Traffic

Broadcast and multicast packets will always use the same path between the source and destination edge switches unless link failures create the need to select new paths. (Broadcast and multicast traffic entering the mesh from different edge switches are likely to take different paths.) When an edge switch receives a broadcast from a non-mesh port, it floods the broadcast out all its other non-mesh ports, but sends the broadcast out only those ports in the mesh that represent the path from that edge switch through the mesh domain. (Only one copy of the broadcast packet gets to each edge switch for broadcast out of its nonmeshed ports. This helps to keep the latency for these packets to each switch as low as possible.)

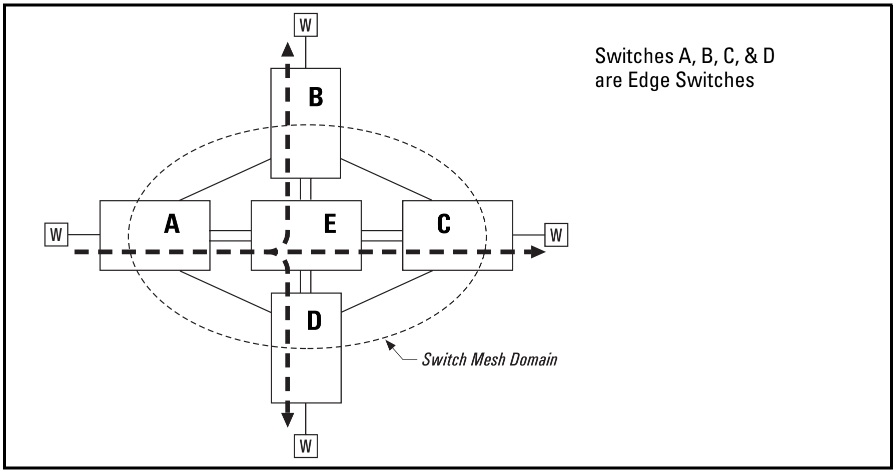


Figure 5-14. Example of a Broadcast Path Through a Switch Mesh Domain

Any mesh switches that are not edge switches will flood the broadcast packets only through ports (paths) that link to separate edge switches in the controlled broadcast tree. The edge switches that receive the broadcast will flood the broadcast out all non-meshed ports. Some variations on broadcast/multicast traffic patterns, including the situation where multiple VLANs are configured and a broadcast path through the mesh domain leads only to ports that are in the same VLAN as the device originating the broadcast.

Unicast Packets with Unknown Destinations

A meshed switch receiving a unicast packet with an unknown destination does not flood the packet onto the mesh. Instead, the switch sends a query on the mesh to learn the location of the unicast destination. The meshed switches then send 802.2 test packets through their non-meshed ports. After the unicast destination is found and learned by the mesh, subsequent packets having the same destination address will be forwarded. By increasing the **MAC Age Time** you can cause the switch address table to retain device addresses longer. (For more on **MAC Age Time**, refer to “System Information” in the chapter titled “Interface Access and System Information” in the *Management and Configuration Guide* for your switch.) Because the switches in a mesh exchange address information, this will help to decrease the number of unicast packets with unknown destinations, which improves latency within the switch mesh. Also, in an IP environment, ProCurve recommends that you configure IP addresses on meshed switches. This makes the discovery mechanism more robust, which contributes to decreased latency.

Spanning Tree Operation with Switch Meshing

Using MSTP with several switches and no switch meshing configured can result in unnecessarily blocking links and reducing available bandwidth. For example:

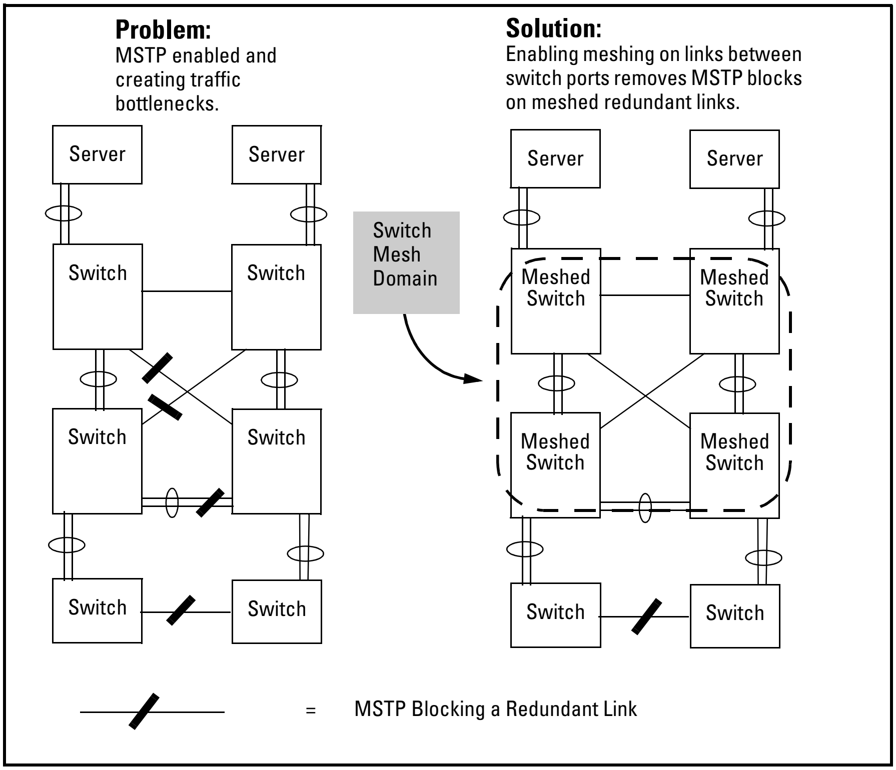

Problem:
MSTP enabled and creating traffic bottlenecks.
Solution:
Enabling meshing on links between switch ports removes MSTP blocks on meshed redundant links.
Server
Server
Switch
Switch
Switch
Switch
Switch
Switch
Switch Mesh Domain
Server
Server
Meshed Switch
Meshed Switch
Meshed Switch
Meshed Switch
Switch
Switch
= MSTP Blocking a Redundant Link

Figure 5-15. Example Using STP Without and With Switch Meshing

If you are going to use spanning-tree in a switch mesh, all switches in the mesh should be configured with the same type of spanning-tree: 802.1d/STP, 802.1w/RSTP, or 802.1s/MSTP. Spanning-Tree interprets a meshed domain as a single link. However, on edge switches in the domain, MSTP will manage non-meshed redundant links from other devices. For example:

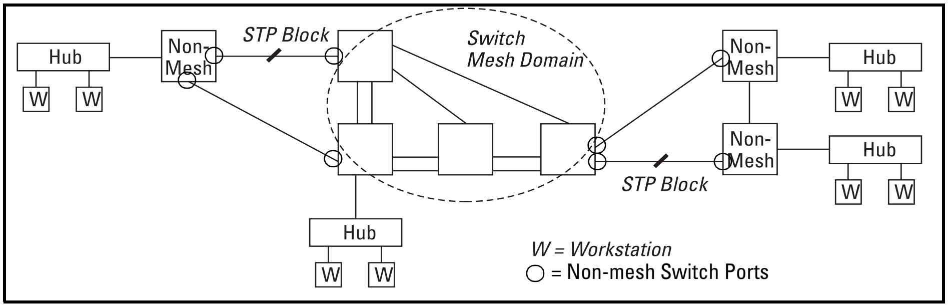


Figure 5-16. Connecting a Switch Mesh Domain to Non-Meshed Devices

Note on the Edge-Port Mode in MSTP

When using MSTP and interconnecting switches covered in this guide in a mesh with switches that are not in the mesh, all the non-mesh switch ports (as indicated in the figure above) should have the **edge-port** parameter disabled.

MSTP should be configured on non-mesh devices that use redundant links to interconnect with other devices or with multiple switch mesh domains. For example:

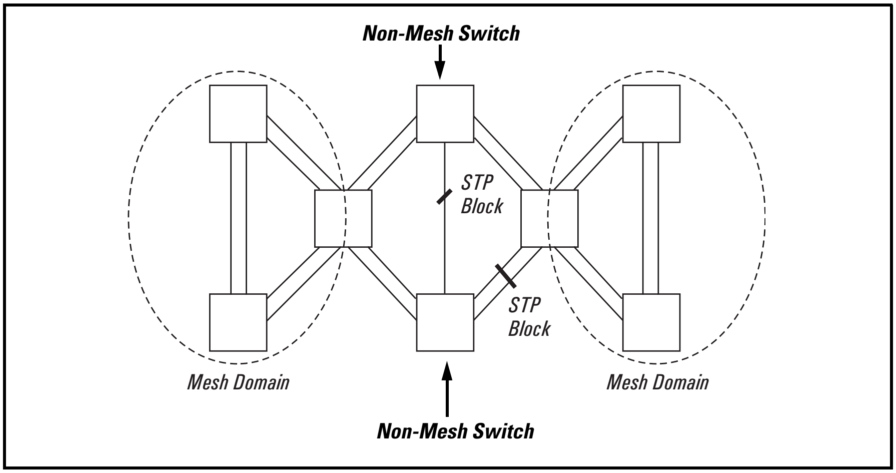


Figure 5-17. Interconnecting Switch Mesh Domains with Redundant Links

In the above case of multiple switch meshes linked with redundant trunks there is the possibility that spanning-tree will temporarily block a mesh link. This is because it is possible for spanning-tree to interpret the cost on an external trunked link to be less than the cost on a meshed link. However, if this condition occurs, the meshed switch that has a blocked link will automatically increase the cost on the external (non-meshed) link to the point where spanning tree will block the external link and unblock the meshed link. This process typically resolves itself in approximately 30 seconds.

Caution

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default spanning-tree parameter settings are usually adequate for spanning tree operation. Also, because incorrect spanning tree settings can adversely affect network performance, you should not make changes unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default MSTP timer settings (**Hello Time** and **Forward Delay**) are usually adequate for MSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the MSTP **Hello Time** and **Forward Delay** timers can cause unnecessary topology changes and end-node connectivity problems.

For more on spanning-tree, refer to the chapter titled "Multiple Instance Spanning-Tree Operation" in this guide. Also, you may want to examine the IEEE 802.1d, 802.1w, or 802.1s standards, depending on which version of spanning-tree you are using. The switches covered in this guide use 802.1s.

Filtering/Security in Meshed Switches

Because paths through the mesh can vary with network conditions, configuring filters on meshed ports can create traffic problems that are difficult to predict, and is not recommended. However, configuring filters on nonmeshed ports in an edge switch provides you with control and predictability.

IP Multicast (IGMP) in Meshed Switches

Like trunked ports, the switch mesh domain appears as a single port to IGMP. However, unlike trunked ports, IGMP protocol and multicast traffic may be sent out over several links in the mesh in the same manner as broadcast packets.

Static VLANs

In a network having a switch mesh domain and multiple static VLANs configured, all static VLANs must be configured on each meshed switch, even if no ports on the switch are assigned to any VLAN. (The switch mesh is a member of all static VLANs configured on the switches in the mesh.)

When static VLANs are configured, the mesh is seen as a single entity by each VLAN. All ports in the mesh domain are members of all VLANs and can be used to forward traffic for any VLAN. However, the non-mesh ports on edge switches that allow traffic to move between the mesh and non-meshed devices belong to specific VLANs and do not allow packets originating in a specific VLAN to enter non-meshed devices that do not belong to that same VLAN. (It is necessary to use a router to communicate between VLANs.) For example, in the following illustration, traffic from host A entering the switch mesh can only exit the mesh at the port for hosts B and E. Traffic from host A for any other host (such as C or D) will be dropped because only hosts B and E are in the same VLAN as host A.

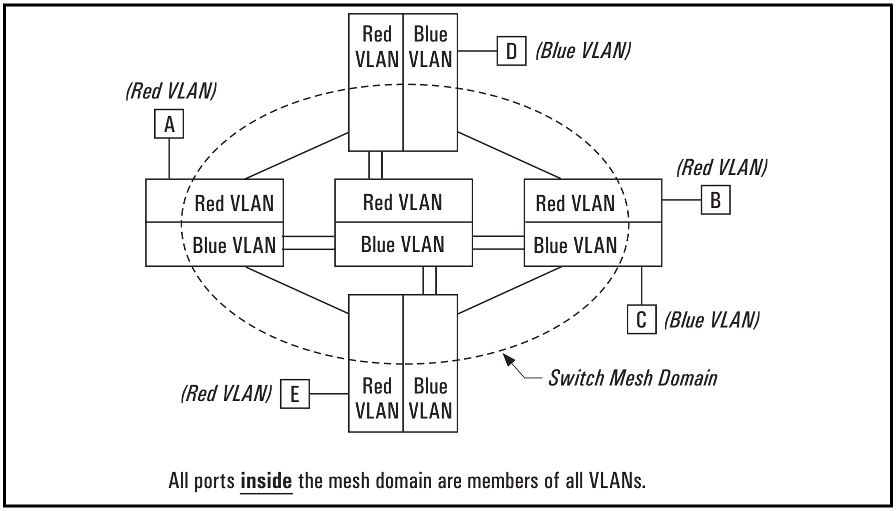


Figure 5-18. VLAN Operation with a Switch Mesh Domain

Dynamic VLANs

If GVRP is enabled, meshed ports in a switch become members of any dynamic VLANs created in the switch in the same way that they would if meshing was not configured in the switch. (For more on GVRP, refer to chapter 3, “GVRP”.)

Jumbo Packets

If you enable jumbo traffic on any VLAN, then all meshed ports on the switch will be enabled to support jumbo traffic. (On a given meshed switch, every meshed port becomes a member of every VLAN configured on the switch.) If a port in a meshed domain does not belong to any VLANs configured to support jumbo traffic, then the port drops any jumbo packets it receives from other devices. In this regard, if a mesh domain includes any ProCurve 6200yl switches, Series 5400zl switches, Series 3500yl switches, Series 3400cl or Series 6400cl switches that are configured to support jumbo traffic, only these switches can transmit and receive jumbo packets. Other switch models in the mesh will drop jumbo packets as they are not supported by those switches. For more information on jumbo packets, refer to the chapter titled “Port Traffic Controls” in the *Management and Configuration Guide* for your switch.

Mesh Design Optimization

Mesh performance can be enhanced by using mesh designs that are as small and compact as possible while still meeting the network design requirements. The following are limits on the design of meshes and have not changed:

1. Any switch in the mesh can have up to 24 meshed ports.
2. A mesh domain can contain up to 12 switches.
3. Up to 5 inter-switch meshed hops are allowed in the path connecting two nodes.
4. A fully interconnected mesh domain can contain up to 5 switches.

Mesh performance can be optimized by keeping the number of switches and the number of possible paths between any two nodes as small as possible. As mesh complexity grows, the overhead associated with dynamically calculating and updating the cost of all of the possible paths between nodes grows exponentially. Cost discovery packets are sent out by each switch in the mesh every 30 seconds and are flooded to all mesh ports. Return packets include a cost metric based on inbound and outbound queue depth, port speed, number of dropped packets, etc. Also, as mesh complexity grows, the number of hops over which a downed link has to be reported may increase, thereby increasing the reconvergence time.

The simplest design is the two-tier design because the number of possible paths between any two nodes is kept low and any bad link would have to be communicated only to it's neighbor switch.

Other factors affecting the performance of mesh networks include the number of destination addresses that have to be maintained, and the overall traffic levels and patterns. However a conservative approach when designing new mesh implementations is to use the two-tier design and limit the mesh domain to eight switches where possible.

As shown here, meshing allows multiple redundant links between switches in the domain, and more than one link is allowed between any two switches in the domain.

Note also that a switch can have up to 24 ports configured for meshing.

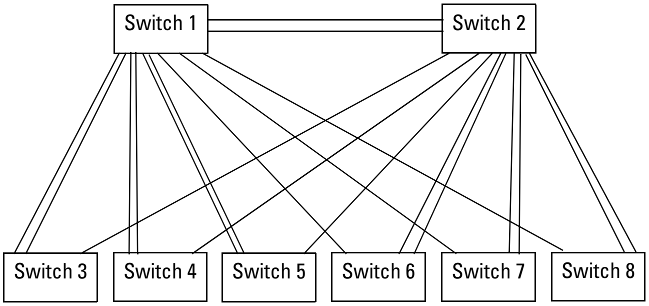
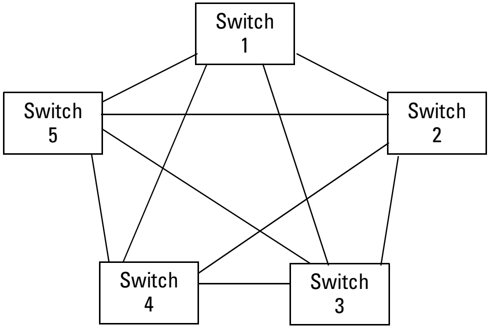


Figure 5-19. Example of a Two-Tier Mesh Design



As noted above, a switch can have up to 24 ports configured for meshing.

Figure 5-20. Example of a Fully Interconnected Mesh with the Maximum Switch Count

Other factors affecting the performance of mesh networks include the number of destination addresses that have to be maintained, and the overall traffic levels and patterns. However a conservative approach when designing new mesh implementations is to use the two-tier design and limit the mesh domain to eight switches where possible.

Other Requirements and Restrictions

- **Mesh Support Within the Domain:** All switches in the mesh domain, including edge switches, must support the ProCurve switch meshing protocol.
- **Switch Hop Count in the Mesh Domain:** A maximum of five (meshed) switch hops is allowed in the path connecting two nodes in a switch mesh domain. A path of six meshed hops is unusable. However, this does not interfere with other, shorter paths in the same domain.
- **Connecting Mesh Domains:** To connect two separate switch meshing domains, you must use non-meshed ports. (The non-meshed link can be a port trunk or a single link.) Refer to figure 5-3 on page 5-6.
- **Multiple Links Between Meshed Switches:** Multiple mesh ports can be connected between the same two switches, to provide higher bandwidth. Each port that you want in the mesh domain should be configured as **Mesh** (and not as a trunk—**Trk**). Note that if you configure a port as **Mesh**, there is no "Type" selection for that port.
- **Network Monitor Port:** If a network monitor port is configured, broadcast packets may be duplicated on this port if more than one port is being monitored and switch meshing is enabled.
- **Compatibility with Other Switches:** The switches covered in this guide operate with the Series 5300xl switches in normal mode.
- **Rate-Limiting Not Recommended on Meshed Ports: Rate-Limiting can reduce the efficiency of paths through a mesh domain.**

(See also "Operating Rules" on page 5-5.)

For additional information on troubleshooting meshing problems, refer to "Using a Heterogeneous Switch Mesh" on page 5-7 and "Mesh-Related Problems" in appendix C, "Troubleshooting" of the Management and Configuration Guide for your switch.

Quality of Service (QoS): Managing Bandwidth More Effectively

Contents

Introduction

QoS Feature	Default	Menu	CLI	Web
UDP/TCP Priority	Disabled	—	page 6-16	Refer to the Online Help.
IP-Device Priority	Disabled	—	page 6-23	“
IP Type-of-Service Priority	Disabled	—	page 6-29	“
LAN Protocol Priority	Disabled	—	page 6-41	“
VLAN-ID Priority	Disabled	—	page 6-43	“
Source-Port Priority	Disabled	—	page 6-49	“
DSCP Policy Table	Various	—	page 6-54	“

As the term suggests, *network policy* refers to the network-wide controls you can implement to:

- Ensure uniform and efficient traffic handling throughout your network, while keeping the most important traffic moving at an acceptable speed, regardless of current bandwidth usage.
- Exercise control over the priority settings of inbound traffic arriving in and travelling through your network.

Adding bandwidth is often a good idea, but it is not always feasible and does not completely eliminate the potential for network congestion. There will always be points in the network where multiple traffic streams merge or where network links will change speed and capacity. The impact and number of these congestion points will increase over time as more applications and devices are added to the network.

When (not *if*) network congestion occurs, it is important to move traffic on the basis of relative importance. However, without *Quality of Service* (QoS) prioritization, less important traffic can consume network bandwidth and slow down or halt the delivery of more important traffic. That is, without QoS, most traffic received by the switch is forwarded with the same priority it had upon entering the switch. In many cases, such traffic is “normal” priority and competes for bandwidth with all other normal-priority traffic, regardless of its relative importance to your organization’s mission.

This section gives an overview of QoS operation and benefits, and describes how to configure QoS in the console interface.

Quality of Service is a general term for classifying and prioritizing traffic throughout a network. That is, QoS enables you to establish an end-to-end traffic priority policy to improve control and throughput of important data. You can manage available bandwidth so that the most important traffic goes first. For example, you can use Quality of Service to:

- Upgrade or downgrade traffic from various servers.
- Control the priority of traffic from dedicated VLANs or applications.
- Change the priorities of traffic from various segments of your network as your business needs change.
- Set priority policies in edge switches in your network to enable traffic-handling rules across the network.

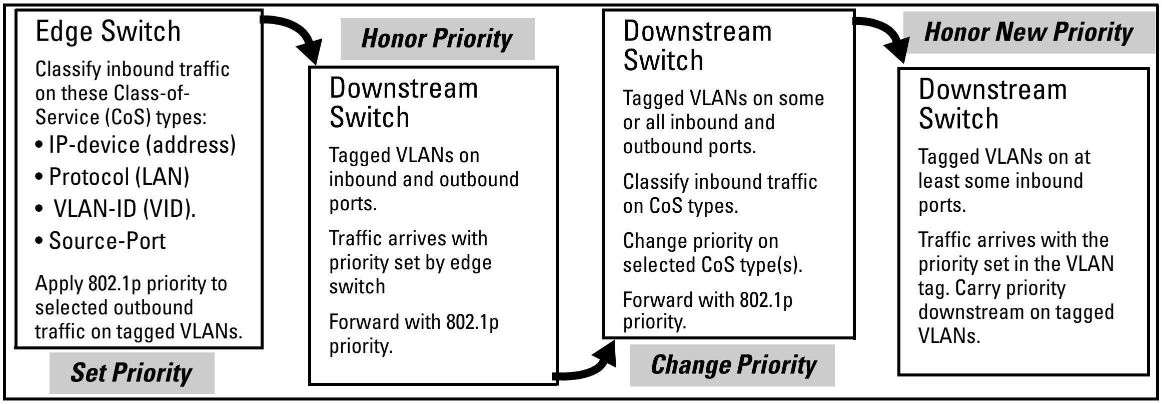

Edge Switch
Classify inbound traffic on these Class-of-Service (CoS) types:
• IP-device (address)
• Protocol (LAN)
• VLAN-ID (VID).
• Source-Port
Apply 802.1p priority to selected outbound traffic on tagged VLANs.
Set Priority
Honor Priority
Downstream Switch
Tagged VLANs on inbound and outbound ports.
Traffic arrives with priority set by edge switch
Forward with 802.1p priority.
Downstream Switch
Tagged VLANs on some or all inbound and outbound ports.
Classify inbound traffic on CoS types.
Change priority on selected CoS type(s).
Forward with 802.1p priority.
Change Priority
Honor New Priority
Downstream Switch
Tagged VLANs on at least some inbound ports.
Traffic arrives with the priority set in the VLAN tag. Carry priority downstream on tagged VLANs.

Figure 6-1. Example of 802.1p Priority Based on CoS (Class-of-Service) Types and Use of VLAN Tags

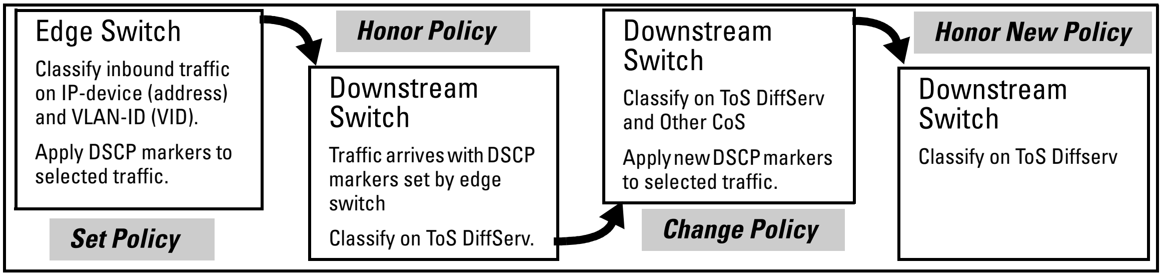

Edge Switch
Classify inbound traffic on IP-device (address) and VLAN-ID (VID).
Apply DSCP markers to selected traffic.
Set Policy
Honor Policy
Downstream Switch
Traffic arrives with DSCP markers set by edge switch
Classify on ToS DiffServ.
Downstream Switch
Classify on ToS DiffServ and Other CoS
Apply new DSCP markers to selected traffic.
Change Policy
Honor New Policy
Downstream Switch
Classify on ToS Diffserv

Figure 6-2. Example Application of Differentiated Services Codepoint (DSCP) Policies

At the edge switch, QoS classifies certain traffic types and in some cases applies a DSCP policy. At the next hop (downstream switch) QoS honors the policies established at the edge switch. Further downstream, another switch may reclassify some traffic by applying new policies, and yet other downstream switches can be configured to honor the new policies.

QoS is implemented in the form of rules or policies that are configured on the switch. While you can use QoS to prioritize only the outbound traffic while it is moving through the switch, you derive the maximum benefit by using QoS in an 802.1Q VLAN environment (with 802.1p priority tags) or in an untagged VLAN environment (with DSCP policies) where QoS can set priorities that downstream devices can support without re-classifying the traffic.

By prioritizing traffic, QoS supports traffic growth on the network while optimizing the use of existing resources—and delaying the need for further investments in equipment and services. That is, QoS enables you to:

- Specify which traffic has higher or lower priority, regardless of current network bandwidth or the relative priority setting of the traffic when it is received on the switch.
- Change (upgrade or downgrade) the priority of outbound traffic.
- Override "illegal" packet priorities set by upstream devices or applications that use 802.1Q VLAN tagging with 802.1p priority tags.
- Avoid or delay the need to add higher-cost NICs (network interface cards) to implement prioritizing. (Instead, control priority through network policy.)

QoS on the switches covered in this guide support these types of traffic marking:

- **802.1p prioritization:** Controls the outbound port queue priority for traffic leaving the switch, and (if traffic exits through a VLAN-tagged port) sends the priority setting with the individual packets to the downstream devices.
- **IP Type-of-Service (ToS):** Enables the switch to set, change, and honor prioritization policies by using the Differentiated Services (diffserv) bits in the ToS byte of IPv4 packet headers.

Terminology

Term	Use in This Document
802.1p priority	A traffic priority setting carried by a VLAN-tagged packet moving from one device to another through ports that are tagged members of the VLAN to which the packet belongs. This setting can be from 0 - 7. The switch handles an outbound packet on the basis of its 802.1p priority. However, if the packet leaves the switch through a VLAN on which the port is an untagged member, this priority is dropped, and the packet arrives at the next, downstream device without an 802.1p priority assignment.
802.1Q field	A four-byte field that is present in the header of Ethernet packets entering or leaving the switch through a port that is a tagged member of a VLAN. This field includes an 802.1p priority setting, a VLAN tag, or ID number (VID), and other data. A packet entering or leaving the switch through a port that is an untagged member of the outbound VLAN does not have this field in its header and thus does not carry a VID or an 802.1p priority. See also "802.1p priority".
codepoint	Refer to DSCP, below.
downstream device	A device linked directly or indirectly to an outbound switch port. That is, the switch sends traffic to downstream devices.
DSCP	***Differentiated Services Codepoint.*** (Also termed ***codepoint***.) A DSCP is comprised of the upper six bits of the ToS (Type-of-Service) byte in IP packets. There are 64 possible codepoints. In the default QoS configuration for the switches covered in this guide, some codepoints are configured with default 802.1p priority settings for Assured-Forwarding and Expedited Forwarding. All other codepoints are unused (and listed with **No-override** for a priority).
DSCP policy	A DSCP configured with a specific 802.1p priority (0- 7). (Default: **No-override**). Using a DSCP policy, you can configure the switch to assign priority to IP packets. That is, for an IP packet identified by the specified classifier, you can assign a new DSCP and an 802.1p priority (0-7). For more on DSCP, refer to "Details of QoS IP Type-of-Service" on page 6-38. For the DSCP map, see figure 6-21 on page 6-39.
edge switch	In the QoS context, this is a switch that receives traffic from the edge of the LAN or from outside the LAN and forwards it to devices within the LAN. Typically, an edge switch is used with QoS to recognize packets based on classifiers such as TCP/UDP application type, IP-device (address), Protocol (LAN), VLAN-ID (VID), and Source-Port (although it can also be used to recognize packets on the basis of ToS bits). Using this packet recognition, the edge switch can be used to set 802.1p priorities or DSCP policies that downstream devices will honor.
inbound port	Any port on the switch through which traffic enters the switch.
IP Options	In an IPv4 packet, optional, these are extra fields in the packet header.
IP-precedence bits	The upper three bits in the Type of Service (ToS) field of an IP packet.
IPv4	Version 4 of the IP protocol.
outbound packet	A packet leaving the switch through any LAN port.
outbound port	Any port on the switch through which traffic leaves the switch.

Term	Use in This Document
outbound port queue	For any port, a buffer that holds outbound traffic until it can leave the switch through that port. There are eight outbound queues for each port in the switch. Queue 8 is the highest priority queue; queue 1 is the lowest priority queue. Traffic in a port's high priority queue leaves the switch before any traffic in the port's medium or low priority queues.
re-marking (DSCP re-marking)	Assigns a new QoS policy to an outbound packet by changing the DSCP bit settings in the ToS byte.
tagged port membership	Identifies a port as belonging to a specific VLAN and enables VLAN-tagged packets belonging to that VLAN to carry an 802.1p priority setting when outbound from that port. Where a port is an untagged member of a VLAN, outbound packets belonging to that VLAN do not carry an 802.1p priority setting.
Type-of-Service (ToS) byte	Comprised of a three-bit (high-order) precedence field and a five-bit (low-order) Type-of-Service field. Later implementations may use this byte as a six-bit (high-order) Differentiated Services field and a two-bit (low-order) reserved field. See also "IP-precedence bits" and DSCP elsewhere in this table.
upstream device	A device linked directly or indirectly to an inbound switch port. That is, the switch receives traffic from upstream devices.

Overview

QoS settings operate on two levels:

- **Controlling the priority of outbound packets moving through the switch:** Each switch port has eight outbound traffic queues; queue one has the lowest priority and queue eight has the highest priority. Packets leave the switch port on the basis of their queue assignment and whether any higher queues are empty:

6-1. Port Queue Exit Priorities

Port Queue and 802.1p Priority Values	Priority for Exiting From the Port
Low (1)	Eighth
Low (2)	Seventh
Normal (0)	Sixth
Normal (3)	Fifth
Medium (4)	Fourth
Medium (5)	Third
High (6)	Second
High (7)	First

A QoS configuration enables you to set the outbound priority queue to which a packet is sent. (In an 802.1Q VLAN environment with VLAN-tagged ports, if QoS is *not* configured on the switch, but *is* configured on an upstream device, the priorities carried in the packets determine the forwarding queues in the switch.)

- **Configuring a priority for outbound packets and a service (priority) policy for use by downstream devices:**
  - **DSCP Policy:** This feature enables you to set a priority policy in outbound IP packets. (You can configure downstream devices to read and use this policy.) This method is not dependent on VLAN-tagged ports to carry priority policy to downstream devices, and can:
    - Change the codepoint (the upper six bits) in the ToS byte.
    - Set a new 802.1p priority for the packet.

    (Setting DSCP policies requires IPv4 inbound packets. Refer to the “IPv4” entry under “Terminology” on page 6-6.)
  - **802.1p Priority Rules:** An outbound, VLAN-tagged packet carries an 802.1p priority setting that was configured (or preserved) in the switch. This priority setting ranges from 0 to 7, and can be used by downstream devices having up to eight outbound port queues. Thus, while packets within the switch move at the eight priority levels shown in table 6-1, above, they still can carry an 802.1p priority that can be used by downstream devices having more or less than the eight priority levels in the switches covered in this guide. Also, if the packet enters the switch with an 802.1p priority setting, QoS can override this setting if configured with an 802.1p priority rule to do so.

Notes:

If your network uses only one VLAN (and therefore does not require VLAN-tagged ports) you can still preserve 802.1p priority settings in your traffic by configuring the ports as tagged VLAN members on the links between devices you want to honor traffic priorities.

Rule and Policy Limits: The switches covered in this guide allow up to **250** 802.1p priority rules and/or DSCP policies in any combination. For more information, refer to “Maximum QoS Configuration Entries” under “QoS Operating Notes and Restrictions” on page 6-63.

You can configure a QoS priority of 0 through 7 for an outbound packet. When the packet is then sent to a port, the QoS priority determines which outbound queue the packet uses:

6-2. QoS Priority Settings and Operation

QoS Priority Setting	Outbound Port Queue
1 - 2	low priority (1, 2)
0 - 3	normal priority (3, 4)
4 - 5	medium priority (5, 6)
6 - 7	high priority (7, 8)

If a packet is not in a VLAN-tagged port environment, then the QoS settings in table 6-2 control only to which outbound queue the packet goes. Without VLAN tagging, no 802.1p priority is added to the packet for downstream device use. But if the packet is in a VLAN-tagged environment, then the above setting is also added to the packet as an 802.1p priority for use by downstream devices and applications (shown in table 6-3). In either case, an IP packet can also carry a priority policy to downstream devices by using DSCP-marking in the ToS byte.

6-3. Mapping Switch QoS Priority Settings to Device Queues

Priority Setting	Outbound Port Queues in the Switch	802.1p Priority Setting Added to Tagged VLAN Packets Leaving the Switch	Queue Assignment in Downstream Devices With:		
			8 Queues	3 Queues	2 Queues
1	Queue 1	1 (low priority)	Queue 1	Queue 1	Queue 1
2	Queue 2	2	Queue 2		
0	Queue 3	0 (normal priority)	Queue 3	Queue 2	
3	Queue 4	3	Queue 4		
4	Queue 5	4 (medium priority)	Queue 5	Queue 3	Queue 2
5	Queue 6	5	Queue 6		
6	Queue 7	6 (high priority)	Queue 7		
7	Queue 8	7	Queue 8		

Classifiers for Prioritizing Outbound Packets

Note On Using Multiple Criteria

ProCurve recommends that you configure a minimum number of the available QoS classifiers for prioritizing any given packet type. Increasing the number of active classifier options for a packet type increases the complexity of the possible outcomes and consumes switch resources.

Packet Classifiers and Evaluation Order

The switches covered in this guide provide six QoS classifiers (packet criteria) you can use to configure QoS priority.

6-4. Classifier Search Order and Precedence

Search Order	Precedence	QoS Classifier Type
1	1 (highest)	UDP/TCP Application Type (port)
2	2	Device Priority (destination or source IP address)
3	3	IP Type of Service (ToS) field (IP packets only)
4	4	Protocol Priority (IP, IPX, ARP, AppleTalk, SNA, and NetBeui)
5	5	VLAN Priority
6	6	Incoming source-port on the switch
7	7 (lowest)	Incoming 802.1p Priority (present in tagged VLAN environments)

Where multiple classifier types are configured, a switch uses the highest-to-lowest search order shown in table 6-4 to identify the highest-precedence classifier to apply to any given packet. When a match between a packet and a classifier is found, the switch applies the QoS policy configured for that classifier and the packet is handled accordingly.

Note that on the switches covered in this guide, if the switch is configured with multiple classifiers that address the same packet, the switch uses only the QoS configuration for the QoS classifier that has the highest precedence. In this case, the QoS configuration for another, lower-precedence classifier that may apply is ignored. For example, if QoS assigns high priority to packets belonging to VLAN 100, but normal priority to all IP protocol packets, since protocol priority (4) has precedence over VLAN priority (5), IP protocol packets on VLAN 100 will be set to normal priority.

Preparation for Configuring QoS

Preserving 801.1p Priority

QoS operates in VLAN-tagged and VLAN-untagged environments. If your network does not use multiple VLANs, you can still implement the 802.1Q VLAN capability for packets to carry their 802.1p priority to the next downstream device. To do so, configure ports as VLAN-tagged members on the links between switches and routers in your network infrastructure.

6-5. Summary of QoS Capabilities

Outbound Packet Options	Port Membership in VLANs	
	Tagged	Untagged
Control Port Queue Priority for Packet Types	Yes	Yes
Carry 802.1p Priority Assignment to Next Downstream Device	Yes	No
Carry DSCP Policy to Downstream Devices. The policy includes:	Yes [1]	Yes [1]
Assigning a ToS Codepoint		
Assigning an 802.1p Priority [2] to the Codepoint		

[1] Except for non-IPv4 packets or packets processed using either the Layer 3 Protocol or QoS IP-Precedence methods, which do not include the DSCP policy option. Also, to use a service policy in this manner, the downstream devices must be configured to interpret and use the DSCP carried in the IP packets.

[2] This priority corresponds to the 802.1p priority scheme and is used to determine the packet's port queue priority. When used in a VLAN-tagged environment, this priority is also assigned as the 802.1p priority carried outbound in packets having an 802.1Q field in the header.

Steps for Configuring QoS on the Switch

1. Determine the QoS policy you want to implement. This includes analyzing the types of traffic flowing through your network and identifying one or more traffic types to prioritize. In order of QoS precedence, these are:
   a. UDP/TCP applications
   b. Device Priority—destination or source IP address (Note that destination has precedence over source. See 6-6.)
   c. IP Type-of-Service Precedence Bits (Leftmost three bits in the ToS field of IP packets)
   d. IP Type-of-Service Differentiated Service bits (Leftmost six bits in the ToS field of IP packets)
   e. Protocol Priority

f. VLAN Priority (requires at least one tagged VLAN on the network)

g. Source-Port

h. Incoming 802.1p Priority (requires at least one tagged VLAN on the network)

2. Select the QoS option you want to use. Table 6-6 lists the traffic types (QoS classifiers) and the QoS options you can use for prioritizing or setting a policy on these traffic types:

6-6. Applying QoS Options to Traffic Types Defined by QoS Classifiers

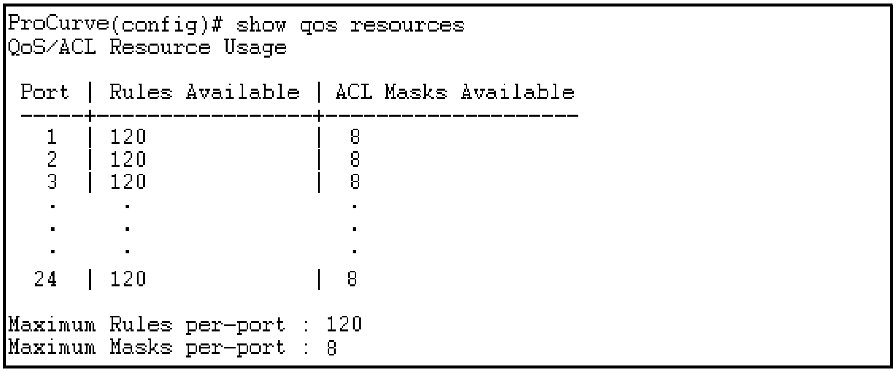
QoS Options for Prioritizing Outbound Traffic		**QoS Classifiers**						
		UDP/ TCP	**IP Device**	**IP-ToS Precedence**	**IP-DiffServ**	**L3 Protocol**	**VLAN -ID**	**Source -Port**
Option 1: Configure 802.1p Priority Rules Only	Prioritize traffic by sending specific packet types (determined by QoS classifier) to different outbound port queues on the switch. Rely on VLAN-tagged ports to carry packet priority as an 802.1p value to downstream devices.	Yes	Yes	Yes [1]	Yes	Yes[2]	Yes	Yes
Option 2: Configure ToS DSCP Policies with 802.1p Priorities	Prioritize traffic by sending specific packet types (determined by QoS classifier) to different outbound port queues on the switch. Propagate a service policy by reconfiguring the DSCP in outbound IP packets according to packet type. The packet is placed in an outbound port queue according to the 802.1p priority configured for that DSCP policy. (The policy assumes that downstream devices can be configured to recognize the DSCP in IP packets and implement the service policy it indicates.) Use VLAN-tagged ports to include packet priority as an 802.1p value to downstream devices.	Yes	Yes	No	Yes	No	Yes	Yes

[1] In this mode the configuration is fixed. You cannot change the automatic priority assignment when using IP-ToS Precedence as a QoS classifier.

3. If you want 802.1p priority settings to be included in outbound packets, ensure that tagged VLANs are configured on the appropriate downstream links.

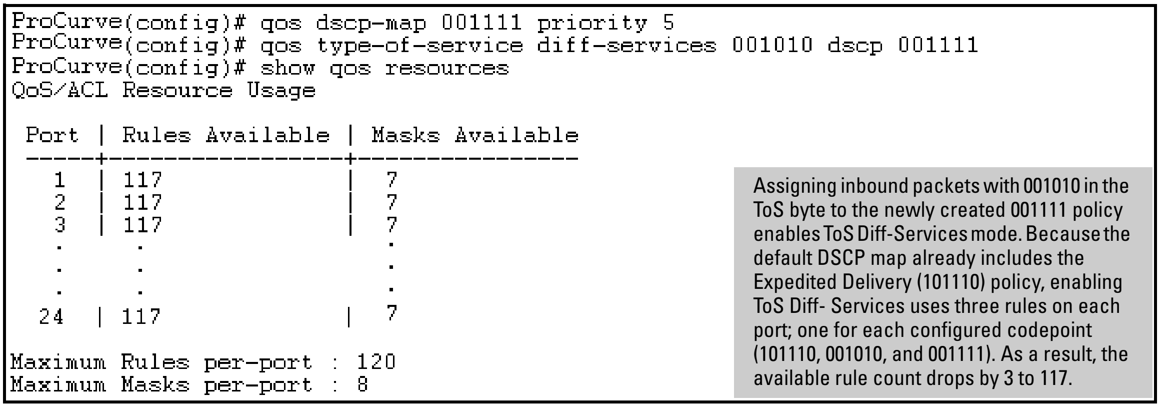
4. Determine the actual QoS configuration changes you will need to make on each QoS-capable device in your network in order to implement the desired policy. Also, if you want downstream devices to read and use DSCPs in IP packets from the switch, configure them to do so by enabling ToS Differentiated Service mode and making sure the same DSCP policies are configured.

Demonstrating How the Switch Uses Resources in DSCP Configurations. In the default configuration, the DSCP map is configured with one DSCP policy (Expedited Forwarding; 101110 with a "7" priority) but, because no ToS Diff-Services options are configured, no rules are used. If ToS Diff-Services mode is enabled, then one rule is immediately used for this codepoint. Adding a new DSCP policy (for example, 001111 with a "5" priority) and then configuring ToS Diff-Services to assign inbound packets with a codepoint of 001010 to the 001111 policy implements all policies configured in the DSCP map and, in this case, uses three rules; one for each codepoint invoked in the switch's current DSCP configuration (101110-the default, 001111, and 001010). Adding another Diff-Services assignment, such as assigning inbound packets with a codepoint of 000111 to the Expedited Forwarding policy (101110), would use one more rule on all ports.



```
ProCurve(config)# show qos resources
QoS/ACL Resource Usage

 Port | Rules Available | ACL Masks Available
 -----+-----------------+--------------------
   1  | 120             |  8
   2  | 120             |  8
   3  | 120             |  8
   .     .                 .
   .     .                 .
   .     .                 .
  24  | 120             |  8

Maximum Rules per-port : 120
Maximum Masks per-port : 8
```

Figure 6-3. Example of Rule Resources in the Default Configuration



```
ProCurve(config)# qos dscp-map 001111 priority 5
ProCurve(config)# qos type-of-service diff-services 001010 dscp 001111
ProCurve(config)# show qos resources
QoS/ACL Resource Usage

 Port | Rules Available | Masks Available
 -----+-----------------+----------------
   1  | 117             |  7
   2  | 117             |  7
   3  | 117             |  7
   .     .                 .
   .     .                 .
   .     .                 .
  24  | 117             |  7

Maximum Rules per-port : 120
Maximum Masks per-port : 8
```

Assigning inbound packets with 001010 in the ToS byte to the newly created 001111 policy enables ToS Diff-Services mode. Because the default DSCP map already includes the Expedited Delivery (101110) policy, enabling ToS Diff- Services uses three rules on each port; one for each configured codepoint (101110, 001010, and 001111). As a result, the available rule count drops by 3 to 117.

Figure 6-4. Example of Rule Usage When a Configuration Includes DSCP-Map and Type-of-Service Options

Using QoS Classifiers To Configure Quality of Service for Outbound Traffic

QoS Feature	Default	Menu	CLI	Web
UDP/TCP Priority	Disabled	—	page 6-16	Refer to Online Help.
IP-Device Priority	Disabled	—	page 6-23	"
IP Type-of-Service Priority	Disabled	—	page 6-29	"
VLAN-ID Priority	Disabled	—	page 6-43	"
Source-Port Priority	Disabled	—	page 6-49	"

Note

In addition to the information in this section on the various QoS classifiers, refer to "QoS Operating Notes and Restrictions" on page 6-63.

Viewing the QoS Configuration

All of these commands are available on the switches covered in this guide. Examples of the **show qos** output are included with the example for each priority type.

Syntax: show qos < *priority-classifier* >

tcp-udp-port-priority

Displays the current TCP/UDP port priority configuration. Refer to figure 6-9 on page 6-23.

device-priority

Displays the current device (IP address) priority configuration. Refer to figure 6-10 on page 6-25.

type-of-service

Displays the current type-of-service priority configuration. The display output differs according to the ToS option used:

- *IP Precedence: Refer to figure 6-14 on page 6-30.*
- *Diffserve: Refer to figure 6-16 on page 6-34.*

protocol-priority

Displays the current protocol priority configuration.

vlan-priority

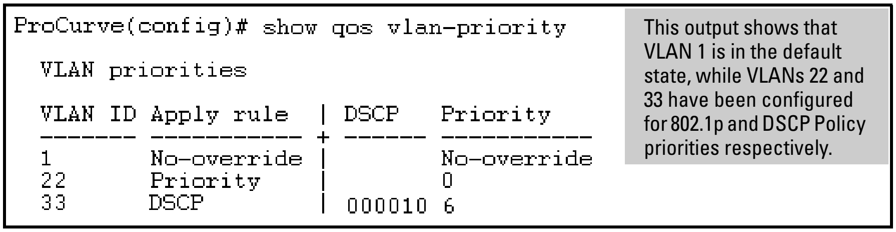
Displays the current VLAN priority configuration. Refer to figure 6-24 on page 6-45.

port-priority

Displays the current source-port priority configuration. Refer to figure 6-29 on page 6-50.

No Override

By default, the IP ToS, Protocol, VLAN-ID, and (source) port **show** outputs automatically list **No-override** for priority options that have not been configured. This means that if you do not configure a priority for a specific option, QoS does not prioritize packets to which that option applies, resulting in the **No override** state. In this case, IP packets received through a VLAN-tagged port receive whatever 802.1p priority they carry in the 802.1Q tag in the packet's header. VLAN-Tagged packets received through an untagged port are handled in the switch with "normal" priority. For example, figure 6-5 below shows a qos VLAN priority output in a switch where non-default priorities exist for VLANs 22 and 33, while VLAN 1 remains in the default configuration.



```
ProCurve(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       No-override |        No-override
  22      Priority    |        0
  33      DSCP        | 000010 6
```

This output shows that VLAN 1 is in the default state, while VLANs 22 and 33 have been configured for 802.1p and DSCP Policy priorities respectively.

Figure 6-5. Example of the Show QoS Output for VLAN Priority

QoS UDP/TCP Priority

QoS Classifier Precedence: 1

When you use UDP or TCP and a layer 4 Application port number as a QoS classifier, traffic carrying the specified UDP/TCP port number(s) is marked with the UDP/TCP classifier's configured priority level, without regard for any other QoS classifiers in the switch.

Note

UDP/TCP QoS applications are supported only for IPv4 packets only. For more information on packet-type restrictions, refer to “Details of Packet Criteria and Restrictions for QoS Support”, on page 6-63.

Options for Assigning Priority. Priority control options for TCP or UDP packets carrying a specified TCP or UDP port number include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets must be IPv4.)

For a given TCP or UDP port number, you can use only one of the above options at a time. However, for different port numbers, you can use different options.

TCP/UDP Port Number Ranges. There are three ranges:

- Well-Known Ports: 0 - 1023
- Registered Ports: 1024 - 49151
- Dynamic and/or Private Ports: 49152 - 65535

For more information, including a listing of UDP/TCP port numbers, go to the *Internet Assigned Numbers Authority* (IANA) website at:

www.iana.org

Then click on:

Protocol Number Assignment Services

P (Under “Directory of General Assigned Numbers” heading)

Port Numbers

Assigning an 802.1p Priority Based on TCP or UDP Port Number

This option assigns an 802.1p priority to (IPv4) TCP or UDP packets as described below.

Syntax: qos < udp-port | tcp-port > < *tcp or udp port number* > priority < 0 - 7 >

Configures an 802.1p priority for outbound packets having the specified TCP or UDP application port number. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: Disabled)

no qos < udp-port | tcp-port > < *tcp-udp port number* >

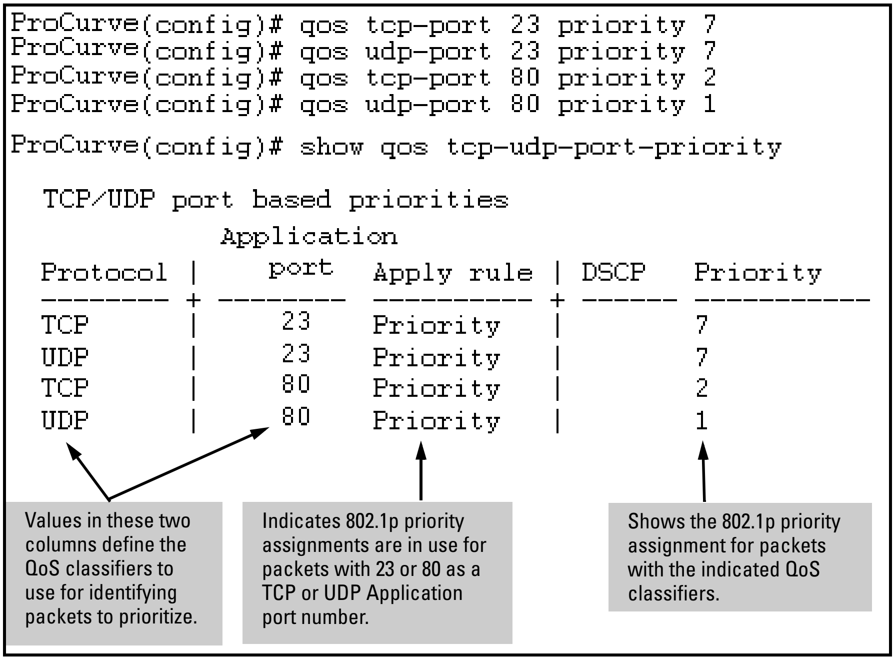
Deletes the specified UDP or TCP port number as a QoS classifier.

show qos tcp-udp-port-priority

Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.

For example, configure and list 802.1p priority for the following UDP and TCP port prioritization:

TCP/UDP Port	802.1p Priority for TCP	802.1p Priority for UDP
TCP Port 23 (Telnet)	7	7
UDP Port 23 (Telnet)	7	7
TCP Port 80 (World Wide Web HTTP)	2	2
UDP Port 80 (World Wide Web HTTP)	1	1



```
ProCurve(config)# qos tcp-port 23 priority 7
ProCurve(config)# qos udp-port 23 priority 7
ProCurve(config)# qos tcp-port 80 priority 2
ProCurve(config)# qos udp-port 80 priority 1

ProCurve(config)# show qos tcp-udp-port-priority

  TCP/UDP port based priorities
             Application
  Protocol |    port    Apply rule | DSCP   Priority
  -------- + --------  ---------- + ------ ----------
  TCP      |     23     Priority   |          7
  UDP      |     23     Priority   |          7
  TCP      |     80     Priority   |          2
  UDP      |     80     Priority   |          1
```

Values in these two columns define the QoS classifiers to use for identifying packets to prioritize.

Indicates 802.1p priority assignments are in use for packets with 23 or 80 as a TCP or UDP Application port number.

Shows the 802.1p priority assignment for packets with the indicated QoS classifiers.

Figure 6-6. Example of Configuring and Listing 802.1p Priority Assignments on TCP/UDP Ports

Assigning a DSCP Policy Based on TCP or UDP Port Number

Note The switches covered in this guide do not support DSCP policies on IPv4 packets with IP options. For more information on packet-type restrictions, refer to “Details of Packet Criteria and Restrictions for QoS Support”, on page 6-63.

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to (IPv4) TCP or UDP packets having the specified port number. That is, the switch:

1. Selects an incoming IP packet if the TCP or UDP port number it carries matches the port number specified in the TCP or UDP classifier (as shown in figure 6-6, above).
2. Overwrites (re-marks) the packet’s DSCP with the DSCP configured in the switch for such packets.

3. Assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)
4. Forwards the packet through the appropriate outbound port queue.

For more on DSCP, refer to "Terminology" on page 6-6.

Steps for Creating a DSCP Policy Based on TCP/UDP Port Number Classifiers. This procedure creates a DSCP policy for IPv4 packets carrying the selected UDP or TCP port-number classifier.

1. Identify the TCP or UDP port-number classifier you want to use for assigning a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected TCP or UDP port number.
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite (re-mark) the DSCP carried in packets received from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.
3. Configure the DSCP policy by using **qos dscp-map** to configure the priority to the codepoint you selected in step 2a. (For details, refer to the example later in this section, and to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)

Note

A codepoint must have an 802.1p priority assignment (0 - 7) before you can configure a policy for prioritizing packets by TCP or UDP port numbers. If a codepoint you want to use shows **No-override** in the **Priority** column of the DSCP map (**show qos dscp-map**), then you must assign a 0 - 7 priority before proceeding.

4. Configure the switch to assign the DSCP policy to packets with the specified TCP or UDP port number.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

This command is optional if a priority has already been assigned to the < ***codepoint*** *>. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this policy to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. For IPv4 packets, the DSCP will be replaced by the codepoint specified in this command. (Default:* **No-override** *for most codepoints. See table 6-9 on page 6-55.)*

Syntax: qos < udp-port | tcp-port > < *tcp or udp port number* > dscp < *codepoint* >

Assigns a DSCP policy to outbound packets having the specified TCP or UDP application port number and overwrites the DSCP in these packets with the assigned ***<codepoint>*** *value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. (The* ***<codepoint>*** *must be configured with an 802.1p setting. See step 3 on page 6-20.) If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default:* **No-override***)*

no qos < udp-port | tcp-port > < *tcp-udp port number* >

Deletes the specified UDP or TCP port number as a QoS classifier.

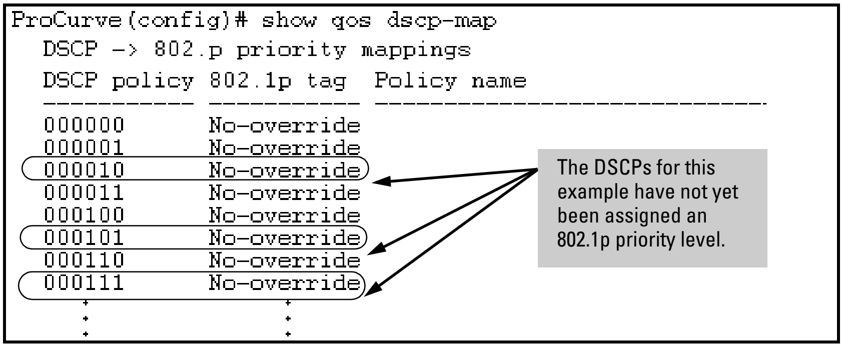
show qos tcp-udp-port-priority

Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.

For example, suppose you wanted to assign these DSCP policies to the packets identified by the indicated UDP and TDP port applications:

Port Applications	DSCP Policies	
	DSCP	Priority
23-UDP	000111	7
80-TCP	000101	5
914-TCP	000010	1
1001-UDP	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. (Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

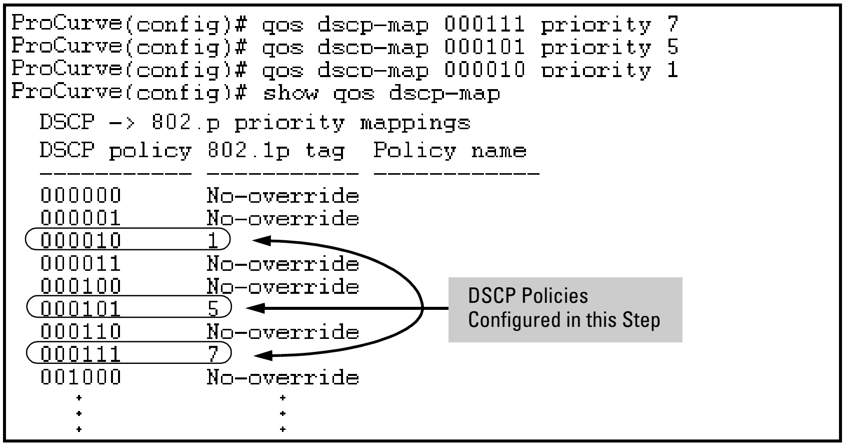


```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     .                .
     .                .
     .                .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 6-7. Display the Current DSCP-Map Configuration

2. Configure the DSCP policies for the codepoints you want to use.

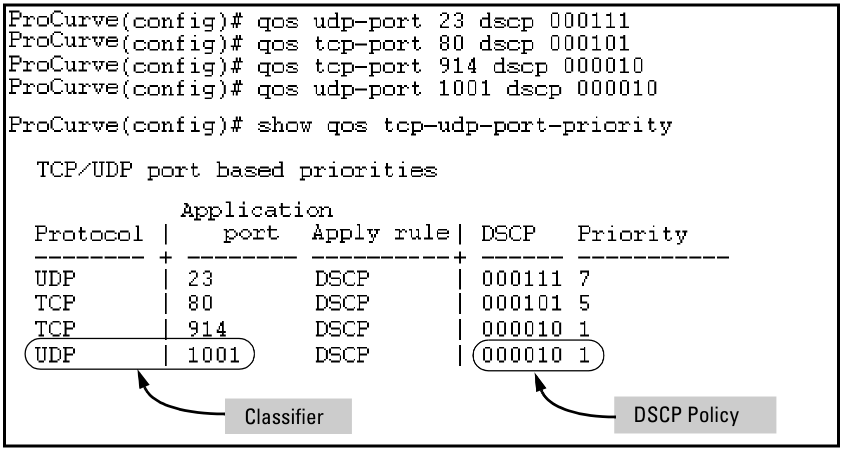


```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override
  000101      5
  000110      No-override
  000111      7
  001000      No-override
     .                .
     .                .
     .                .
```

DSCP Policies Configured in this Step

Figure 6-8. Assign Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected UDP/TCP port applications and display the result.



```
ProCurve(config)# qos udp-port 23 dscp 000111
ProCurve(config)# qos tcp-port 80 dscp 000101
ProCurve(config)# qos tcp-port 914 dscp 000010
ProCurve(config)# qos udp-port 1001 dscp 000010

ProCurve(config)# show qos tcp-udp-port-priority

  TCP/UDP port based priorities

             Application
  Protocol |    port   Apply rule | DSCP   Priority
  -------- + -------- ----------+ ------ -----------
  UDP      | 23        DSCP       | 000111 7
  TCP      | 80        DSCP       | 000101 5
  TCP      | 914       DSCP       | 000010 1
  UDP      | 1001      DSCP       | 000010 1
```

Figure 6-9. The Completed DSCP Policy Configuration for the Specified UDP/TCP Port Applications

The switch will now apply the DSCP policies in figure 6-9 to IPV4 packets received in the switch with the specified UDP/TCP port applications. This means the switch will:

- Overwrite the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assign the 802.1p priorities in the above policies to the selected packets.

QoS IP-Device Priority

QoS Classifier Precedence: 2

The IP device option, which applies only to IPv4 packets, enables you to use up to 256 IP addresses (source or destination) as QoS classifiers.

Where a particular device-IP address classifier has the highest precedence in the switch for traffic addressed to or from that device, then traffic received on the switch with that address is marked with the IP address classifier's configured priority level. Different IP device classifiers can have differing priority levels.

Note

The switch does not allow a QoS IP-device priority for the Management VLAN IP address, if configured. If there is no Management VLAN configured, then the switch does not allow configuring a QoS IP-device priority for the Default VLAN IP address.

Ip address QoS does not support layer-2 SAP incapsulation. For more information on packet-type restrictions, refer to table 6-10, "Details of Packet Criteria and Restrictions for QoS Support", on page 6-63.

Options for Assigning Priority. Priority control options for packets carrying a specified IP address include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an 802.1p priority; inbound packets must be IPv4.)

(For operation when other QoS classifiers apply to the same traffic, refer to "Classifiers for Prioritizing Outbound Packets" on page 6-10.)

For a given IP address, you can use only one of the above options at a time. However, for different IP addresses, you can use different options.

Assigning a Priority Based on IP Address

This option assigns an 802.1p priority to all IPv4 packets having the specified IP address as either a source or destination. (If both match, the priority for the IP destination address has precedence.)

Syntax: qos device-priority < *ip-address* > priority < 0 - 7 >

Configures an 802.1p priority for outbound packets having the specified IP address. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: Disabled)

no qos device-priority < *ip-address* >

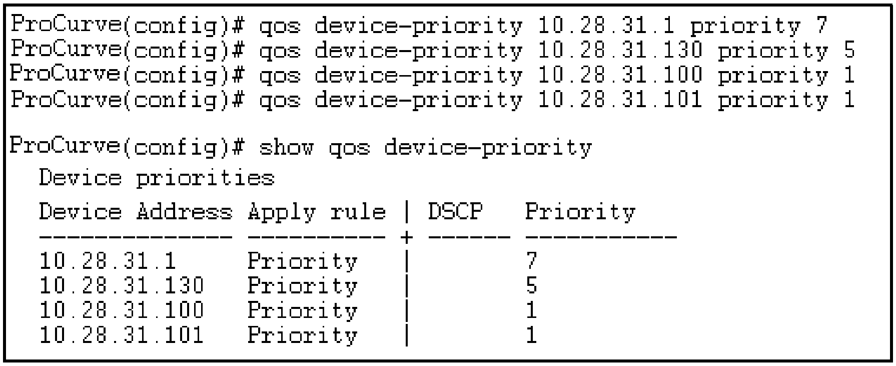
Removes the specified IP device-priority QoS classifier and resets the priority for that VLAN to **No-override**.

show qos device-priority

Displays a listing of all IP device-priority QoS classifiers currently in the running-config file.

For example, configure and list the 802.1p priority for packets carrying the following IP addresses:

IP Address	802.1p Priority
10.28.31.1	7
10.28.31.130	5
10.28.31.100	1
10.28.31.101	1



```
ProCurve(config)# qos device-priority 10.28.31.1 priority 7
ProCurve(config)# qos device-priority 10.28.31.130 priority 5
ProCurve(config)# qos device-priority 10.28.31.100 priority 1
ProCurve(config)# qos device-priority 10.28.31.101 priority 1

ProCurve(config)# show qos device-priority
  Device priorities
  Device Address Apply rule | DSCP   Priority
  -------------- ---------- + ------ -----------
  10.28.31.1     Priority   |        7
  10.28.31.130   Priority   |        5
  10.28.31.100   Priority   |        1
  10.28.31.101   Priority   |        1
```

Figure 6-10. Example of Configuring and Listing 802.1p Priority Assignments for Packets Carrying Specific IP Addresses

Assigning a DSCP Policy Based on IP Address

Note On the switches covered in this guide, DSCP policies cannot be applied to IPv4 packets having IP options. For more information on packet criteria and restrictions, refer to table 6-10 on page 6-63.

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets having the specified IP address (either source or destination). That is, the switch:

1. Selects an incoming IPv4 packet on the basis of the source or destination IP address it carries.
2. Overwrites the packet's DSCP with the DSCP configured in the switch for such packets, and assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)
3. Forwards the packet through the appropriate outbound port queue.

For more on DSCP, refer to “Terminology” on page 6-6.

Steps for Creating a Policy Based on IP Address. This procedure creates a DSCP policy for IPv4 packets carrying the selected IP address (source or destination).

1. Identify the IP address to use as a classifier for assigning a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected IP address:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.
3. Configure the DSCP policy by using **dscp-map** to configure the priority to the codepoint you selected in step 2a. (For details, refer to “Differentiated Services Codepoint (DSCP) Mapping” on page 6-54.)

Notes

A codepoint must have an 802.1p priority assignment (0 - 7) before you can configure a policy for prioritizing packets by IP address. If a codepoint you want to use shows **No-override** in the **Priority** column of the DSCP map (**show qos dscp-map**), then you must assign a 0 - 7 priority before proceeding.

On the switches covered in this guide, DSCP policies cannot be applied to IPv4 packets having IP options. For more information on packet criteria and restrictions, refer to 6-10 on page 6-63.

4. Configure the switch to assign the DSCP policy to packets with the specified IP address.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

This command is optional if a priority is already assigned to the < ***codepoint*** *>. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this policy to a packet, the priority determines the packet’s queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. If the packet is IPv4, the packet’s DSCP will be replaced by the codepoint specified in this command. (Default: For most codepoints,* **No-override**. *See figure 6-9 on page 6-55.)*

Syntax: qos device-priority < *ip-address* > dscp < *codepoint* >

Assigns a DSCP policy to packets carrying the specified IP address, and overwrites the DSCP in these packets with the assigned **< codepoint >** *value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default:* **No-override***)*

no qos device-priority < *ip-address* >

Deletes the specified IP address as a QoS classifier.

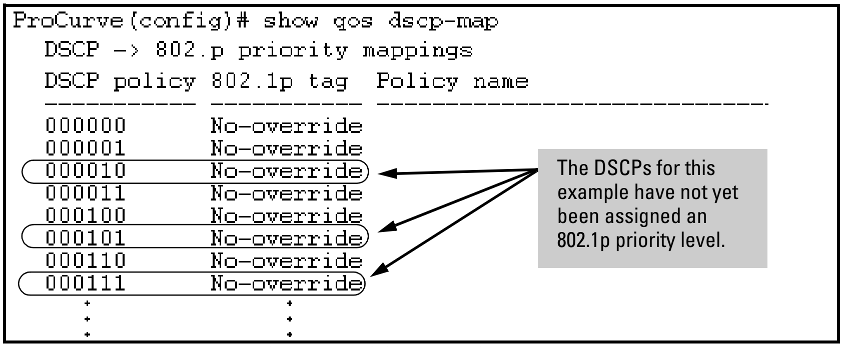
show qos device-priority

Displays a listing of all QoS Device Priority classifiers currently in the running-config file.

For example, suppose you wanted to assign these DSCP policies to the packets identified by the indicated IP addresses:

IP Address	DSCP Policies	
	DSCP	Priority
10.28.31.1	000111	7
10.28.31.130	000101	5
10.28.31.100	000010	1
10.28.31.101	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem if the configured priorities are acceptable for all applications using the same DSCP. (Refer to the “Note On Changing a Priority Setting” on page 6-57. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

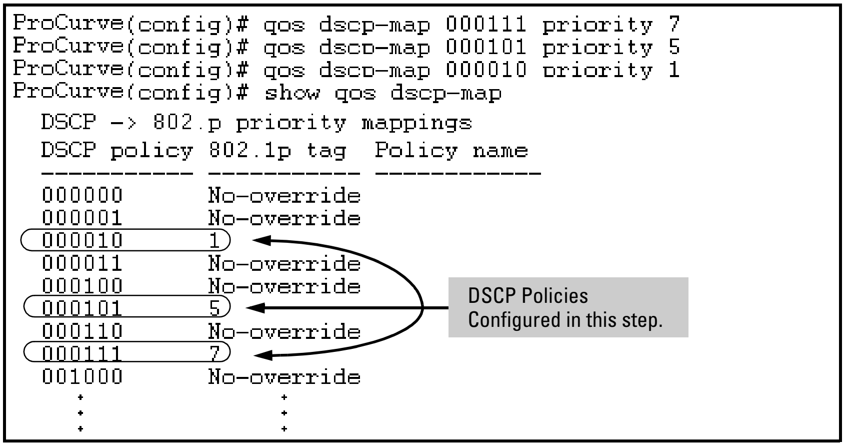


```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ----------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     .             .
     .             .
     .             .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 6-11. Display the Current DSCP-Map Configuration

2. Configure the priorities for the DSCPs you want to use.

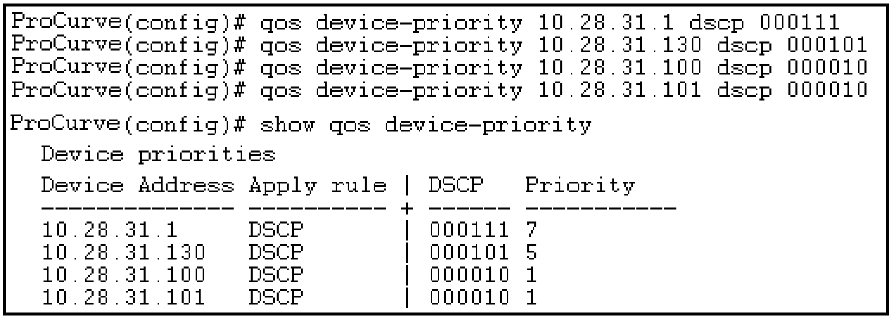


```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override
  000101      5
  000110      No-override
  000111      7
  001000      No-override
     .              .
     .              .
     .              .
```

DSCP Policies Configured in this step.

Figure 6-12. Assigning 802.1p Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected device IP addresses and display the result.



```
ProCurve(config)# qos device-priority 10.28.31.1 dscp 000111
ProCurve(config)# qos device-priority 10.28.31.130 dscp 000101
ProCurve(config)# qos device-priority 10.28.31.100 dscp 000010
ProCurve(config)# qos device-priority 10.28.31.101 dscp 000010

ProCurve(config)# show qos device-priority
  Device priorities
  Device Address Apply rule | DSCP   Priority
  -------------- ---------- + ------ -----------
  10.28.31.1     DSCP       | 000111 7
  10.28.31.130   DSCP       | 000101 5
  10.28.31.100   DSCP       | 000010 1
  10.28.31.101   DSCP       | 000010 1
```

Figure 6-13. The Completed Device-Priority/Codepoint Configuration

The switch will now apply the DSCP policies in figure 6-12 to IPv4 packets received on the switch with the specified IP addresses (source or destination). This means the switch will:

- Overwrite the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assign the 802.1p priorities in the above policies to the appropriate packets.

QoS IP Type-of-Service (ToS) Policy and Priority

QoS Classifier Precedence: 3

This feature applies only to IPv4 traffic and performs either of the following:

- **ToS IP-Precedence Mode:** All IP packets generated by upstream devices and applications include precedence bits in the ToS byte. Using this mode, the switch uses these bits to compute and assign the corresponding 802.1p priority.
- **ToS Differentiated Services (Diffserv) Mode:** This mode requires knowledge of the codepoints set in IP packets by the upstream devices and applications. It uses the ToS codepoint in IP packets coming from upstream devices and applications to assign 802.1p priorities to the packets. You can use this option to do both of the following:
  - **Assign a New Prioritization Policy:** A "policy" includes both a codepoint and a corresponding 802.1p priority. This option selects an incoming IPv4 packet on the basis of its codepoint and assigns a new codepoint and corresponding 802.1p priority. (Use the **qos dscp-map** command to specify a priority for any codepoint—page 6-54.)
  - **Assign an 802.1p Priority:** This option reads the DSCP of an incoming IPv4 packet and, without changing this codepoint, assigns the 802.1p priority to the packet, as configured in the DSCP Policy Table (page 6-54). This means that a priority value of 0 - 7 must be configured for a DSCP before the switch will attempt to perform a QoS match on the packet's DSCP bits.

  Before configuring the ToS Diffserv mode, you must use the **dscp-map** command to configure the desired 802.1p priorities for the codepoints you want to use for either option. This command is illustrated in the following examples and is described under "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.

Unless IP-Precedence mode and Diffserv mode are both disabled (the default setting), enabling one automatically disables the other. *For more on ToS operation, refer to "Details of QoS IP Type-of-Service" on page 6-38.*

Assigning an 802.1p Priority to IPv4 Packets on the Basis of the ToS Precedence Bits

If a device or application upstream of the switch sets the precedence bits in the ToS byte of IPv4 packets, you can use this feature to apply that setting for prioritizing packets for outbound port queues. If the outbound packets are in a tagged VLAN, this priority is carried as an 802.1p value to the adjacent downstream devices.

Syntax: qos type-of-service ip-precedence

Causes the switch to automatically assign an 802.1p priority to all IPv4 packets by computing each packet's 802.1p priority from the precedence bits the packet carries. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (ToS IP Precedence Default: Disabled)

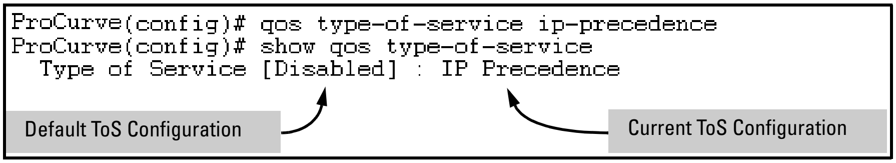
no qos type-of-service

Disables all ToS classifier operation, including prioritization using the precedence bits.

show qos type-of-service

When ip-precedence is enabled (or if neither ToS option is configured), shows the ToS configuration status. If diff-services is enabled, lists codepoint data as described under "Assigning a DSCP Policy on the Basis of the DSCP in IPv4 Packets Received from Upstream Devices" on page 6-35.

With this option, prioritization of outbound packets relies on the IP-Precedence bit setting that IP packets carry with them from upstream devices and applications. To configure and verify this option:



```
ProCurve(config)# qos type-of-service ip-precedence
ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : IP Precedence
```

Figure 6-14. Example of Enabling ToS IP-Precedence Prioritization

To replace this option with the ToS diff-services option, just configure **diff-services** as described below, which automatically disables IP-Precedence. To disable IP-Precedence without enabling the diff-services option, use this command:

```
ProCurve(config)# no qos type-of-service
```

Assigning an 802.1p Priority to IPv4 Packets on the Basis of Incoming DSCP

One of the best uses for this option is on an interior switch where you want to honor (continue) a policy set on an edge switch. That is, it enables you to select incoming packets having a specific DSCP and forward these packets with the desired 802.1p priority. For example, if an edge switch "A" marks all packets received on port A5 with a particular DSCP, you can configure a downstream (interior) switch "B" to handle such packets with the desired priority (regardless of whether 802.1Q tagged VLANs are in use).

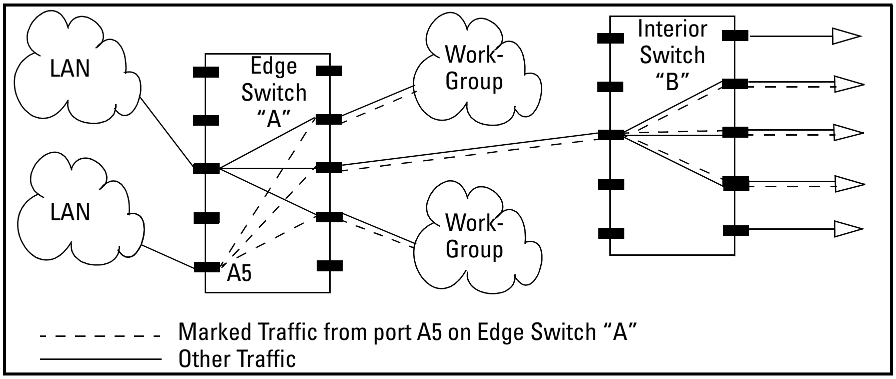


Figure 6-15. Interior Switch "B" Honors the Policy Established in Edge Switch "A"

To do so, assign the desired 802.1p priority to the same codepoint that the upstream or edge switch assigns to the selected packets. When the downstream switch receives an IPv4 packet carrying one of these codepoints, it assigns the configured priority to the packet and sends it out the appropriate priority queue. (The packet retains the codepoint it received from the upstream or edge switch). You can use this option concurrently with the diffserv DSCP Policy option (described later in this section), as long as the DSCPs specified in the two options do not match.

Operating Notes

Different applications may use the same DSCP in their IP packets. Also, the same application may use multiple DSCPs if the application originates on different clients, servers, or other devices. Using an edge switch enables you to select the packets you want and mark them with predictable DSCPs that can be used by downstream switches to honor policies set in the edge switch.

When enabled, the switch applies direct 802.1p prioritization to all packets having codepoints that meet these criteria:

- The codepoint is configured with an 802.1p priority in the DSCP table. (Codepoints configured with **No-override** are not used.)
- The codepoint is not configured for a new DSCP policy assignment.

Thus, the switch does not allow the same incoming codepoint (DSCP) to be used simultaneously for directly assigning an 802.1p priority and also assigning a DSCP policy. For a given incoming codepoint, if you configure one option and then the other, the second overwrites the first.

To use this option:

1. Identify a DSCP used to set a policy in packets received from an upstream or edge switch.
2. Determine the 802.1p priority (0 - 7) you want to apply to packets carrying the identified DSCP. (You can either maintain the priority assigned in the upstream or edge switch, or assign a new priority.)
3. Use **qos dscp-map <** ***codepoint*** **> priority < 0 - 7 >** to assign the 802.1p priority you want to the specified DSCP. (For more on this topic, refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)
4. Enable **diff-services**

.

Syntax: qos type-of-service diff-services < *codepoint*>

Causes the switch to read the **< *codepoint*>** *(DSCP) of an incoming IPv4 packet and, when a match occurs, assign a corresponding 802.1p priority, as configured in the switch's DSCP table (page 6-55).*

no qos type-of-service

Disables all ToS classifier operation.

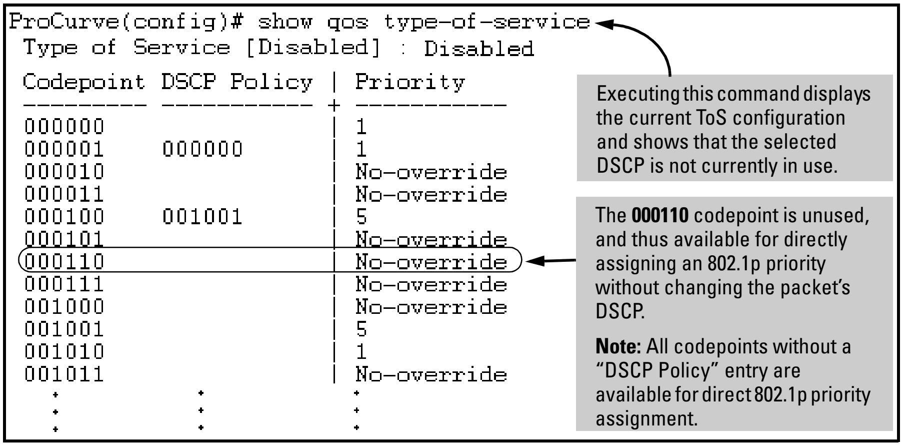
no qos dscp-map < *codepoint*>

Disables direct 802.1p priority assignment to packets carrying the < ***codepoint***> *by reconfiguring the codepoint priority assignment in the DSCP table to* **No-override**. *Note that if this codepoint is in use as a DSCP policy for another diffserv codepoint, you must disable or redirect the other diffserv codepoint's DSCP policy before you can disable or change the codepoint. For example, in figure 6-16 you cannot change the priority for the 000000 codepoint until you redirect the DSCP policy for 000001 away from using 000000 as a policy. (Refer to "Note On Changing a Priority Setting" on page 6-57. Refer also to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)*

show qos type-of-service

Displays current Type-of-Service configuration. In diffserv mode it also shows the current direct 802.1p assignments and the current DSCP assignments covered later in this section.

For example, an edge switch "A" in an untagged VLAN assigns a DSCP of 000110 on IP packets it receives on port A6, and handles the packets with high priority (7). When these packets reach interior switch "B" you want the switch to handle them with the same high priority. To enable this operation you would configure an 802.1p priority of 7 for packets received with a DSCP of **000110**, and then enable **diff-services**:



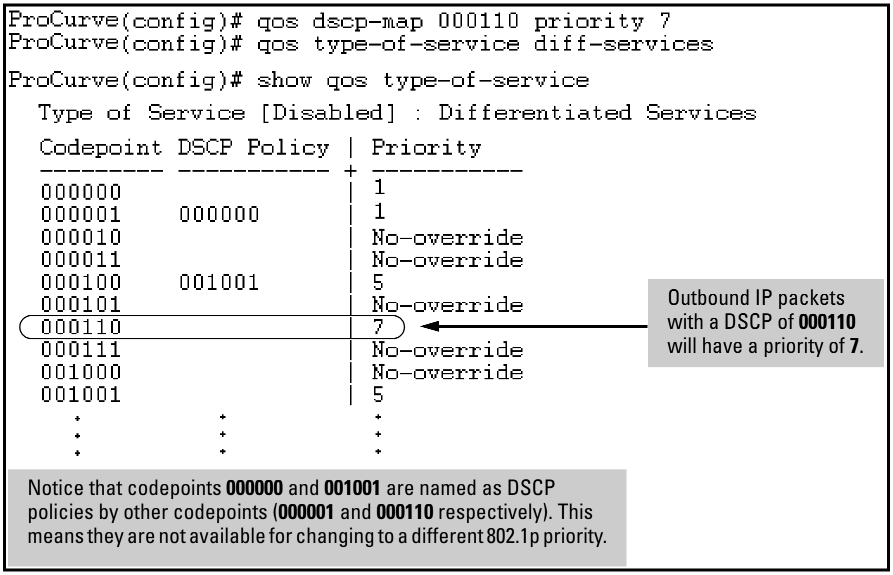
```
ProCurve(config)# show qos type-of-service
 Type of Service [Disabled] : Disabled

 Codepoint DSCP Policy | Priority
 --------- ----------- + -----------
 000000                | 1
 000001    000000      | 1
 000010                | No-override
 000011                | No-override
 000100    001001      | 5
 000101                | No-override
 000110                | No-override
 000111                | No-override
 001000                | No-override
 001001                | 5
 001010                | 1
 001011                | No-override
   .          .          .
   .          .          .
   .          .          .
```

Executing this command displays the current ToS configuration and shows that the selected DSCP is not currently in use.

The **000110** codepoint is unused, and thus available for directly assigning an 802.1p priority without changing the packet's DSCP.

Note: All codepoints without a "DSCP Policy" entry are available for direct 802.1p priority assignment.

Figure 6-16. Example Showing Codepoints Available for Direct 802.1p Priority Assignments



```
ProCurve(config)# qos dscp-map 000110 priority 7
ProCurve(config)# qos type-of-service diff-services

ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : Differentiated Services

  Codepoint DSCP Policy | Priority
  --------- ----------- + -----------
  000000                | 1
  000001    000000      | 1
  000010                | No-override
  000011                | No-override
  000100    001001      | 5
  000101                | No-override
  000110                | 7
  000111                | No-override
  001000                | No-override
  001001                | 5
    .          .          .
    .          .          .
    .          .          .
```

Outbound IP packets with a DSCP of **000110** will have a priority of **7**.

Notice that codepoints **000000** and **001001** are named as DSCP policies by other codepoints (**000001** and **000110** respectively). This means they are not available for changing to a different 802.1p priority.

Figure 6-17. Example of a Type-of-Service Configuration Enabling Both Direct 802.1p Priority Assignment and DSCP Policy Assignment

Assigning a DSCP Policy on the Basis of the DSCP in IPv4 Packets Received from Upstream Devices

The preceding section describes how to forward a policy set by an edge (or upstream) switch. This option changes a DSCP policy in an IPv4 packet by changing its IP ToS codepoint and applying the priority associated with the new codepoint. (A DSCP policy consists of a differentiated services codepoint and an associated 802.1p priority.) You can use this option concurrently with the diffserv 802.1p priority option (above), as long as the DSCPs specified in the two options do not match.

To use this option to configure a change in policy:

1. Identify a DSCP used to set a policy in packets received from an upstream or edge switch.
2. Create a new policy by using **qos dscp-map <** ***codepoint*** **> priority < 0 - 7 >** to configure an 802.1p priority for the codepoint you will use to overwrite the DSCP the packet carries from upstream. (For more on this topic, refer to “Differentiated Services Codepoint (DSCP) Mapping” on page 6-54.)
3. Use **qos type-of-service diff-services <** ***incoming-DSCP*** **> dscp <** ***outgoing-DSCP*** **>** to change the policy on packets coming from the edge or upstream switch with the specified incoming DSCP.

   (Figure 6-15 on page 6-31 illustrates this scenario.)

Note

On the switches covered in this guide, DSCP policies (codepoint re-marking) cannot be applied to outbound IPv4 packets having IP options. (The 802.1p priority in the VLAN tag is applied.) For more information on packet criteria and restrictions, refer to 6-10 on page 6-63.

.

Syntax: qos type-of-service diff-services

Enables ToS diff-services.

Syntax: qos type-of-service diff-services < *current-codepoint* > dscp < *new-codepoint* >

*Configures the switch to select an incoming IP packet carrying the <***current-codepoint***> and then use the <***new-codepoint***> to assign a new, previously configured DSCP policy to the packet. The policy overwrites the <***current-codepoint***> with the < ***new-codepoint*** > and assigns the 802.1p priority specified by the policy. (Use the* **qos dscp-map** *command to define the priority for the DSCPs—page 6-54.)*

Syntax: no qos type-of-service

Disables all ToS classifier operation. Current ToS DSCP policies and priorities remain in the configuration and will become available if you re-enable ToS diff-services.

Syntax: no qos type-of-service [diff-services < *codepoint* >]

*Deletes the DSCP policy assigned to the < ***codepoint*** > and returns the < ***codepoint*** > to the 802.1p priority setting it had before the DSCP policy was assigned. (This will be either a value from 0 - 7 or* **No-override***.)*

Syntax: show qos type-of-service

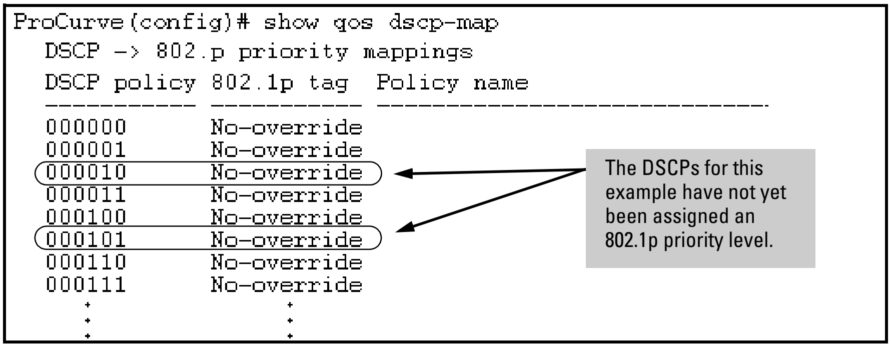
Displays a listing of codepoints, with any corresponding DSCP policy re-assignments for outbound packets. Also lists the (802.1p) priority for each codepoint that does not have a DSCP policy assigned to it.

For example, suppose you want to configure the following two DSCP policies for packets received with the indicated DSCPs.

Received DSCP	Policy DSCP	802.1p Priority	Policy Name (Optional)
001100	000010	6	Level 6
001101	000101	4	Level 4

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem as long as the configured priorities are acceptable for all applications using the

same DSCP. (Refer to the “Note On Changing a Priority Setting” on page 6-57. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

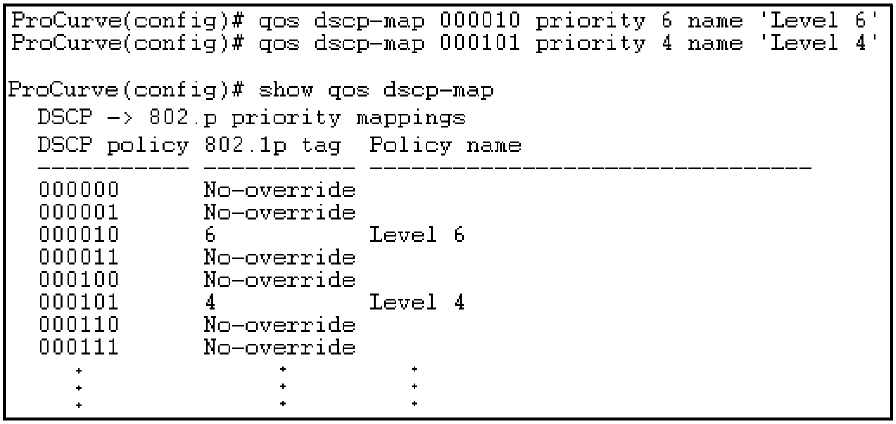


```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     .              .
     .              .
     .              .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 6-18. Display the Current DSCP-Map Configuration

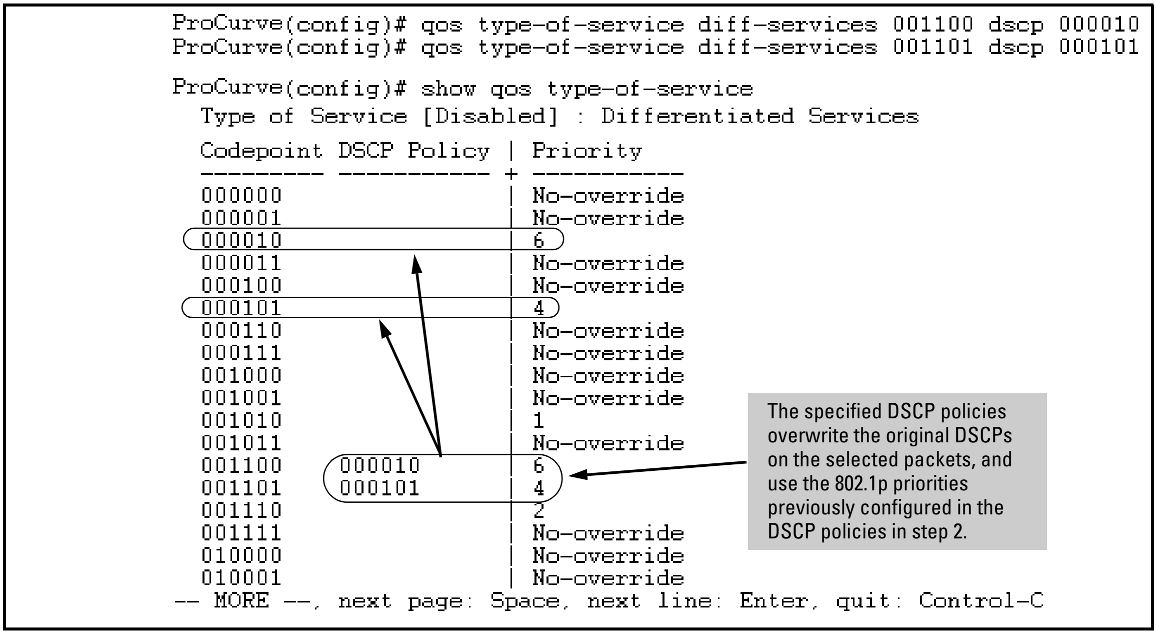
2. Configure the policies in the DSCP table:



```
ProCurve(config)# qos dscp-map 000010 priority 6 name 'Level 6'
ProCurve(config)# qos dscp-map 000101 priority 4 name 'Level 4'

ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- --------------------------------
  000000      No-override
  000001      No-override
  000010      6           Level 6
  000011      No-override
  000100      No-override
  000101      4           Level 4
  000110      No-override
  000111      No-override
     .              .         .
     .              .         .
     .              .         .
```

Figure 6-19. Example of Policies Configured (with Optional Names) in the DSCP Table

3. Assign the policies to the codepoints in the selected packet types.



```
ProCurve(config)# qos type-of-service diff-services 001100 dscp 000010
ProCurve(config)# qos type-of-service diff-services 001101 dscp 000101

ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : Differentiated Services

  Codepoint DSCP Policy | Priority
  --------- ----------- + -----------
  000000                | No-override
  000001                | No-override
  000010                | 6
  000011                | No-override
  000100                | No-override
  000101                | 4
  000110                | No-override
  000111                | No-override
  001000                | No-override
  001001                | No-override
  001010                | 1
  001011                | No-override
  001100    000010      | 6
  001101    000101      | 4
  001110                | 2
  001111                | No-override
  010000                | No-override
  010001                | No-override
-- MORE --, next page: Space, next line: Enter, quit: Control-C
```

The specified DSCP policies overwrite the original DSCPs on the selected packets, and use the 802.1p priorities previously configured in the DSCP policies in step 2.

Figure 6-20. Example of Policy Assignment to Outbound Packets on the Basis of the DSCP in the Packets Received from Upstream Devices

Details of QoS IP Type-of-Service

IP packets include a Type of Service (ToS) byte. The ToS byte includes:

- **A Differentiated Services Codepoint (DSCP):** This element is comprised of the upper six bits of the ToS byte). There are 64 possible codepoints.
  - In the switches covered in this guide, the default **qos** configuration includes some codepoints with 802.1p priority settings for Assured-Forwarding and Expedited Forwarding (codepoint 101110), while others are unused (and listed with **No-override** for a Priority).

Refer to figure 6-9 on page 6-55 for an illustration of the default DSCP policy table.

Using the **qos dscp map** command, you can configure the switch to assign different prioritization policies to IPv4 packets having different codepoints. As an alternative, you can configure the switch to assign a new codepoint to an IPv4 packet, along with a corresponding 802.1p priority (0-7). To use this option in the simplest case, you would:

a. Configure a specific DSCP with a specific priority in an edge switch.

b. Configure the switch to mark a specific type of inbound traffic with that DSCP (and thus create a policy for that traffic type).

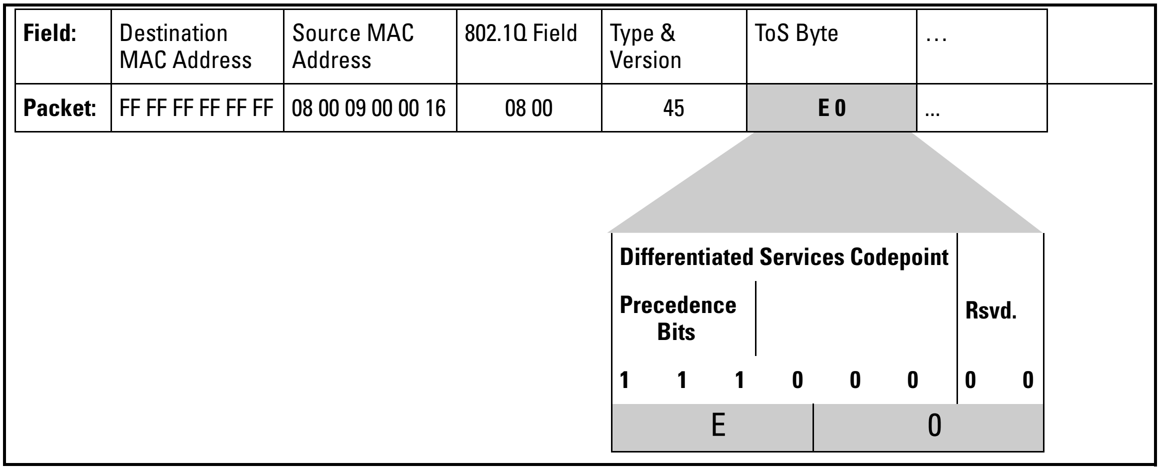
c. Configure the internal switches in your LAN to honor the policy.

(For example, you could configure an edge switch to assign a codepoint of 000001 to all packets received from a specific VLAN, and then handle all traffic with that codepoint at high priority.)

For a codepoint listing and the commands for displaying and changing the DSCP Policy table, refer to “Differentiated Services Codepoint (DSCP) Mapping” on page 6-54.

- **Precedence Bits:** This element is a subset of the DSCP and is comprised of the upper three bits of the ToS byte. When configured to do so, the switch uses the precedence bits to determine a priority for handling the associated packet. (The switch does not change the setting of the precedence bits.) Using the ToS Precedence bits to prioritize IPv4 packets relies on priorities set in upstream devices and applications.

Figure 6-21 shows an example of the ToS byte in the header for an IPv4 packet, and illustrates the diffserv bits and precedence bits in the ToS byte. (Note that the Precedence bits are a subset of the Differentiated Services bits.)



Field:	Destination MAC Address	Source MAC Address	802.1Q Field	Type & Version	ToS Byte	. . .
Packet:	FF FF FF FF FF FF	08 00 09 00 00 16	08 00	45	**E 0**	...

Figure 6-21. The ToS Codepoint and Precedence Bits

6-7. How the Switch Uses the ToS Configuration

Outbound Port	ToS Option:	
	802.1p (Value = 0 - 7)	**Differentiated Services**
IP Packet Sent Out an Untagged Port in a VLAN	Depending on the value of the IP Precedence bits in the packet's ToS field, the packet will go to one of eight outbound port queues in the switch: 1 - 2 = low priority (queue 1, 2) 0 - 3 = normal priority (queue 3, 4) 4 - 5 = medium priority (queue 5, 6) 6 - 7 = high priority (queue 7, 8)	For a given packet carrying a ToS codepoint that the switch has been configured to detect: • Change the codepoint according to the configured policy and assign the 802.1p priority specified for the new codepoint in the DSCP Policy Table (page 6-54). • Do not change the codepoint, but assign the 802.1p priority specified for the existing codepoint in the DSCP Policy Table (page 6-54). Depending on the 802.1p priority used, the packet will leave the switch through one of the following queues: 1 - 2 = low priority (queue 1, 2) 0 - 3 = normal priority (queue 3, 4) 4 - 5 = medium priority (queue 5, 6) 6 - 7 = high priority (queue 7, 8) If **No-override** (the default) has been configured for a specified codepoint, then the packet is not prioritized by ToS and, by default, is sent to the "normal priority" queue.
IP Packet Sent Out an Untagged Port in a VLAN	Same as above, plus the IP Precedence value (0 - 7) will be used to set a corresponding 802.1p priority in the VLAN tag carried by the packet to the next downstream device. Refer to table 6-8, below.	Same as above, plus the Priority value (0 - 7) will be used to set a corresponding 802.1p priority in the VLAN tag carried by the packet to the next downstream device. Where **No-override** is the assigned priority, the VLAN tag carries a "0" (normal priority) 802.1p setting if not prioritized by other QoS classifiers.

6-8. ToS IP-Precedence Bit Mappings to 802.1p Priorities

ToS Byte IP Precedence Bits	Corresponding 802.1p Priority	Service Priority Level
000	1	Lowest
001	2	Low
002	0	Normal
003	3	
004	4	
005	5	
006	6	
007	7	Highest

Assigning a Priority Based on Layer-3 Protocol

When QoS on the switch is configured with a Layer-3 protocol as the highest-precedence classifier and the switch receives traffic carrying that protocol, then this traffic is assigned the priority configured for this classifier. (For operation when other QoS classifiers apply to the same traffic, refer to "Classifiers for Prioritizing Outbound Packets" on page 6-10.)

Syntax: qos protocol

< ip | ipx | arp | appletalk | sna | netbeui > priority < 0 - 7 >

Configures an 802.1p priority for outbound packets having the specified protocol. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each protocol type. (Default: **No-override***)*

no qos protocol

< ip | ipx | arp | appletalk | sna | netbeui >

Disables use of the specified protocol as a QoS classifier and resets the protocol priority to **No-override**.

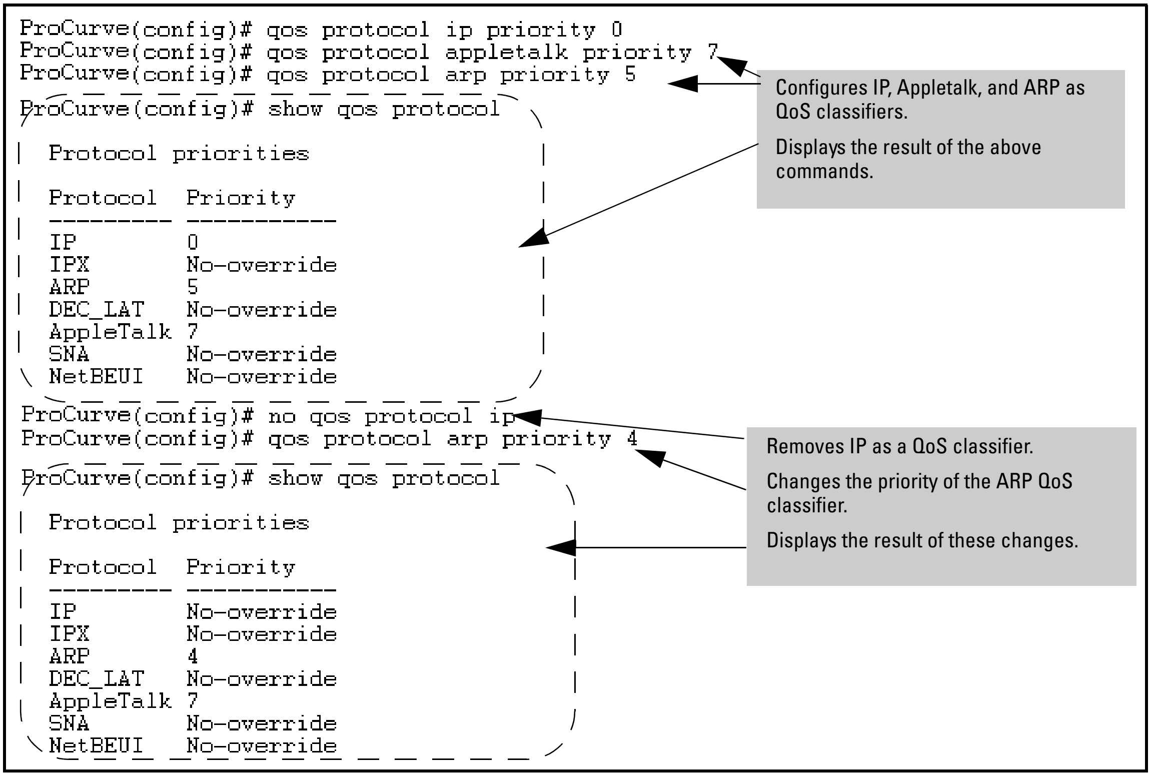
show qos protocol

Lists the QoS protocol classifiers with their priority settings.

For example:

1. Configure QoS protocol classifiers with IP at 0 (normal), ARP at 5 (medium), and AppleTalk at 7 (high) and display the QoS protocol configuration.
2. Disable the QoS IP protocol classifier, downgrade the ARP priority to 4, and again display the QoS protocol configuration.

Figure 6-22 shows the command sequence and displays for the above steps.



```
ProCurve(config)# qos protocol ip priority 0
ProCurve(config)# qos protocol appletalk priority 7
ProCurve(config)# qos protocol arp priority 5
ProCurve(config)# show qos protocol

  Protocol priorities

  Protocol  Priority
  --------- -----------
  IP        0
  IPX       No-override
  ARP       5
  DEC_LAT   No-override
  AppleTalk 7
  SNA       No-override
  NetBEUI   No-override
ProCurve(config)# no qos protocol ip
ProCurve(config)# qos protocol arp priority 4
ProCurve(config)# show qos protocol

  Protocol priorities

  Protocol  Priority
  --------- -----------
  IP        No-override
  IPX       No-override
  ARP       4
  DEC_LAT   No-override
  AppleTalk 7
  SNA       No-override
  NetBEUI   No-override
```

Configures IP, Appletalk, and ARP as QoS classifiers.

Displays the result of the above commands.

Removes IP as a QoS classifier.

Changes the priority of the ARP QoS classifier.

Displays the result of these changes.

Figure 6-22. Adding, Displaying, Removing, and Changing QoS Protocol Classifiers

QoS VLAN-ID (VID) Priority

QoS Classifier Precedence: 5

The QoS protocol option enables you to use up to 256 VIDs as QoS classifiers. Where a particular VLAN-ID classifier has the highest precedence in the switch for traffic in that VLAN, then traffic received in that VLAN is marked with the VID classifier's configured priority level. Different VLAN-ID classifiers can have differing priority levels.

Options for Assigning Priority. Priority control options for packets carrying a specified VLAN-ID include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets must be IPv4.)

(For operation when other QoS classifiers apply to the same traffic, refer to "Classifiers for Prioritizing Outbound Packets" on page 6-10.)

Note

QoS with VID priority applies to static VLANs only, and applying QoS to dynamic VLANs created by GVRP operation is not supported. A VLAN must exist while a subject of a QoS configuration, and eliminating a VLAN from the switch causes the switch to clear any QoS features configured for that VID.

Assigning a Priority Based on VLAN-ID

This option assigns a priority to all outbound packets having the specified VLAN-ID (VID). You can configure this option by either specifying the VLAN-ID ahead of the **qos** command or moving to the VLAN context for the VLAN you want to configure for priority.

Syntax: vlan < *vid* > qos priority < 0 - 7 >

Configures an 802.1p priority for outbound packets belonging to the specified VLAN. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each VLAN-ID. (Default: **No-override***)*

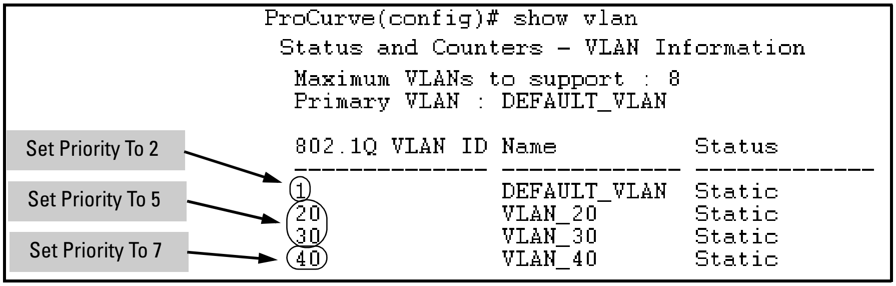
Syntax: no vlan < *vid* > qos

Removes the specified VLAN-ID as a QoS classifier and resets the priority for that VLAN to **No- override**.

Syntax: show qos vlan-priority

Displays a listing of the QoS VLAN-ID classifiers currently in the running-config file, with their priority data.

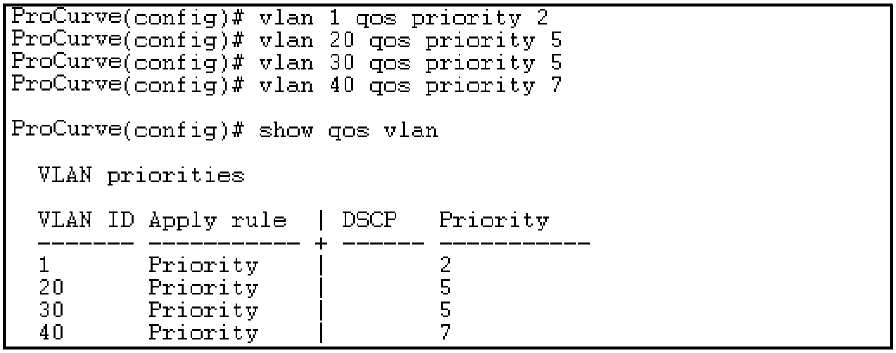
1. For example, suppose that you have the following VLANs configured on the switch and want to prioritize them as shown:



```
ProCurve(config)# show vlan
 Status and Counters - VLAN Information
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN

  802.1Q VLAN ID Name          Status
  -------------- ------------- -------------
  1              DEFAULT_VLAN  Static
  20             VLAN_20       Static
  30             VLAN_30       Static
  40             VLAN_40       Static
```

Figure 6-23. Example of a List of VLANs Available for QoS Prioritization

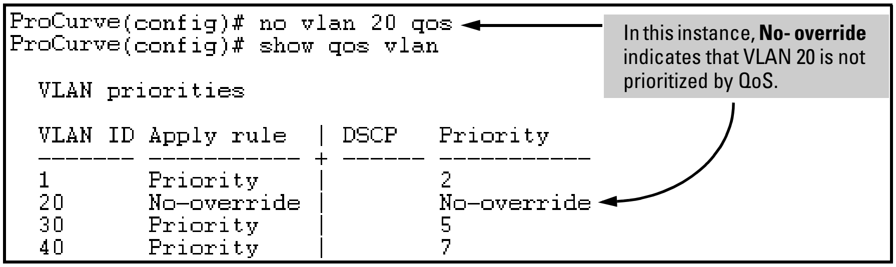
2. You would then execute the following commands to prioritize the VLANs by VID:



```
ProCurve(config)# vlan 1 qos priority 2
ProCurve(config)# vlan 20 qos priority 5
ProCurve(config)# vlan 30 qos priority 5
ProCurve(config)# vlan 40 qos priority 7

ProCurve(config)# show qos vlan

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       Priority    |        2
  20      Priority    |        5
  30      Priority    |        5
  40      Priority    |        7
```

Figure 6-24. Configuring and Displaying QoS Priorities on VLANs

If you then decided to remove VLAN_20 from QoS prioritization:



```
ProCurve(config)# no vlan 20 qos
ProCurve(config)# show qos vlan

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       Priority    |        2
  20      No-override |        No-override
  30      Priority    |        5
  40      Priority    |        7
```

In this instance, **No- override** indicates that VLAN 20 is not prioritized by QoS.

Figure 6-25. Returning a QoS-Prioritized VLAN to "No-override" Status

Assigning a DSCP Policy Based on VLAN-ID (VID)

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets having the specified VLAN-ID (VID). That is, the switch:

1. Selects an incoming IP packet on the basis of the VLAN-ID it carries.
2. Overwrites the packet's DSCP with the DSCP configured in the switch for such packets.
3. Assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)
4. Forwards the packet through the appropriate outbound port queue.

For more on DSCP, refer to "Terminology" on page 6-6.

Steps for Creating a Policy Based on VLAN-ID Classifier.

1. Determine the VLAN-ID classifier to which you want to assign a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected VLAN-ID:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.
3. Configure the DSCP policy by using **qos dscp-map** to configure the priority for each codepoint. (For details, see the example later in this section, and to “Differentiated Services Codepoint (DSCP) Mapping” on page 6-54.)

Note

A codepoint must have an 802.1p priority (0 - 7) before you can configure the codepoint for use in prioritizing packets by VLAN-ID. If a codepoint you want to use shows **No-override** in the **Priority** column of the DSCP Policy table (**show qos dscp-map**), then assign a priority before proceeding.

4. Configure the switch to assign the DSCP policy to packets with the specified VLAN-ID.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

This command is optional if a priority has already been assigned to the < ***codepoint*** *>. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this priority to a packet, the priority determines the packet’s queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. If the packet is IPv4, the packet’s DSCP will be replaced by the codepoint specified in this command. (Default: For most codepoints,* **No-override**. *See figure 6-9 on page 6-55 on page 6-55.)*

Syntax: vlan < *vid* > qos dscp < *codepoint* >

Assigns a DSCP policy to packets carrying the specified VLAN-ID, and overwrites the DSCP in these packets with the assigned < **codepoint** *> value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default:* **No-override***)*

Syntax: no vlan < *vid* > qos

Removes QoS classifier for the specified VLAN.

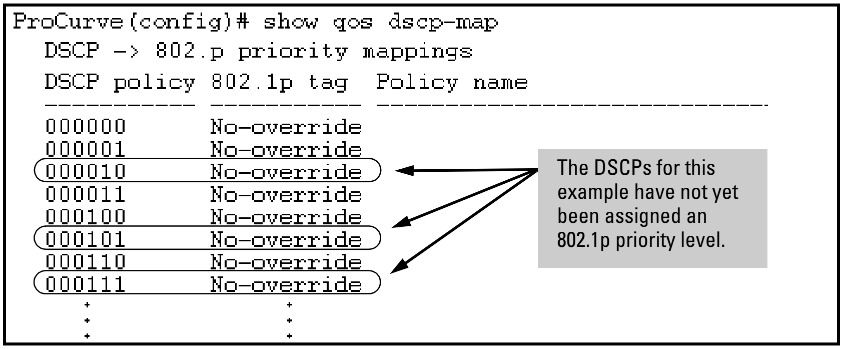
Syntax: show qos device-priority

Displays a listing of all QoS VLAN-ID classifiers currently in the running-config file.

For example, suppose you wanted to assign this set of priorities:

VLAN-ID	DSCP	Priority
40	000111	7
30	000101	5
20	000010	1
1	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP. (Refer to the "Note On Changing a Priority Setting" on page 6-57. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

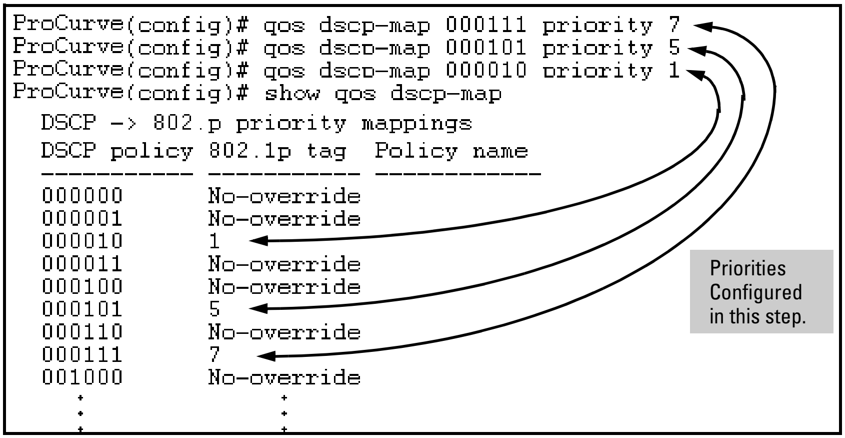


```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     :             :
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 6-26. Display the Current Configuration in the DSCP Policy Table

2. Configure the priorities for the DSCPs you want to use.

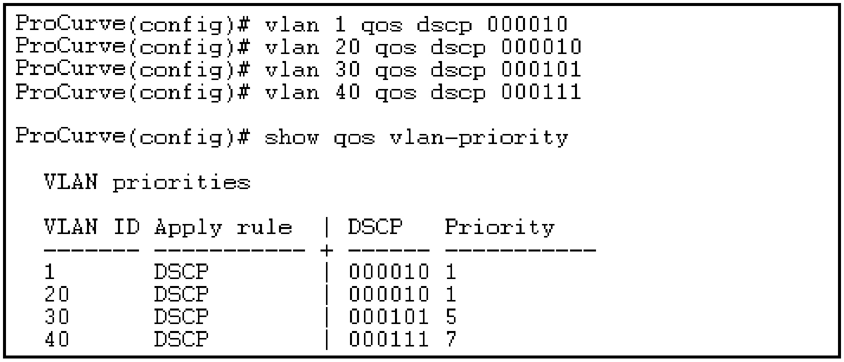


```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override
  000101      5
  000110      No-override
  000111      7
  001000      No-override
     .              .
     .              .
     .              .
```

Priorities Configured in this step.

Figure 6-27. Assign Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected VIDs and display the result.



```
ProCurve(config)# vlan 1 qos dscp 000010
ProCurve(config)# vlan 20 qos dscp 000010
ProCurve(config)# vlan 30 qos dscp 000101
ProCurve(config)# vlan 40 qos dscp 000111

ProCurve(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       DSCP        | 000010 1
  20      DSCP        | 000010 1
  30      DSCP        | 000101 5
  40      DSCP        | 000111 7
```

Figure 6-28. The Completed VID-DSCP Priority Configuration

The switch will now apply the DSCP policies in figure 6-28 to packets received on the switch with the specified VLAN-IDs. This means the switch will:

- Overwrite the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assign the 802.1p priorities in the above policies to the appropriate packets.

QoS Source-Port Priority

QoS Classifier Precedence: 6

The QoS source-port option enables you to use a packet's source-port on the switch as a QoS classifier. Where a particular source-port classifier has the highest precedence in the switch for traffic entering through that port, then traffic received from the port is marked with the source-port classifier's configured priority level. Different source-port classifiers can have different priority levels.

Options for Assigning Priority on the Switch. Priority control options for packets from a specified source-port include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets must be IPv4.)

(For operation when other QoS classifiers apply to the same traffic, refer to "Classifiers for Prioritizing Outbound Packets" on page 6-10.)

Options for Assigning Priority From a RADIUS Server. You can use a RADIUS server to impose a QoS source-port priority during an 802.1X port-access authentication session. Refer to the RADIUS chapter in the *Access Security Guide* for your switch (January 2005 or later).

Assigning a Priority Based on Source-Port

This option assigns a priority to all outbound packets having the specified source-port. You can configure this option by either specifying the source-port ahead of the **qos** command or moving to the port context for the port you want to configure for priority. (If you are configuring multiple source-ports with the same priority, you may find it easier to use the **interface <** ***port-list*** **>** command to go to the port context instead of individually configuring the priority for each port.)

Syntax: interface < *port-list* > qos priority < 0 - 7 >

Configures an 802.1p priority for packets entering the switch through the specified (source) ports. This priority determines the packet queue in the outbound port(s) to which traffic is sent. If a packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each source-port or group of source-ports. (Default: **No-override***)*

Syntax: no interface < *port-list* > qos

Disables use of the specified source-port(s) for QoS classifier(s) and resets the priority for the specified source-port(s) to **No-override**.

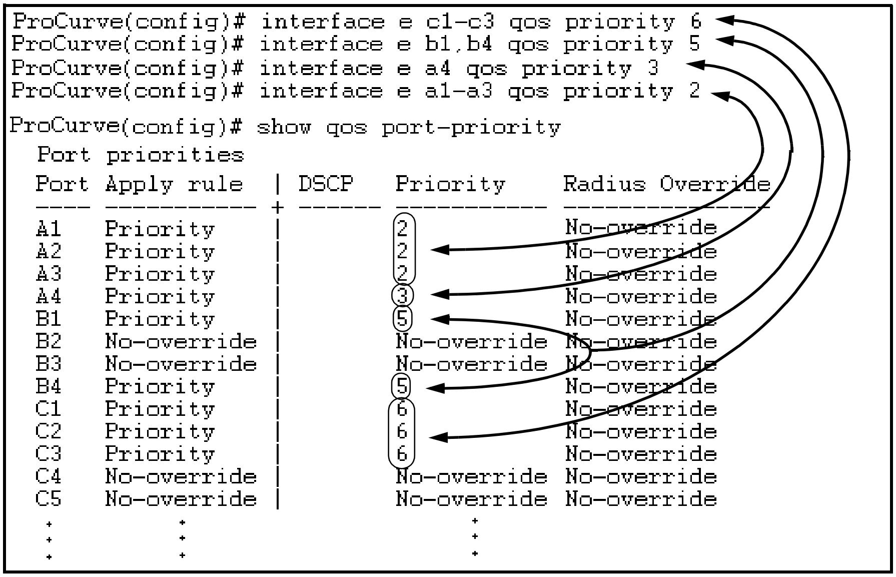
Syntax: show qos port-priority

Lists the QoS port-priority classifiers with their priority data.

For example, suppose that you want to prioritize inbound traffic on the following source-ports:

Source-Port	Priority
A1 - A3	2
A4	3
B1, B4	5
C1-C3	6

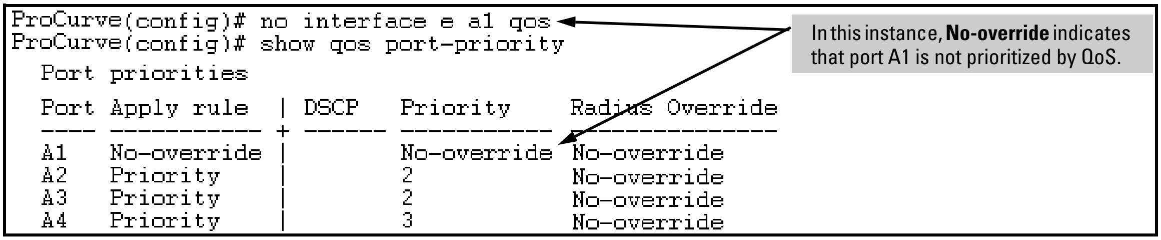
You would then execute the following commands to prioritize traffic received on the above ports:



```
ProCurve(config)# interface e c1-c3 qos priority 6
ProCurve(config)# interface e b1,b4 qos priority 5
ProCurve(config)# interface e a4 qos priority 3
ProCurve(config)# interface e a1-a3 qos priority 2

ProCurve(config)# show qos port-priority
  Port priorities
  Port Apply rule   | DSCP   Priority    Radius Override
  ---- ----------- + ------ ----------- ---------------
  A1   Priority     |        2           No-override
  A2   Priority     |        2           No-override
  A3   Priority     |        2           No-override
  A4   Priority     |        3           No-override
  B1   Priority     |        5           No-override
  B2   No-override  |        No-override No-override
  B3   No-override  |        No-override No-override
  B4   Priority     |        5           No-override
  C1   Priority     |        6           No-override
  C2   Priority     |        6           No-override
  C3   Priority     |        6           No-override
  C4   No-override  |        No-override No-override
  C5   No-override  |        No-override No-override
   .        .                     .
   .        .                     .
   .        .                     .
```

Figure 6-29. Configuring and Displaying Source-Port QoS Priorities

If you then decided to remove port A1 from QoS prioritization:



```
ProCurve(config)# no interface e a1 qos
ProCurve(config)# show qos port-priority
  Port priorities

  Port Apply rule   | DSCP   Priority    Radius Override
  ---- ----------- + ------ ----------- ---------------
  A1   No-override |        No-override No-override
  A2   Priority    |        2           No-override
  A3   Priority    |        2           No-override
  A4   Priority    |        3           No-override
```

In this instance, **No-override** indicates that port A1 is not prioritized by QoS.

Figure 6-30. Returning a QoS-Prioritized VLAN to "No-override" Status

Assigning a DSCP Policy Based on the Source-Port

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets (received from the specified source-ports). That is, the switch:

1. Selects an incoming IP packet on the basis of its source-port on the switch.
2. Overwrites the packet's DSCP with the DSCP configured in the switch for such packets.
3. Assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)
4. Forwards the packet through the appropriate outbound port queue.

For more on DSCP, refer to "Terminology" on page 6-6.

Steps for Creating a Policy Based on Source-Port Classifiers.

Note

You can select one DSCP per source-port. Also, configuring a new DSCP for a source-port automatically overwrites (replaces) any previous DSCP or 802.1p priority configuration for that port.)

1. Identify the source-port classifier to which you want to assign a DSCP policy.
2. Determine the DSCP policy for packets having the selected source-port:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received through the source-port from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.

3. Configure the DSCP policy by using **qos dscp-map** to configure the priority for each codepoint. (For details, refer to the example later in this section and to "Differentiated Services Codepoint (DSCP) Mapping" on page 6-54.)

Note

A codepoint must have an 802.1p priority assignment (0 - 7) before you can configure that codepoint as a criteria for prioritizing packets by source-port. If a codepoint shows **No-override** in the **Priority** column of the DSCP Policy Table (**show qos dscp-map**), then you must assign a 0 - 7 priority before proceeding.

4. Configure the switch to assign the DSCP policy to packets from the specified source-port.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

This command is optional if a priority has already been assigned to the < ***codepoint*** *>. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this priority to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: For most codepoints,* **No-override**. *See figure 6-9 on page 6-55 on page 6-55.)*

Syntax: interface < *port-list* > qos dscp < *codepoint* >

Assigns a DSCP policy to packets from the specified source-port(s), and overwrites the DSCP in these packets with the assigned < ***codepoint*** *> value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default:* **No-override***)*

Syntax: no interface [e] < *port-list* > qos

Removes QoS classifier for the specified source-port(s).

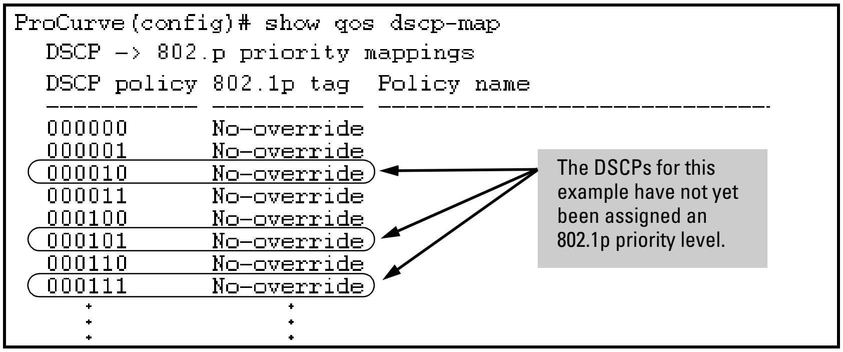
Syntax: show qos source-port

Displays a listing of all source-port QoS classifiers currently in the running-config file.

For example, suppose you wanted to assign this set of priorities:

Source-Port	DSCP	Priority
A2	000111	7
B1-B3	000101	5
B4, C2	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP. (Refer to the "Note On Changing a Priority Setting" on page 6-57. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

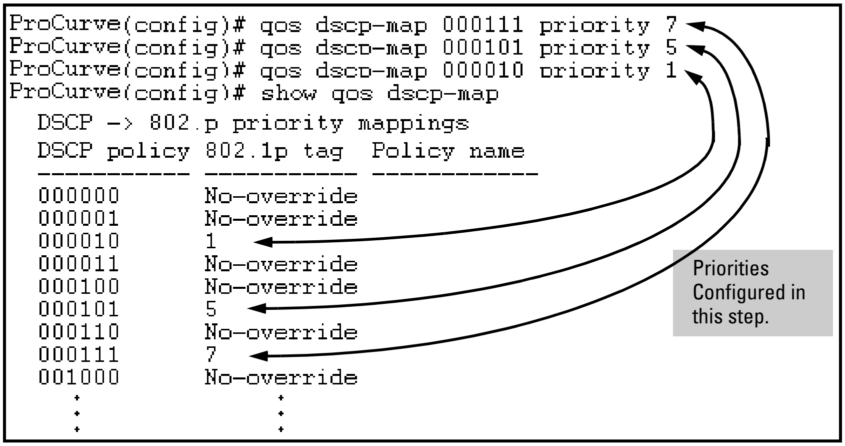


```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- -------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     .           .
     .           .
     .           .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 6-31. Display the Current Configuration in the DSCP Policy Table

2. Configure the priorities for the DSCPs you want to use.

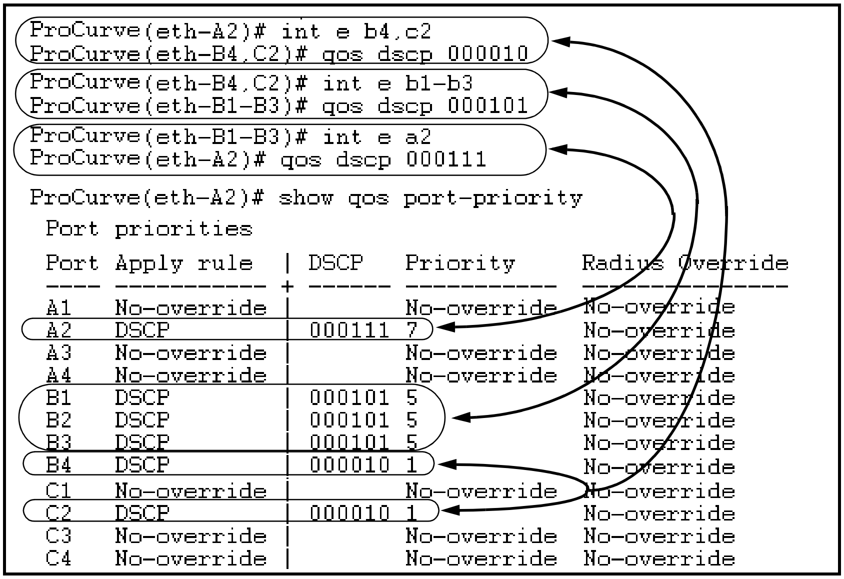


```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override
  000101      5
  000110      No-override
  000111      7
  001000      No-override
     .           .
     .           .
     .           .
```

Priorities Configured in this step.

Figure 6-32. Assign Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected source-ports and display the result.



```
ProCurve(eth-A2)# int e b4,c2
ProCurve(eth-B4,C2)# qos dscp 000010
ProCurve(eth-B4,C2)# int e b1-b3
ProCurve(eth-B1-B3)# qos dscp 000101
ProCurve(eth-B1-B3)# int e a2
ProCurve(eth-A2)# qos dscp 000111

ProCurve(eth-A2)# show qos port-priority
  Port priorities

  Port Apply rule   | DSCP   Priority     Radius Override
  ---- ----------- + ------ ----------- ---------------
  A1   No-override  |        No-override  No-override
  A2   DSCP         | 000111 7            No-override
  A3   No-override  |        No-override  No-override
  A4   No-override  |        No-override  No-override
  B1   DSCP         | 000101 5            No-override
  B2   DSCP         | 000101 5            No-override
  B3   DSCP         | 000101 5            No-override
  B4   DSCP         | 000010 1            No-override
  C1   No-override  |        No-override  No-override
  C2   DSCP         | 000010 1            No-override
  C3   No-override  |        No-override  No-override
  C4   No-override  |        No-override  No-override
```

Figure 6-33. The Completed Source-Port DSCP-Priority Configuration

Radius Override Field. During a client session authenticated by a RADIUS server, the server can impose a port priority that applies only to that client session. Refer to the RADIUS chapter in the *Access Security Guide* for your switch (January 2005 or later).

Differentiated Services Codepoint (DSCP) Mapping

The DSCP Policy Table associates an 802.1p priority with a specific ToS byte codepoint in an IPv4 packet. This enables you to set a LAN policy that operates independently of 802.1Q VLAN-tagging.

In the default state, most of the 64 codepoints do not assign an 802.1p priority, as indicated by **No-override** in table 6-9 on page 6-55.

You can use the following command to list the current DSCP Policy table, change the codepoint priority assignments, and assign optional names to the codepoints.

Syntax: show qos dscp-map

Displays the DSCP Policy Table.

qos dscp-map **< *codepoint* >** priority < 0 - 7 > [name **< *ascii-string* >**]

Configures an 802.1p priority for the specified codepoint and, optionally, an identifying (policy) name.

no qos dscp-map **< *codepoint* >**

*Reconfigures the 802.1p priority for <**codepoint**> to* **No-override**. *Also deletes the codepoint policy name, if configured.*

no qos dscp-map < *codepoint* > name

Deletes only the **policy name, if** *configured, for <* ***codepoint*** *>.*

6-9. The Default DSCP Policy Table

DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority
000000	No-override	010110	3*	101011	No-override
000001	No-override	010111	No-override	101100	No-override
000010	No-override	011000	No-override	101101	No-override
000011	No-override	011001	No-override	101110	7**
000100	No-override	011010	4*	101111	No-override
000101	No-override	011011	No-override	110000	No-override
000110	No-override	011100	4*	110001	No-override
000111	No-override	011101	No-override	110010	No-override
001000	No-override	011110	5*	110011	No-override
001001	No-override	011111	No-override	110100	No-override
001010	1*	100000	No-override	110101	No-override
001011	No-override	100001	No-override	110110	No-override
001100	1*	100010	6*	110111	No-override
001101	No-override	100011	No-override	111000	No-override
001110	2*	100100	6*	111001	No-override
001111	No-override	100101	No-override	111010	No-override
010000	No-override	100110	7*	111011	No-override
010001	No-override	100111	No-override	111100	No-override
010010	0 *	101000	No-override	111101	No-override
010011	No-override	101001	No-override	111110	No-override
010100	0 *	101010	No-override	111111	No-override
010101	No-override				

DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority
*Assured Forwarding codepoints; configured by default on the switches covered in this guide. These codepoints are configured as “No-override” in the Series 3400cl, Series 6400cl and Series 2600/2800 switches. **Expedited Forwarding codepoint configured by default.					

Default Priority Settings for Selected Codepoints

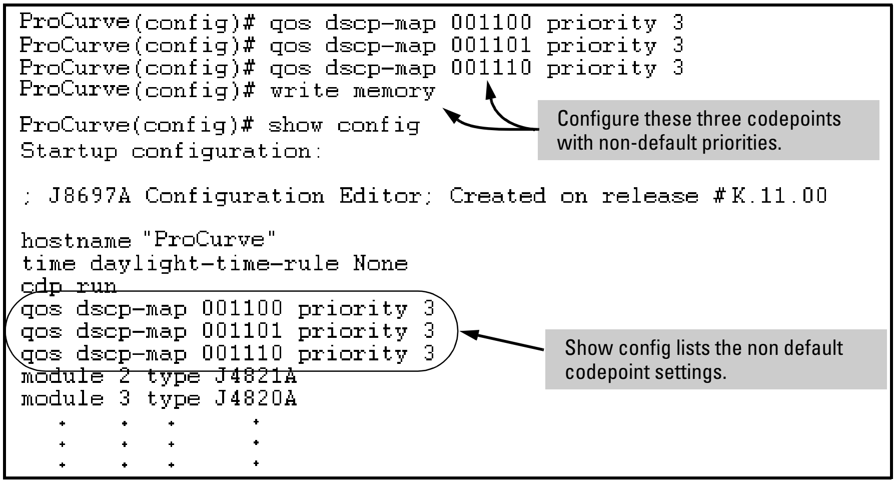
In a few cases, such as 001010 and 001100, a default policy (implied by the DSCP standards for Assured-Forwarding and Expedited-Forwarding) is used. You can change the priorities for the default policies by using **qos dscp-map < *codepoint* > priority < 0 - 7 >)**.(These policies are not in effect unless you have either applied the policies to a QoS classifier or configured QoS Type-of-Service to be in **diff-services** mode.)

Quickly Listing Non-Default Codepoint Settings

Table 6-9 lists the switch’s default codepoint/priority settings. If you change the priority of any codepoint setting to a non-default value and then execute **write memory**, the switch will list the non-default setting in the show config display. For example, in the default configuration, the following codepoint settings are true:

Codepoint	Default Priority
001100	1
001101	No-override
001110	2

If you change all three settings to a priority of 3, and then execute **write memory**, the switch will reflect these changes in the show config listing:



```
ProCurve(config)# qos dscp-map 001100 priority 3
ProCurve(config)# qos dscp-map 001101 priority 3
ProCurve(config)# qos dscp-map 001110 priority 3
ProCurve(config)# write memory
ProCurve(config)# show config
Startup configuration:

; J8697A Configuration Editor; Created on release #K.11.00

hostname "ProCurve"
time daylight-time-rule None
cdp run
qos dscp-map 001100 priority 3
qos dscp-map 001101 priority 3
qos dscp-map 001110 priority 3
module 2 type J4821A
module 3 type J4820A
   .    .   .     .
   .    .   .     .
   .    .   .     .
```

Configure these three codepoints with non-default priorities.

Show config lists the non default codepoint settings.

Figure 6-34. Example of Show Config Listing with Non-Default Priority Settings in the DSCP Table

Effect of "No-override". In the QoS Type-of-Service differentiated services mode, a **No-override** assignment for the codepoint of an outbound packet means that QoS is effectively disabled for such packets. That is, QoS does not affect the packet queuing priority or VLAN tagging. In this case, the packets are handled as follows (as long as no other QoS feature creates priority assignments for them):

802.1Q Status	Outbound 802.1p Priority
Received and Forwarded on a tagged port member of a VLAN.	Unchanged
Received on an Untagged port member of a VLAN; Forwarded on a tagged port member of a VLAN.	0 (zero)—"normal"
Forwarded on an Untagged port member of a VLAN.	None

Note On Changing a Priority Setting

If a QoS classifier is using a policy (codepoint and associated priority) in the DSCP Policy table, you must delete or change this usage before you can change the priority setting on the codepoint. Otherwise the switch blocks the change and displays this message:

Cannot modify DSCP Policy < *codepoint* > - in use by other qos rules.

In this case, use **show qos < *classifier* >** to identify the specific classifiers using the policy you want to change; that is:

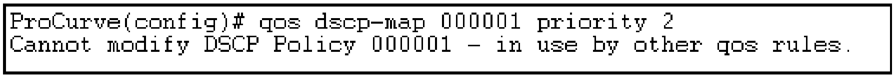
```
show qos device-priority
show qos port-priority
show qos tcp-udp-port-priority
show qos vlan-priority
show qos type-of-service
```

For example, suppose that the 000001 codepoint has a priority of 6, and several classifiers use the 000001 codepoint to assign a priority to their respective types of traffic. If you wanted to change the priority of codepoint 000001 you would do the following:

1. Identify which QoS classifiers use the codepoint.
2. Change the classifier configurations by assigning them to a different DSCP policy, or to an 802.1p priority, or to **No-override**.
3. Reconfigure the desired priority for the 000001 codepoint.
4. Either reassign the classifiers to the 00001 codepoint policy or leave them as they were after step 2, above.

Example of Changing the Priority Setting on a Policy When One or More Classifiers Are Currently Using the Policy

Suppose that codepoint 000001 is in use by one or more classifiers. If you try to change its priority, you see a result similar to the following:

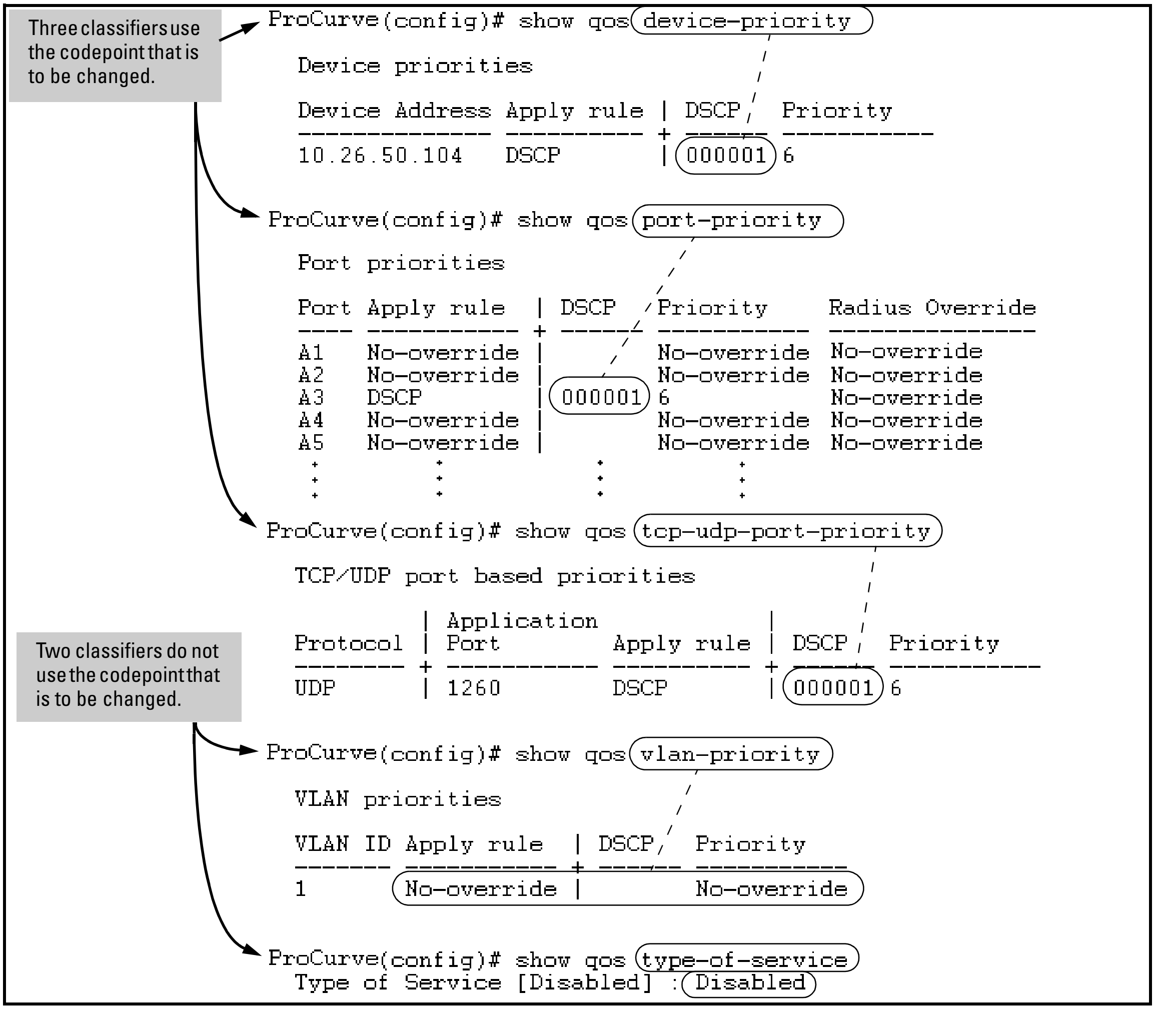


```
ProCurve(config)# qos dscp-map 000001 priority 2
Cannot modify DSCP Policy 000001 - in use by other qos rules.
```

Figure 6-35. Example of Trying To Change the Priority on a Policy In Use by a Classifier

In this case, you would use steps similar to the following to change the priority.

1. Identify which classifiers use the codepoint you want to change.



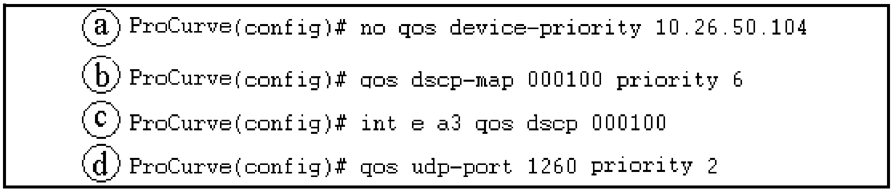
```
ProCurve(config)# show qos device-priority

  Device priorities

  Device Address Apply rule | DSCP   Priority
  -------------- ---------- + ------ -----------
  10.26.50.104   DSCP       | 000001 6

ProCurve(config)# show qos port-priority

  Port priorities

  Port Apply rule  | DSCP   Priority    Radius Override
  ---- ----------- + ------ ----------- ---------------
  A1   No-override |        No-override No-override
  A2   No-override |        No-override No-override
  A3   DSCP        | 000001 6           No-override
  A4   No-override |        No-override No-override
  A5   No-override |        No-override No-override
   .        .           .        .
   .        .           .        .
   .        .           .        .
ProCurve(config)# show qos tcp-udp-port-priority

  TCP/UDP port based priorities

           | Application            |
  Protocol | Port        Apply rule | DSCP   Priority
  -------- + ----------- ---------- + ------ -----------
  UDP      | 1260        DSCP       | 000001 6

ProCurve(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       No-override |        No-override

ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] :Disabled
```

Three classifiers use the codepoint that is to be changed.

Two classifiers do not use the codepoint that is to be changed.

Figure 6-36. Example of a Search to Identify Classifiers Using a Codepoint You Want To Change

2. Change the classifier configurations by assigning them to a different DSCP policy, or to an 802.1p priority, or to **No-override**. For example:
   a. Delete the policy assignment for the **device-priority** classifier. (That is, assign it to **No-override**.)
   b. Create a new DSCP policy to use for re-assigning the remaining classifiers.
   c. Assign the **port-priority** classifier to the new DSCP policy.
   d. Assign the **udp-port 1260** classifier to an 802.1p priority.



```
ProCurve(config)# no qos device-priority 10.26.50.104
ProCurve(config)# qos dscp-map 000100 priority 6
ProCurve(config)# int e a3 qos dscp 000100
ProCurve(config)# qos udp-port 1260 priority 2
```

3. Reconfigure the desired priority for the 000001 codepoint.

```
ProCurve(config)# qos dscp-map 000001 priority 4
```

4. You could now re-assign the classifiers to the original policy codepoint or leave them as currently configured.

IP Multicast (IGMP) Interaction with QoS

IGMP high-priority-forward causes the switch to service the subscribed IP multicast group traffic at high priority, even if QoS on the switch has relegated the traffic to a lower priority. This does not affect any QoS priority settings, so the QoS priority is honored by downstream devices. However, QoS does take precedence over IGMP normal-priority traffic.

The switch's ability to prioritize IGMP traffic for either a normal or high priority outbound queue overrides any QoS criteria, and does not affect any 802.1p priority settings the switch may assign. For a given packet, if both IGMP high priority and QoS are configured, the QoS classification occurs and the switch marks the packet for downstream devices, but the packet is serviced by the high-priority queue when leaving the switch.

IGMP High Priority	QoS Configuration Affects Packet	Switch Port Output Queue	Outbound 802.1p Setting (Requires Tagged VLAN)
Not Enabled	Yes	Determined by QoS	Determined by QoS
Enabled	See above paragraph.	High	As determined by QoS if QoS is active.

QoS Messages in the CLI

Message	Meaning
DSCP Policy < ***decimal-codepoint*** > not configured	You have attempted to map a QoS classifier to a codepoint for which there is no configured priority (**No-override**). Use the **qos dscp-map** command to configure a priority for the codepoint, then map the classifier to the codepoint.
Cannot modify DSCP Policy < ***codepoint*** > - in use by other qos rules.	You have attempted to map a QoS classifier to a codepoint that is already in use by other QoS classifiers. Before remapping the codepoint to a new priority, you must reconfigure the other QoS classifiers so that they do not use this codepoint. You can have multiple QoS classifiers use this same codepoint as long as it is acceptable for all such classifiers to use the same priority.

QoS Operating Notes and Restrictions

6-10. Details of Packet Criteria and Restrictions for QoS Support

Packet Criteria or Restriction	**QoS Classifiers**							**DSCP Overwrite (Re-Marking)**
	UDP/TCP	**Device Priority (IP Address)**	**IP Type-of-Service**	**Layer 3 Protocol**	**VLAN**	**Source Port**	**Incoming 802.1p**	
Restricted to IPv4 Packets Only	Yes	Yes	Yes	No	No	No	No	Yes
Allow Packets with IP Options[1]	Yes	Yes[2]	Yes[2]	Yes[2]	Ye2[3]	Yes[2]	Yes[2]	No
Support IPv6 Packets[1]	No	No	No	Yes	Yes	Yes	Yes	No
Support Layer-2 SAP Encapsulation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[1] For explicit QoS support of IPv6 packets, force IPv6 traffic into its own set of VLANs and then configure VLAN-based classifiers for those VLANs.

[2] On IPv4 packets with IP options, the switches covered in this guide support QoS for 802.1p priority policies, but does **not** do any DSCP re-marking for DSCP policies.

- **All Switches:** For explicit QoS support of IP subnets, ProCurve recommends forcing IP subnets onto separate VLANs and then configuring VLAN-based classifiers for those VLANs.
- **For Devices that Do Not Support 802.1Q VLAN-Tagged Ports:** For communication between these devices and the switch, connect the device to a switch port configured as **Untagged** for the VLAN in which you want the device's traffic to move.
- **Port Tagging Rules:** For a port on the switch to be a member of a VLAN, the port must be configured as either **Tagged** or **Untagged** for that VLAN. A port can be an untagged member of only one VLAN of a given protocol type. Otherwise, the switch cannot determine which VLAN should receive untagged traffic. For more on VLANs, refer to chapter 2, "Static Virtual LANs (VLANs)".
- **Maximum QoS Configuration Entries:** The switches covered in this guide accept the maximum outbound priority and/or DSCP policy configuration entries shown in table 6-11.

6-11. Maximum QoS Entries.

Switch	Software Version	Maximum QoS Entries	Notes
Series 5400zl Series 5300yl Switch 6200yl		250*	• Each device (IP address) QoS configuration uses two entries. • Each TCP/UDP port QoS configuration uses four entries. • All other classifier configurations use one entry each.
*Configuring device (IP address) or TCP/UDP QoS entries reduces this maximum. See the "Notes" column.			

Attempting to exceed the above limits generates the following message in the CLI:

```
Unable to add this QoS rule. Maximum number (entry-#)
already reached.
```

- **Series 6200yl, 5400zl, 3500yl Switches—Non-Supported IP Packets:** The DSCP policy codepoint-remarking operation is not supported in any QoS classifier for packets carrying IP options in the packet header.
- **Not Supported:** Use of an inbound 802.1p packet priority as a classifier for remapping a packet's outbound priority to different 802.1p priority. For example, where inbound packets carry an 802.1p priority of 1, QoS cannot be configured use this priority as a classifier for changing the outbound priority to 0.
- **Monitoring Shared Resources:** The QoS feature shares internal switch resources with several other features. The switch provides ample resources for all features. However, if the internal resources become fully subscribed, additional QoS provisions cannot be configured until the necessary resources are released from other uses. For information on determining the current resource availability and usage, refer to the appendix titled "Monitoring Resources" in the *Management and Configuration Guide* for your switch.

Access Control Lists (ACLs)

Contents

Introduction

This chapter describes how to configure, apply, and edit Access Control Lists (ACLs) in a network populated with the routing switches covered by this guide (with IP routing support enabled) and how to monitor ACL actions.

.

Feature	Default	CLI
Standard ACLs	None	7-41
Extended ACLs	None	7-50
Enable or Disable an ACL	n/a	7-71
Display ACL Data	n/a	7-83
Delete an ACL	n/a	7-72
Configure an ACL from a TFTP Server	n/a	7-89
Enable ACL Logging	n/a	7-96

Layer 3 IP filtering with ACLs can help improve network performance and restrict network use by creating policies for:

- **Switch Management Access:** Permits or denies in-band management access. This includes limiting and/or preventing the use of designated protocols that ride on top of IP, such as TCP, UDP, IGMP, ICMP, and others. Also included are the use of precedence and ToS criteria, and control for application transactions based on source and destination IP addresses and transport layer port numbers.
- **Application Access Security:** Eliminates unwanted IP traffic in a path by filtering packets where they enter or leave the routing switch on specific VLAN interfaces.

ACLs can filter traffic to or from a host, a group of hosts, or entire subnets.

Notes

ACLs can enhance network security by blocking selected IP traffic, and can serve as part of your network security program. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

ACLs on the routing switches covered by this manual do not screen non-IP traffic such as AppleTalk and IPX.

For ACL filtering to take effect, configure an ACL and then assign it to either the inbound or outbound traffic on a statically configured VLAN on the routing switch. (Except for ACEs that screen traffic to an IP address on the routing switch itself, ACLs assigned to VLANs can operate only while IP routing is enabled. Refer to “Notes on IP Routing” on page 7-16.)

Table 7-1. Command Summary for Standard ACLs

Action	Command(s)	Page
Create a Standard, **Named** ACL *or* Add an ACE to the End of an Existing Standard, **Named** ACL	ProCurve(config)# ip access-list standard < *name-str* > ProCurve(config-std-nacl)# < deny | permit > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] [log] [2]	7-43
Create a Standard, **Numbered** ACL or Add an ACE to the End of an Existing Standard, **Numbered** ACL	ProCurve(config)# access-list < 1-99 > < deny | permit > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >> [log][2]	7-46
Use a Sequence Number To Insert an ACE in a Standard ACL	ProCurve(config)# ip access-list standard < *name-str* | 1-99 > ProCurve(config-std-nacl)# 1-2147483647 < deny | permit > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] [log] [2]	7-74
Use an ACE’s Sequence Number To Delete the ACE from a Standard ACL	ProCurve(config)# ip access-list standard < *name-str* | 1-99 > ProCurve(config-std-nacl)# no < 1-2147483647 >	7-77
Resequence the ACEs in a Standard ACL	ProCurve(config)# ip access-list resequence < *name-str* | 1-99 > < 1-2147483647 > < 1-2147483646 >	7-78
Enter or Remove a Remark from a Standard ACL	ProCurve(config)# ip access-list standard < *name-str* | 1-99 > ProCurve(config-ext-nacl)# [remark < *remark-str* > | no remark] *For numbered, standard ACLs only, the following* **remark** *commands can be substituted for the above:* ProCurve(config)# access-list < 1 - 99 > remark < remark-str > ProCurve(config)# [no] access-list < 1 - 99 > remark	7-79 7-81
Delete a Standard ACL	ProCurve(config)# no ip access-list standard < *name-str* | 1-99 > *For numbered, standard ACLs, the following command can be substituted for the above:* ProCurve(config)# access-list < 1 - 99 > remark < remark-str >	7-72

[1] The mask can be in either dotted-decimal notation (such as 0.0.15.255) or CIDR notation (such as /20).
[2] The [log] function applies only to “deny” ACLs, and generates a message only when there is a “deny” match.

Table 7-2. Command Summary for Extended ACLs

Action	Command(s)	Page
Create an Extended, **Named** ACL *or* Add an ACE to the End of an Existing, Extended ACL	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-std-nacl)# < deny | permit > < ip | *ip-protocol* | *ip-protocol-nbr* > < any | host <*SA* > | *SSA*/< mask-length > | *SA* < *mask* >>[1] < any | host < *DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] < tcp | udp > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] [*comparison-operator* < *value* >] < any | host <*DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] [*comparison-operator* < *value* >] [established] < igmp > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] < any | host < *DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] [*igmp-packet-type*] < icmp > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] < any | host < *DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] [[< 0 - 255 > [0 - 255]] | *icmp-message*] [precedence < *priority* >] [tos < *tos- setting* >] [log][2]	7-52
Create an Extended, **Numbered** ACL or Add an ACE to the End of an Existing, **Numbered** ACL	ProCurve(config)# access-list < 100-199 > < deny | permit > < *ip-options* |*tcp/udp-options* |*igmp-options* |*icmp-options* > [precedence < *priority* >] [tos < *tos- setting* >] [log][2] ***Note:*** *Uses the same IP, TCP/UDP, IGMP, and ICMP options as shown above for "Create an Extended, Named ACL".*	7-64
Insert an ACE by Assigning a Sequence Number	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-ext-nacl)# 1-2147483647 < deny | permit > *Uses the options shown above for "Create an Extended, Named ACL".*	7-75
Delete an ACE by Specifying Its Sequence Number	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-std-nacl)# no < 1-2147483647 >	7-77
Resequence the ACEs in an ACL	ProCurve(config)# ip access-list resequence < *name-str* | 100-199 > < 1-2147483647 > < 1-2147483646 >	7-78

[1] The mask can be in either dotted-decimal notation (such as 0.0.15.255) or CIDR notation (such as /20).
[2] The [log] function applies only to "deny" ACLs, and generates a message only when there is a "deny" match.

Action	Command(s)	Page
Enter or Remove a Remark	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-ext-nacl)# [remark < *remark-str* > | no remark] *For numbered, extended ACLs only, the following* **remark** *commands can be substituted for the above:* ProCurve(config)# access-list < 100 - 199 > remark < remark-str > ProCurve(config)# [no] access-list < 100 - 199 > remark	7-79 7-81
Delete an Extended ACL	ProCurve(config)# no ip access-list extended < *name-str* | 100-199 > *For numbered, extended ACLs only, the following command can also be used:* ProCurve(config)# no access-list < 100 - 199 >	7-72

Table 7-3. Command Summary for Enabling, Disabling, and Displaying ACLs

Enable or Disable an ACL	ProCurve(config)# [no] vlan < vid > ip access-group < *identifier* > < in | out >	7-71
Displaying ACL Data	ProCurve(config)# show access-list ProCurve(config)# show access-list < *acl-identifier* > ProCurve(config)# show access-list config ProCurve(config)# show access-list vlan < *vid* > ProCurve(config)# show access-list radius	7-83

Terminology

Access Control Entry (ACE): A policy consisting of criteria and an action (permit or deny) to take on a packet if it meets the criteria. The elements composing the criteria include:

- source IP address and mask (standard and extended ACLs)
- destination IP address and mask (extended ACLs only)
- either of the following:
  - all IP traffic
  - traffic of a specific IP protocol (extended ACLs only)
    (In the cases of TCP, UDP, ICMP, and IGMP, the criteria can include either all traffic of the protocol type or only the traffic of a specific sub-type within the protocol.)
- optional use of IP precedence and ToS settings (extended ACLs only)

Access Control List (ACL): A list (or set) consisting of one or more explicitly configured Access Control Entries (ACEs) and terminating with an implicit “deny” default which drops any packets that do not have a match with any explicit ACE in the named ACL. The two classes of ACLs are “standard” and “extended”. See “Standard ACL” and “Extended ACL”.

ACE: See “Access Control Entry”.

ACL: See “Access Control List”.

ACL ID: A number or alphanumeric string used to identify an ACL. A *standard* ACL ID can have either an alphanumeric string or a number in the range of 1 to 99. An *extended* ACL ID can have either an alphanumeric string or a number in the range of 100 to 199.

ACL Mask: Follows any IP address (source or destination) listed in an ACE. Defines which bits in a packet’s corresponding IP addressing must exactly match the IP addressing in the ACE, and which bits need not match (wildcards). See also “How an ACE Uses a Mask To Screen Packets for Matches” on page 7-26.)

CIDR: This is the acronym for Classless Inter-Domain Routing.

Connection-Rate ACL: An optional feature used with Connection-Rate filtering based on virus-throttling technology. For more information, refer to the chapter titled “Virus Throttling” in the *Access Security Guide* for your routing switch.

DA: The acronym used in text to represent *Destination IP Address*. In an IP packet, this is the destination IP address carried in the header, and identifies the destination intended by the packet's originator. In an extended ACE, this is the second of two IP addresses required by the ACE to determine whether there is a match between a packet and the ACE. See also "SA".

Deny: An ACE configured with this action causes the routing switch to drop a packet for which there is a match within an applicable ACL.

Extended ACL: This type of Access Control List uses layer-3 IP criteria composed of source and destination IP addresses and (optionally) TCP/UDP port, ICMP, IGMP, precedence, or ToS criteria to determine whether there is a match with an IP packet. You can apply extended ACLs to either inbound or outbound routed traffic and to any inbound switched or routed traffic with a DA belonging to the routing switch itself. Extended ACLs require an alphanumeric name or an identification number (ID) in the range of 100 - 199.

Implicit Deny: If the routing switch finds no matches between a routed packet and the configured criteria in an applicable ACL, then the routing switch denies (drops) the packet with an implicit **deny any** function (for standard ACLs) or an implicit **deny ip any any** function (for extended ACLs). You can preempt the Implicit Deny in a given ACL by configuring a **permit any** (standard) or **permit IP any any** (extended) as the last explicit ACE in the ACL. Doing so permits any routed IP packet that is not explicitly permitted or denied by other ACEs configured sequentially earlier in the ACL. Unless otherwise noted, Implicit Deny refers to the "deny" function enforced by both standard and extended ACLs.

identifier*:** The term used in ACL syntax statements to represent either the name or number by which the ACL can be accessed. See also ***name-str.

Inbound Traffic: For the purpose of defining where the routing switch applies ACLs to filter traffic, inbound traffic is any IP packet that:

- *Enters the routing switch* on a given VLAN interface or, in the case of a multinetted VLAN, a given subnet.
- Has a destination IP address (DA) that meets either of these criteria:
  - The packet's DA is for an external device on a different VLAN or subnet than the VLAN or subnet on which it arrived.
  - The packet's DA is for an IP address configured on the routing switch itself. (This increases your options for protecting the routing switch from unauthorized management access.)

Because ACLs are assigned to VLANs, an ACL that filters inbound traffic on a particular VLAN examines packets meeting the above criteria that have entered the routing switch through any port on that VLAN.

name-str: The term used in extended ACL syntax statements to represent the "name string"; the alphanumeric string used to identify the ACL. See also ***identifier***.

Named ACL: An ACL created with the **ip access-list < extended | standard > < name-str >** command and then populated using the **< deny | permit >** command in the Named ACL (**nacl**) CLI context. (Refer to "Entering the "Named ACL" (nacl) Context" on page 7-43.)

Numbered ACL: An ACL created and initially populated by using the **access-list < 1-99 | 100 - 199 >** command. (Refer to "Creating or Adding to a Standard, Numbered ACL" on page 7-47.) After a numbered ACL has been created, the routing switch manages it in the same way as a named ACL, meaning that it can be opened and edited in the same way as a named ACL.

Outbound Traffic: For defining the points where the routing switch applies ACLs to filter traffic, outbound traffic is routed traffic *leaving the routing switch* through a VLAN interface or, in the case of a multinetted VLAN, a given subnet. This includes traffic routed between subnets in the same VLAN. Note that for ACL purposes, "outbound traffic" does not include traffic that is switched instead of routed. (Refer also to "ACL Inbound and Outbound Application Points" on page 7-12.)

Permit: An ACE configured with this action allows the routing switch to forward a routed packet for which there is a match within an applicable ACL.

Permit Any Forwarding: An ACE configured with this action causes the routing switch to forward all routed packets that have not been permitted or denied by earlier ACEs in the list. In a standard ACL, this is **permit any**. In an extended ACL, it is **permit ip any any**.

remark-str: The term used in ACL syntax statements to represent the variable "remark string"; a set of alphanumeric characters you can include in a remark in an ACl. A remark string can include up to 100 characters and must be delimited by single or double quotes if any spaces are included in the string.

SA: The acronym for *Source IP Address*. In an IP packet, this is the source IP address carried in the IP header, and identifies the packet's sender. In an extended ACE, this is the first of two IP addresses used by the ACE to determine whether there is a match between a packet and the ACE. See also "DA".

seq-#: The term used in ACL syntax statements to represent the sequence number variable used to insert an ACE within an existing list. The range allowed for sequence numbers is 1 - 2147483647.

Standard ACL: This type of Access Control List uses the layer-3 IP criteria of source IP address to determine whether there is a match with an IP packet. You can apply standard ACLs to either inbound or outbound routed traffic and to any inbound switched or routed traffic with a DA belonging to the routing switch itself. Standard ACLs require an alphanumeric name or an identification number (ID) in the range of 1- 99.

Wildcard: The part of a mask that indicates the bits in a packet's IP addressing that do not need to match the corresponding bits specified in an ACL. See also **ACL Mask** on page 7-8.

Overview

Types of IP ACLs

Standard ACL: Use a standard ACL when you need to permit or deny traffic based on source IP address only. Standard ACLs are also useful when you need to quickly control a performance problem by limiting traffic from a subnet, group of devices, or a single device. (This can block all IP traffic from the configured source, but does not hamper traffic from other sources within the network.) This ACL type uses an alphanumeric ID string or a numeric ID of 1 through 99. You can specify a single host, a finite group of hosts, or any host.

Extended ACL: Extended ACLs are useful whenever simple IP source address restrictions do not provide sufficient traffic selection criteria needed on a VLAN interface. Extended ACLs allow use of the following criteria:

- source and destination IP address combinations
- IP protocol options

Extended, named ACLs also offer an option to permit or deny the establishment of IP connections using TCP for applications such as Telnet, http, ftp, and others.

Connection-Rate ACL. An optional feature used with Connection-Rate filtering based on virus-throttling technology. For more information, refer to the chapter titled “Virus Throttling” in the *Access Security Guide* for your routing switch.

ACL Inbound and Outbound Application Points

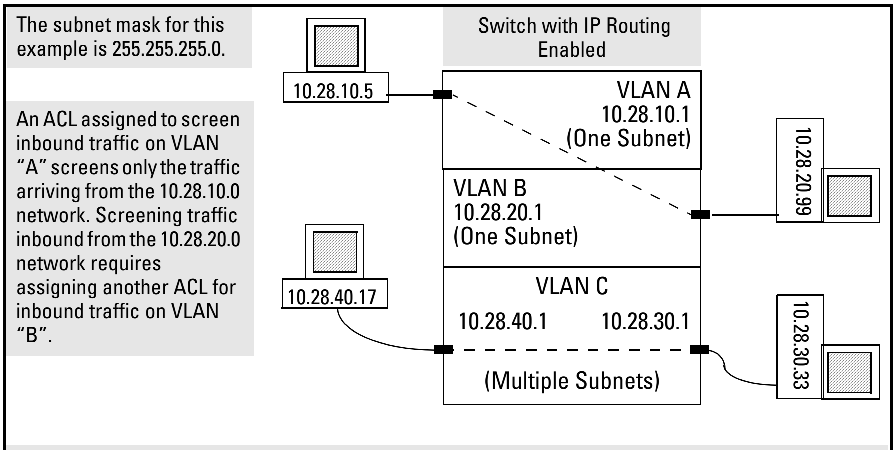
You can apply ACL filtering to the following types of traffic:

- IP traffic routed between different subnets. (IP routing *must* be enabled.)
- IP traffic carrying a destination address (DA) on the routing switch itself. In figure 7-1, below, this is any of the IP addresses shown in VLANs “A”, “B”, and “C” on the routing switch. (IP routing need not be enabled.)

The routing switch can apply ACL filtering to traffic *entering or leaving the routing switch* on VLANs configured to apply ACL filters. (When you assign an ACL to a VLAN, you must specify whether the ACL will filter inbound or outbound traffic.) For example, in figure 7-1:

- You would assign either an inbound ACL on VLAN "A" or an outbound ACL on VLAN "B" to filter a packet routed between subnets; that is, from the workstation 10.28.10.5 on VLAN "A" to the server at 10.28.20.99 on VLAN "B". (An outbound ACL on VLAN "A" or an inbound ACL on VLAN "B" would not filter the packet.)
- Where multiple subnets are configured on the same VLAN, *if*:
  - Traffic you want to filter moves between subnets on the same VLAN.
  - The traffic source and destination IP addresses are on devices external to the routing switch.

  Then you can use either inbound or outbound ACLs to filter the traffic on the VLAN (because the traffic moves between subnets but enters and leaves the routing switch in the same VLAN.)


The subnet mask for this example is 255.255.255.0.
An ACL assigned to screen inbound traffic on VLAN "A" screens only the traffic arriving from the 10.28.10.0 network. Screening traffic inbound from the 10.28.20.0 network requires assigning another ACL for inbound traffic on VLAN "B".
10.28.10.5
Switch with IP Routing Enabled
VLAN A
10.28.10.1
(One Subnet)
10.28.20.99
VLAN B
10.28.20.1
(One Subnet)
10.28.40.17
VLAN C
10.28.40.1
10.28.30.1
10.28.30.33
(Multiple Subnets)

Because of multinetting, traffic routed from the 10.28.40.0 network to the 10.28.30.0 network, and the reverse, remains in VLAN C. This allows you to apply one inbound ACL to screen traffic arriving from either subnet, and one outbound ACL to screen routed traffic going to either subnet.

Figure 7-1. Example of Filter Applications

Note

ACLs do not filter traffic that remains in the same subnet from source to destination (switched traffic) unless the destination IP address (DA) is on the routing switch itself.

Features Common to All per-VLAN ACLs

- On any VLAN you can apply one ACL to inbound traffic and one ACL to outbound traffic. You can use the same ACL or different ACLs for the inbound and outbound traffic.
- Any ACL can have multiple entries (ACEs).
- You can apply any one ACL to multiple VLANs.
- All ACEs in an ACL are automatically sequenced (numbered). For an existing ACL, entering an ACE without specifying a sequence number automatically places the ACE at the end of the list. Specifying a sequence number inserts the ACE into the list at the correct sequential location.
- Automatic sequence numbering begins with "10" and increases in increments of 10. You can renumber the ACEs in an ACL and also change the sequence increment between ACEs.
- The CLI **remark** command option allows you to enter a separate comment for each ACE.
- A source or destination IP address and a mask, together, can define a single host, a range of hosts, or all hosts.
- Every ACL populated with one or more explicit ACEs includes an Implicit Deny as the last entry in the list. The routing switch applies this action to any packets that do not match other criteria in the ACL. (For standard ACLs, the Implicit Deny is **deny any**. For extended ACLs, it is **deny ip any any**.)
- In any ACL, you can apply an ACL log function to ACEs that have an explicit "deny" action. The logging occurs when there is a match on a "deny" ACE. (The routing switch sends ACL logging output to Syslog and, optionally, to a console session.)

You can configure ACLs using either the CLI or a text editor. The text-editor method is recommended when you plan to create or modify an ACL that has more entries than you can easily enter or edit using the CLI alone. Refer to "Creating or Editing ACLs Offline" on page 7-89.

General Steps for Planning and Configuring ACLs

1. Identify the traffic type to filter. Options include:
   - Any routed IP traffic
   - Any routed IP traffic of a specific protocol type (0-255)
   - Any routed TCP traffic (only) for a specific TCP port or range of ports, including optional control of connection traffic based on whether the initial request should be allowed
   - Any routed UDP traffic (only) or routed UDP traffic for a specific UDP port
   - Any routed ICMP traffic (only) or routed ICMP traffic of a specific type and code
   - Any routed IGMP traffic (only) or routed IGMP traffic of a specific type
   - Any of the above with specific precedence and/or ToS settings
2. The SA and/or the DA of routed traffic you want to permit or deny.
3. Determine the best points at which to apply specific ACL controls. For example, you can improve network performance by filtering unwanted traffic at the edge of the network instead of in the core. Also, on the routing switch itself, you can improve performance by filtering unwanted traffic where it is inbound to the routing switch instead of outbound.
4. Design the ACLs for the control points you have selected. Where you are using explicit "deny" ACEs, you can optionally use the ACL logging feature for notification that the routing switch is denying unwanted packets.
5. Create the ACLs in the selected routing switches.
6. Assign the ACLs to filter the inbound and/or outbound traffic on static VLAN interfaces configured on the routing switch.
7. Enable IP routing on the routing switch. (Except for an ACL configured to filter traffic having the routing switch itself as the destination IP address, IP routing must be enabled before ACLs will operate.)
8. Test for desired results.

For more details on ACL planning considerations, refer to "Planning an ACL Application" on page 7-21.

Notes on IP Routing

To activate an ACL to screen inbound traffic for routing between subnets, assign the ACL to the statically configured VLAN on which the traffic enters the routing switch. Also, ensure that IP routing is enabled. Similarly, to activate an ACL to screen routed, outbound traffic, assign the ACL to the statically configured VLAN on which the traffic exits from the routing switch. The only exception to these rules is for an ACL configured to screen inbound traffic with a destination IP address on the routing switch. In this case, an ACL assigned to a VLAN screens traffic addressed to an IP address on the routing switch, regardless of whether IP routing is also enabled. (ACLs do not screen outbound traffic generated by the routing switch, itself. Refer to "ACL Screening of Traffic Generated by the Routing Switch" on page 7-99.)

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the routing switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to use the **no ip source-route** command to disable source routing on the routing switch. (If source routing is disabled in the running-config file, the **show running** command includes "**no ip source-route**" in the running-config file listing.)

ACL Operation

Introduction

An ACL is a list of one or more Access Control Entries (ACEs), where each ACE consists of a matching criteria and an action (permit or deny). An ACL applies only to the routing switch in which it is configured. ACLs operate on assigned static VLANs, and filter these traffic types:

- Routed traffic entering or leaving the routing switch on a VLAN. (Note that ACLs do not screen traffic at the internal point where traffic moves between VLANs or subnets within the routing switch. Refer to "ACL Inbound and Outbound Application Points" on page 7-12.)
- Switched or routed traffic entering the routing switch on a VLAN and having an IP address on the routing switch itself as the destination

You can apply one inbound ACL and one outbound ACL to each static VLAN configured on the routing switch. The complete range of options per VLAN includes:

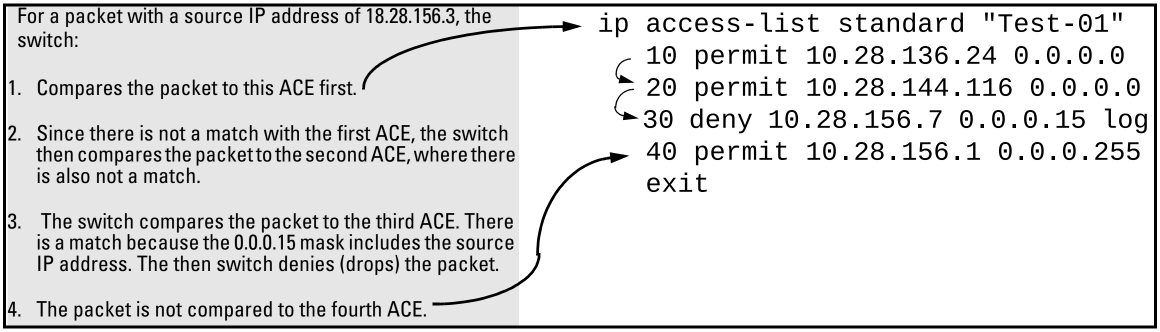
- **No ACL** assigned to a static VLAN. (In this case, all traffic entering or leaving the routing switch on the VLAN does so without any ACL filtering, which is the default.)
- **One ACL** assigned to filter *either* the inbound or the outbound traffic entering or leaving the routing switch on a static VLAN.
- **One ACL** assigned to filter *both* the inbound and the outbound traffic entering or leaving the routing switch on a static VLAN.
- **Two different ACLs** assigned to a static VLAN; one for filtering traffic entering the routing switch and one for filtering traffic leaving the routing switch.

Note

On a given routing switch, after you assign an ACL to a static VLAN, the default action for all physical ports belonging to the VLAN is to implicitly deny any IP traffic that is not specifically permitted by the ACL. (This applies only in the direction of traffic flow filtered by the ACL.)

The Packet-Filtering Process

Sequential Comparison and Action. When the routing switch uses an ACL to filter a packet, it sequentially compares each ACE's filtering criteria to the corresponding data in the packet until it finds a match.



For a packet with a source IP address of 18.28.156.3, the switch:

1. Compares the packet to this ACE first.
2. Since there is not a match with the first ACE, the switch then compares the packet to the second ACE, where there is also not a match.
3. The switch compares the packet to the third ACE. There is a match because the 0.0.0.15 mask includes the source IP address. The then switch denies (drops) the packet.
4. The packet is not compared to the fourth ACE.

```
ip access-list standard "Test-01"
  10 permit 10.28.136.24 0.0.0.0
  20 permit 10.28.144.116 0.0.0.0
  30 deny 10.28.156.7 0.0.0.15 log
  40 permit 10.28.156.1 0.0.0.255
  exit
```

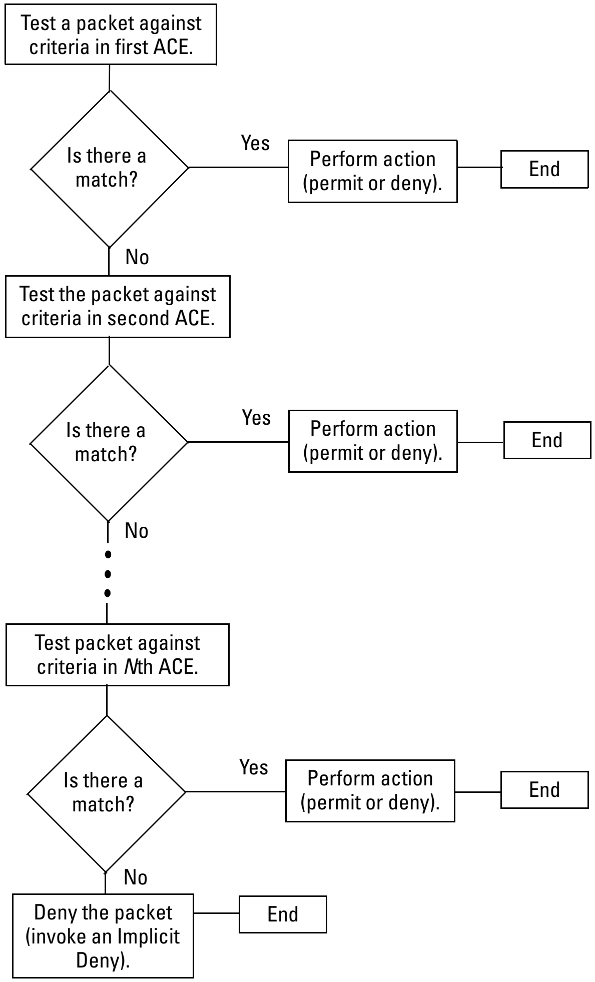
Figure 7-2. Example of Sequential Comparison

That is, the routing switch tries the first ACE in the list. If there is not a match, it tries the second ACE, and so on. When a match is found, the routing switch invokes the configured action for that entry (permit or drop the packet) and no further comparisons of the packet are made with the remaining ACEs in the ACL. This means that when the routing switch finds an ACE whose criteria matches a packet, it invokes the action configured for that ACE, and any remaining ACEs in the ACL are ignored. *Because of this sequential processing, successfully implementing an ACL depends in part on configuring ACEs in the correct order for the overall policy you want the ACL to enforce.*

Implicit Deny. If a packet does not have a match with the criteria in any of the ACEs in the ACL, the routing switch denies (drops) the packet. (This is termed *implicit deny.*) If you need to override the implicit deny so that any packet that does not have a match will be permitted, then you can enter an ACE with Permit Any forwarding as the last ACE in the ACL. This directs the routing switch to permit (forward) any packets that do not have a match with any earlier ACE in the ACL, and prevents these packets from being filtered by the implicit deny.

Note on Implicit Deny

For ACLs configured to filter inbound packets on a VLAN, remember that Implicit Deny filters routed packets *and any bridged packets with a DA specifying the routing switch itself.* This operation helps to prevent management access from unauthorized IP sources.


Test a packet against criteria in first ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Test the packet against criteria in second ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Test packet against criteria in *N*th ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Deny the packet (invoke an Implicit Deny).
End

1. If a match is not found with the first ACE in an ACL, the switch proceeds to the next ACE and so on.
2. If a match with an explicit ACE is subsequently found, the packet is either permitted (forwarded) or denied (dropped), depending on the action specified in the matching ACE. In this case the switch ignores all subsequent ACEs in the ACL.
3. If a match is not found with any explicit ACE in the ACL, the switch invokes the Implicit Deny at the end of every ACL, and drops the packet.

Note: If the list includes an ACE configured with Permit Any forwarding, no packets can reach the Implicit Deny at the end of the list. Also, placing an ACE with Permit Any forwarding at any point in an ACL defeats the purpose of any subsequent ACEs in the list.

Figure 7-3. The Packet-Filtering Process in an ACL with *N* Entries (ACEs)

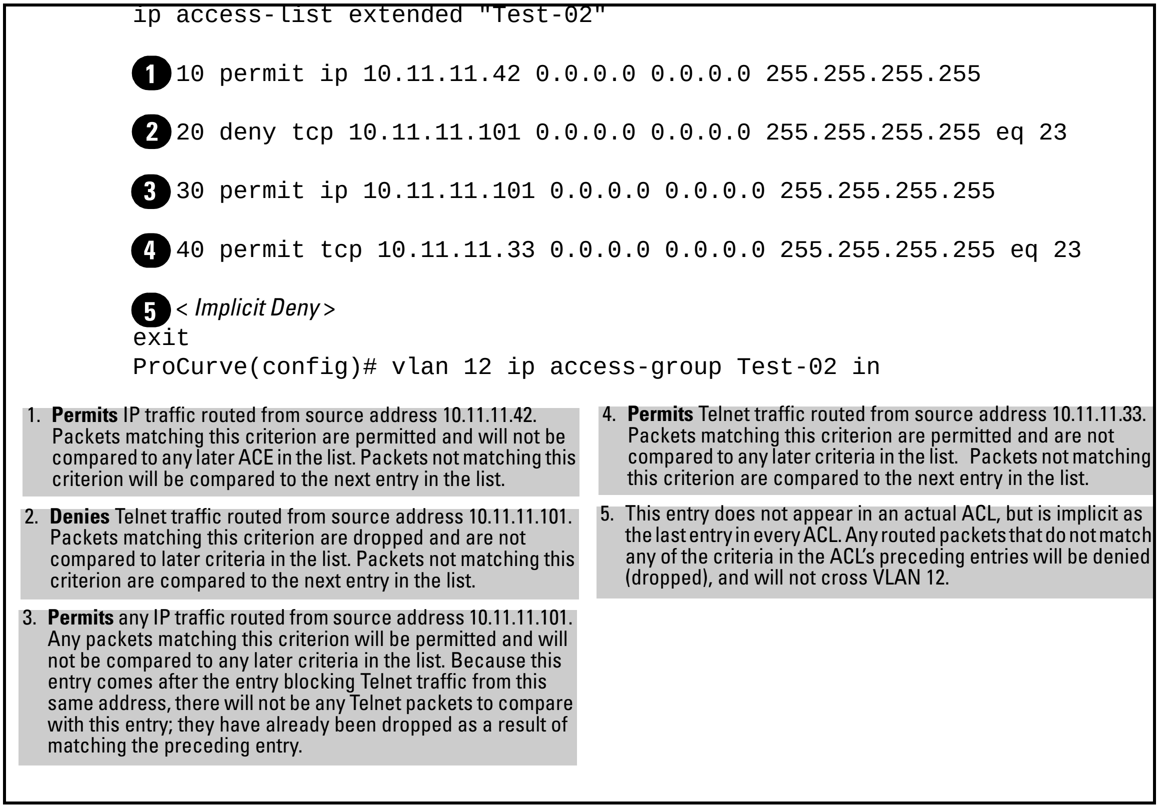
Note

The order in which an ACE occurs in an ACL is significant. For example, if an ACL contains six ACEs, but the first ACE allows Permit Any forwarding, then the ACL permits all IP traffic, and the remaining ACEs in the list do not apply, even if they specify criteria that would make a match with any of the traffic permitted by the first ACE.

For example, suppose you want to configure an ACL on the routing switch (with an ID of "Test-02") to invoke these policies:

1. Permit all inbound traffic on VLAN 12 routed from IP address 10.11.11.42.
2. Deny *only* the inbound Telnet traffic routed from address 10.11.11.101.
3. Permit *only* inbound Telnet traffic routed from IP address 10.11.11.33.
4. Deny *all other* inbound routed traffic on VLAN 12.

The following ACL model, when assigned to inbound filtering on VLAN 12, supports the above case:



```
ip access-list extended "Test-02"

❶10 permit ip 10.11.11.42 0.0.0.0 0.0.0.0 255.255.255.255

❷20 deny tcp 10.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255 eq 23

❸30 permit ip 10.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255

❹40 permit tcp 10.11.11.33 0.0.0.0 0.0.0.0 255.255.255.255 eq 23

❺< Implicit Deny >
exit
ProCurve(config)# vlan 12 ip access-group Test-02 in
```

1. **Permits** IP traffic routed from source address 10.11.11.42. Packets matching this criterion are permitted and will not be compared to any later ACE in the list. Packets not matching this criterion will be compared to the next entry in the list.
2. **Denies** Telnet traffic routed from source address 10.11.11.101. Packets matching this criterion are dropped and are not compared to later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.
3. **Permits** any IP traffic routed from source address 10.11.11.101. Any packets matching this criterion will be permitted and will not be compared to any later criteria in the list. Because this entry comes after the entry blocking Telnet traffic from this same address, there will not be any Telnet packets to compare with this entry; they have already been dropped as a result of matching the preceding entry.
4. **Permits** Telnet traffic routed from source address 10.11.11.33. Packets matching this criterion are permitted and are not compared to any later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.
5. This entry does not appear in an actual ACL, but is implicit as the last entry in every ACL. Any routed packets that do not match any of the criteria in the ACL's preceding entries will be denied (dropped), and will not cross VLAN 12.

Figure 7-4. Example of How an ACL Filters Packets

It is important to remember that this ACL (and all ACLs) include an *Implicit Deny*. That is, routed IP packets (and switched packets having the routing switch as the destination IP address) that the ACL does not *explicitly* permit or deny will be *implicitly* denied, and therefore dropped instead of forwarded on the VLAN. You can preempt the implicit deny by inserting an ACE that allows Permit Any forwarding at the end of an ACL, but this solution does not apply in the preceding example, where the intention is for the routing switch to forward only explicitly permitted packets routed on VLAN 12.

Overriding the Implicit Deny. If you want an ACL to permit any routed packets that are not explicitly denied by other entries in the ACL, you can do so by configuring an ACE with Permit Any forwarding as the last entry in the ACL. Doing so permits any packet not explicitly denied by earlier entries.

Planning an ACL Application

Before creating and implementing ACLs, you need to define the policies you want your ACLs to enforce, and understand how your ACLs will impact your network users.

Traffic Management and Improved Network Performance

You can use ACLs to block unnecessary traffic caused by individual hosts, workgroups, or subnets, and to block user access to subnets, devices, and services. Traffic types you can use as criteria for ACLs include:

- Any routed IP traffic
- Any routed IP traffic of a specific protocol type (0-255)
- Any routed TCP traffic (only) for a specific TCP port or range of ports, including optional control of connection traffic based on whether the initial request should be allowed
- Any routed UDP traffic (only) or routed UDP traffic for a specific UDP port
- Any routed ICMP traffic (only) or routed ICMP traffic of a specific type and code
- Any routed IGMP traffic (only) or routed IGMP traffic of a specific type
- Any of the above with specific precedence and/or ToS settings

Answering the following questions can help you to design and properly position ACLs for optimum network usage.

- What are the logical points for minimizing unwanted traffic? In many cases it makes sense to prevent unwanted traffic from reaching the core of your network by configuring ACLs to drop unwanted traffic at or close to the edge of the network. (The earlier in the network path you can block unwanted traffic, the greater the benefit for network performance.)
- What traffic should you explicitly block? Depending on your network size and the access requirements of individual hosts, this can involve creating a large number of ACEs in a given ACL (or a large number of ACLs), which increases the complexity of your solution.
- What traffic can you implicitly block by taking advantage of the implicit **deny IP any** to deny traffic that you have not explicitly permitted? This can reduce the number of entries needed in an ACL.
- What traffic should you permit? In some cases you will need to explicitly identify permitted traffic. In other cases, depending on your policies, you can insert an ACE with Permit Any forwarding at the end of an ACL. This means that all IP traffic not specifically matched by earlier entries in the list will be permitted.

Security

ACLs can enhance security by blocking routed IP traffic carrying an unauthorized source IP address (SA). This can include:

- Blocking access to or from subnets in your network
- Blocking access to or from the internet
- Blocking access to sensitive data storage or restricted equipment
- Preventing the use of specific TCP or UDP functions (such as Telnet, SSH, web browser) for unauthorized access

You can also enhance routing-switch management security by using ACLs to block bridged IP traffic that has the routing switch itself as the destination address (DA).

Caution

ACLs can enhance network security by blocking selected IP traffic, and can serve as one aspect of maintaining network security. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

Note

ACLs in the routing switches covered by this Guide do not screen non-IP traffic such as AppleTalk and IPX.

Guidelines for Planning the Structure of an ACL

The first step in planning a specific ACL is to determine where you will apply it. (Refer to "ACL Inbound and Outbound Application Points" on page 7-12.) You must then determine the order in which you want the individual ACEs in the ACL to filter traffic.

- The first match dictates the action on a packet. Subsequent matches are ignored.
- On any ACL, the routing switch implicitly denies IP packets that are not explicitly permitted or denied by the ACEs configured in the ACL. If you want the routing switch to forward a packet for which there is not a match in an ACL, append an ACE that enables Permit Any forwarding as the last ACE in an ACL. This ensures that no packets reach the Implicit Deny case.
- Generally, you should list ACEs from the most specific (individual hosts) to the most general (subnets or groups of subnets) unless doing so permits traffic that you want dropped. For example, an ACE allowing a small group of workstations to use a specialized printer should occur earlier in an ACL than an entry used to block widespread access to the same printer.

ACL Configuration and Operating Rules

- **Routing.** Except for any IP traffic with a DA on the routing switch itself, ACLs filter only routed traffic. Thus, if routing is not enabled on the routing switch, there is no routed traffic for ACLs to filter. (To enable routing, execute **ip routing** at the global configuration level.) For more on routing, refer to the chapter titled "IP Routing Features" in the *Multicast and Routing Guide* for your switch.

- **Per Routing-Switch ACL Limits.** At a minimum an ACL must have one, explicit "permit" or "deny" Access Control Entry. You can configure up to 2048 ACL assignments to VLANs, as follows:
  - Named (Extended or Standard) ACLs: Up to 2048 (minus any numeric ACL assignments)
  - Numeric Standard ACLs: Up to 99; numeric range: 1 - 99
  - Numeric Extended ACLs: Up to 100; numeric range: 100 - 199
  - Total ACEs in all ACLs: Depends on the combined resource usage by ACL, QoS, IDM, Virus-Throttling, ICMP, and Management VLAN features (For more on this topic, refer to "Monitoring Shared Resources" on page 7-99.)
- **Implicit Deny:** In any ACL, the routing switch automatically applies an implicit "deny IP any" that does not appear in **show** listings. This means that the ACL denies any packet it encounters that does not have a match with an entry in the ACL. Thus, if you want an ACL to permit any packets that you have not expressly denied, you must enter a **permit any** or **permit ip any any** as the last ACE in an ACL. Because, for a given packet the routing switch sequentially applies the ACEs in an ACL until it finds a match, any packet that reaches the **permit any** or **permit ip any any** entry will be permitted, and will not encounter the "deny ip any" ACE the routing switch automatically includes at the end of the ACL. For an example, refer to figure 7-4 on page 7-20.
- **Explicitly Permitting Any IP Traffic:** Entering a **permit any** or a **permit ip any any** ACE in an ACL permits all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point do not have any effect.
- **Explicitly Denying Any IP Traffic:** Entering a **deny any** or a **deny ip any any** ACE in an ACL denies all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point have no effect.
- **Replacing One ACL with Another:** The last ACL assigned for inbound ("in") or outbound ("out") packet filtering on an interface replaces any other ACL previously configured for the same purpose. For example, if you configured ACL 100 to filter inbound traffic on VLAN 20, but later, you configured ACL 112 to filter inbound traffic on this same VLAN, ACL 112 replaces ACL 100 as the ACL to use for filtering inbound traffic on VLAN 20.

- **ACLs Operate On Static VLANs:** You can assign an ACL to any VLAN that is statically configured on the routing switch. ACLs do not operate with dynamic VLANs.

- **An ACL Affects All Physical Ports in a Static VLAN:** An ACL assigned to a VLAN applies to all physical ports on the routing switch that belong to that VLAN, including ports that have dynamically joined the VLAN.

- **ACLs Screen Routed Traffic Entering or Leaving the Routing Switch on a Given VLAN Interface:** This means that the following traffic is subject to ACL filtering:

  - Traffic arriving on one VLAN and leaving on another VLAN
  - Traffic arriving on one subnet and leaving on another subnet within the same, a multinetted VLAN

  Filtering the desired traffic requires assigning an ACL to screen traffic inbound or outbound on the appropriate VLAN(s). In the case of a multinetted VLAN, it means that traffic inbound from different subnets in the same VLAN is screened by the same inbound ACL, and traffic outbound from different subnets is screened by the same outbound ACL. (Refer to figure 7-1 on page 7-13.)

- **ACLs Do Not Filter Switched Traffic Unless the Routing Switch Itself is the DA:** ACLs do not filter:

  - Traffic moving between ports belonging to the same subnet
  - Traffic leaving the routing switch with an SA on the routing switch itself

  ACLs *do* filter switched or routed traffic having a DA on the routing switch.

How an ACE Uses a Mask To Screen Packets for Matches

When the routing switch applies an ACL to inbound or outbound traffic in a VLAN, each ACE in the ACL uses an IP address and *ACL mask* to enforce a selection policy on the packets being screened. That is, the mask determines the range of IP addresses (SA only or SA/DA) that constitute a match between the policy and a packet being screened.

What Is the Difference Between Network (or Subnet) Masks and the Masks Used with ACLs?

In common IP addressing, a network (or subnet) mask defines which part of the IP address to use for the network number and which part to use for the hosts on the network. For example:

IP Address	Mask	Network Address	Host Address
10.38.252.195	255.255.255.0	first three octets	The fourth octet.
10.38.252.195	255.255.248.0	first two octets and the left-most five bits of the third octet	The right most three bits of the third octet and all bits in the fourth octet.

Thus, the bits set to 1 in a network mask define the part of an IP address to use for the network number, and the bits set to 0 in the mask define the part of the address to use for the host number.

In an ACL, IP addresses and masks provide the criteria for determining whether to deny or permit a packet, or to pass it to the next ACE in the list. If there is a match, the deny or permit action occurs. If there is not a match, the packet is compared with the next ACE in the ACL. Thus, where a standard network mask defines how to identify the network and host numbers in an IP address, the mask used with ACEs defines which bits in a packet's IP address must match the corresponding bits in the IP address listed in an ACE, and which bits can be *wildcards*.

Rules for Defining a Match Between a Packet and an Access Control Entry (ACE)

- For a given ACE, when the routing switch compares an IP address and corresponding mask in the ACE to an IP address carried in a packet:
  - **A mask-bit setting of 0 ("off")** requires that the corresponding bit in the packet's IP address and in the ACE's IP address must be the same. That is, if a bit in the ACE's IP address is set to 1 ("on"), the same bit in the packet's IP address must also be 1.
  - **A mask-bit setting of 1 ("on")** means the corresponding bit in the packet's IP address and in the ACE's IP address do not have to be the same. That is, if a bit in the ACE's IP address is set to 1, the same bit in the packet's IP address can be either 1 or 0 ("on" or "off").

  For an example, refer to "Example of How the Mask Bit Settings Define a Match" on page 7-29.

- In any ACE, a mask of all ones means *any* IP address is a match. Conversely, a mask of all zeros means the *only* match is an IP address identical to the host IP address specified in the ACL.

- Depending on your network, a single ACE that allows a match with more than one source or destination IP address may allow a match with multiple subnets For example, in a network with a prefix of 31.30.240 and a subnet mask of 255.255.240.0 (the leftmost 20 bits), applying an ACL mask of 0.0.31.255 causes the subnet mask and the ACL mask to overlap one bit, which allows matches with hosts in two subnets: 31.30.224.0 and 31.30.240.0.

Bit Position in the Third Octet of Subnet Mask 255.255.240.0								
Bit Values	128	64	32	16	8	4	2	1
Subnet Mask Bits	1	1	1	1	n/a	n/a	n/a	n/a
Mask Bit Settings Affecting Subnet Addresses	0	0	0	**1 or 0**	n/a	n/a	n/a	n/a

This ACL supernetting technique can help to reduce the number of ACLs you need. You can apply it to a multinetted VLAN and to multiple VLANs. However, ensure that you exclude subnets that do not belong in the policy. If this creates a problem for your network, you can eliminate the unwanted match by making the ACEs in your ACL as specific as possible, and using multiple ACEs carefully ordered to eliminate unwanted matches.

- Every IP address and mask pair (source or destination) used in an ACE creates one of the following policies:
  - **Any IP address fits the matching criteria.** In this case, the routing switch automatically enters the IP address and mask in the ACE. For example:

    ```
    access-list 1 deny any
    ```

    produces this policy in an ACL listing:

IP Address	Mask
0.0.0.0	255.255.255.255

    This policy states that every bit in every octet of a packet's SA is a wildcard, which covers any IP address.
  - **One IP address fits the matching criteria.** In this case, you provide the IP address and the routing switch provides the mask. For example:

    ```
    access-list 1 permit host 10.28.100.15
    ```

    produces this policy in an ACL listing:

IP Address	Mask
10.28.100.15	0.0.0.0

    This policy states that every bit in every octet of a packet's SA must be the same as the corresponding bit in the SA defined in the ACE.
  - **A group of IP addresses fits the matching criteria.** In this case you provide both the IP address and the mask. For example:

    ```
    access-list 1 permit 10.28.32.1 0.0.0.31
    ```

IP Address	Mask
10.28.32.1	0.0.0.31

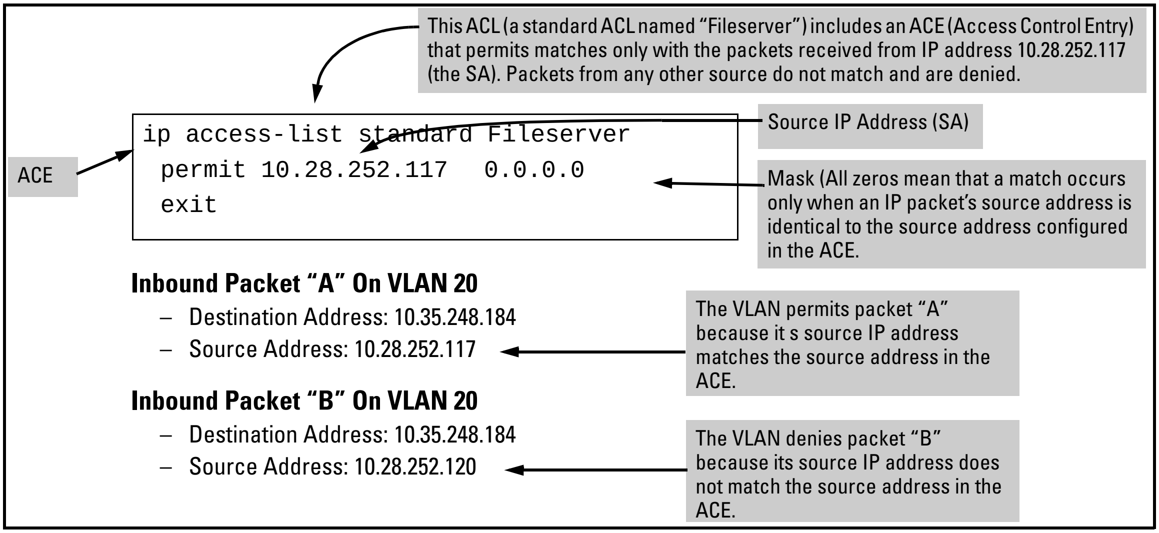
    This policy states that:
    - In the first three octets of a packet's SA, every bit must be set the same as the corresponding bit in the SA defined in the ACE.
    - In the last octet of a packet's SA, the first three bits must be the same as in the ACE, but the last five bits are wildcards and can be any value.
- Unlike subnet masks, the wildcard bits in an ACL mask need not be contiguous. For example, 0.0.7.31 is a valid ACL mask. However, a subnet mask of 255.255.248.224 is not a valid subnet mask.

Example of How the Mask Bit Settings Define a Match . Assume an ACE where the second octet of the mask for an SA is 7 (the rightmost three bits are “on”, or “1”) and the second octet of the corresponding SA in the ACE is 31 (the rightmost five bits). In this case, a match occurs when the second octet of the SA in a packet being filtered has a value in the range of 24 to 31. Refer to table 7-4, below.

Table 7-4. Example of How the Mask Defines a Match

Location of Octet	Bit Position in the Octet							
	128	64	32	16	8	4	2	1
SA in ACE	0	0	0	1	1	1	1	1
Mask for SA	0	0	0	0	0	1	1	1
Corresponding Octet of a Packet’s SA	0	0	0	1	1	0/1	0/1	0/1
The shaded area indicates bits in the packet that must exactly match the bits in the source IP in the ACE. Wherever the mask bits are ones (wildcards), the IP bits in the packet can be any value, and where the mask bits are zeros, the IP bits in the packet must be the same as the IP bits in the ACE. **Note:** This example covers only one octet of an IP address. An actual ACE applies this method to all four octets of an IP address.								

Example of Allowing Only One IP Address (“Host” Option). Suppose, for example, that you have configured the ACL in figure 7-5 to filter inbound packets on VLAN 20. Because the mask is all zeros, the ACE policy dictates that a match occurs only when the source IP address on such packets is identical to the IP address configured in the ACE.



This ACL (a standard ACL named “Fileserver”) includes an ACE (Access Control Entry) that permits matches only with the packets received from IP address 10.28.252.117 (the SA). Packets from any other source do not match and are denied.

```
ip access-list standard Fileserver
 permit 10.28.252.117   0.0.0.0
 exit
```

Mask (All zeros mean that a match occurs only when an IP packet’s source address is identical to the source address configured in the ACE.

Inbound Packet “A” On VLAN 20

– Destination Address: 10.35.248.184
– Source Address: 10.28.252.117

The VLAN permits packet “A” because it s source IP address matches the source address in the ACE.

Inbound Packet “B” On VLAN 20

– Destination Address: 10.35.248.184
– Source Address: 10.28.252.120

The VLAN denies packet “B” because its source IP address does not match the source address in the ACE.

Figure 7-5. Example of an ACL with an Access Control Entry (ACE) that Allows Only One Source IP Address

Examples Allowing Multiple IP Addresses. Table 7-5 provides examples of how to apply masks to meet various filtering requirements.

Table 7-5. Example of Using an IP Address and Mask in an Access Control Entry

IP Address in the ACE	Mask	Policy for a Match Between a Packet and the ACE	Allowed IP Addresses
A: 10.38.252.195	0.0.0.255	Exact match in first three octets only.	10.38.252.< 0-255 > (See row A in table 7-6, below.)
B: 10.38.252.195	0.0.7.255	Exact match in the first two octets and the leftmost five bits (248) of the third octet.	10.38.< 248-255 >.< 0-255 > (In the third octet, only the rightmost three bits are wildcard bits. The leftmost five bits must be a match, and in the ACE, these bits are all set to 1. See row B in table 7-6, below.)
C: 10.38.252.195	0.0.0.0	Exact match in all octets.	10.38.252.195 (There are no wildcard bits in any of the octets. See row C in table 7-6, below.)
D: 10.38.252.195	0.15.255.255	Exact match in the first octet and the leftmost four bits of the second octet.	10.< 32-47 >.< 0-255 >.<0-255> (In the second octet, the rightmost four bits are wildcard bits. See row D in table 7-6, below.)

Table 7-6. Mask Effect on Selected Octets of the IP Addresses in Table 7-5

IP Addr	Octet	Mask	Octet Range	128	64	32	16	8	4	2	1
A	3	**0** all bits	**252**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
B	3	**7** last 3 bits	**248-255**	**1**	**1**	**1**	**1**	**1**	**0 or 1**	**0 or 1**	**0 or 1**
C	4	**0** all bits	**195**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**
D	2	**15** last 4 bits	**32-47**	**0**	**0**	**1**	**0**	**0 or 1**	**0 or 1**	**0 or 1**	**0 or 1**

Shaded areas indicate bit settings that must be an exact match.

If there is a match between the policy in the ACE and the IP address in a packet, then the packet is either permitted or denied, according to how the ACE is configured. If there is not a match, the next ACE in the ACL is then applied to the packet. The same operation applies to a destination IP address (DA) used in an extended ACE. (Where an ACE includes both source and destination IP addresses, there is one IP-address/ACL-mask pair for the source address, and another IP-address/ACL-mask pair for the destination address. See "Configuring and Assigning an ACL" on page 7-31.)

CIDR Notation. For information on using CIDR notation to specify ACL masks, refer to "Using CIDR Notation To Enter the ACL Mask" on page 7-40.

Configuring and Assigning an ACL

ACL Feature	Page
Configuring and Assigning a Standard ACL	7-41
Configuring and Assigning an Extended ACL	7-50
Enabling or Disabling ACL Filtering	7-71

Overview

General Steps for Implementing ACLs

1. Configure at least one ACL. This creates and stores the ACL(s) in the routing switch configuration.
2. Assign an ACL. This applies the ACL to either the inbound or outbound traffic on a designated VLAN.
3. Enable IP routing. Except for instances where the routing switch is the destination, assigned ACLs screen IP traffic only when routing is enabled on the routing switch.

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the routing switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to disable source routing on the routing switch. To do so, execute **no ip source-route**.

Types of ACLs

- **Standard ACL:** Uses only a packet's source IP address as a criterion for permitting or denying the packet. For a standard ACL ID, use either a unique numeric string in the range of 1-99 or a unique name string of up to 64 alphanumeric characters.

- **Extended ACL:** Offers the following criteria as options for permitting or denying a packet:
  - source IP address
  - destination IP address
  - IP protocol options:
    - Any routed IP traffic
    - Any routed IP traffic of a specific protocol type (0-255)
    - Any routed TCP traffic (only) for a specific TCP port or range of ports, including optional control of connection traffic based on whether the initial request should be allowed
    - Any routed UDP traffic (only) or routed UDP traffic for a specific UDP port
    - Any routed ICMP traffic (only) or routed ICMP traffic of a specific type and code
    - Any routed IGMP traffic (only) or routed IGMP traffic of a specific type
    - Any of the above with specific precedence and/or ToS settings

  For an extended ACL ID, use either a unique number in the range of 100-199 or a unique name string of up to 64 alphanumeric characters.

Carefully plan ACL applications before configuring specific ACLs. For more on this topic, refer to “Planning an ACL Application” on page 7-21.

ACL Configuration Structure

After you enter an ACL command, you may want to inspect the resulting configuration. This is especially true where you are entering multiple ACEs into an ACL. Also, it will be helpful to understand the configuration structure when using later sections in this chapter.

The basic ACL structure includes four elements:

1. ACL identity and type: This identifies the ACL as **standard** or **extended** and shows the ACL name or number.

2. Optional **remark** entries.

3. One or more deny/permit list entries (ACEs): One entry per line.

Element	Notes
Type	Standard or Extended
Identifier	• Alphanumeric; Up to 64 Characters, Including Spaces • Numeric: 1 - 99 (Standard) or 100 - 199 (Extended)
Remark	Allows up to 100 alphanumeric characters, including blank spaces. (If any spaces are used, the remark must be enclosed in a pair of single or double quotes.) A remark is associated with a particular ACE and will have the same sequence number as the ACE. (One remark is allowed per ACE.)
Maximum ACEs Per per Routing Switch	The upper limit on ACEs supported by the routing switch depends on the concurrent resource usage by configured QoS;, ICMP rate-limiting, management VLAN, and virus-throttling features. Refer to "Monitoring Shared Resources" on page 7-99.

4. Implicit Deny: Where an ACL is in use, it denies any packets that do not have a match with the ACEs explicitly configured in the list. The Implicit Deny does not appear in ACL configuration listings, but always functions when the routing switch uses an ACL to filter packets. (You cannot delete the Implicit Deny, but you can supersede it with a **permit any** or **permit ip any any** statement.)

Standard ACL Structure

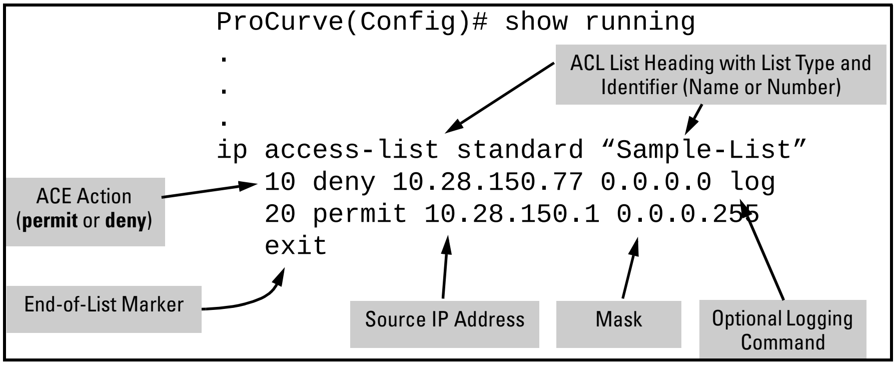
Individual ACEs in a standard ACL include only a permit/deny "type" statement, the source IP addressing, and an optional **log** command (available with "deny" statements).

```
ip access-list standard < identifier >"
   [ [ seq-# ] remark < remark-str >]
   < permit | deny > < SA > [log]
   .
   .
   .
   < Implicit Deny >
   exit
```

Note: The optional **log** function is available only for explicit "deny" ACEs.

Figure 7-6. Example of the General Structure for a Standard ACL

For example, figure 7-7 shows how to interpret the entries in a standard ACL.



```
ProCurve(Config)# show running
.
.
.
ip access-list standard “Sample-List”
   10 deny 10.28.150.77 0.0.0.0 log
   20 permit 10.28.150.1 0.0.0.255
   exit
```

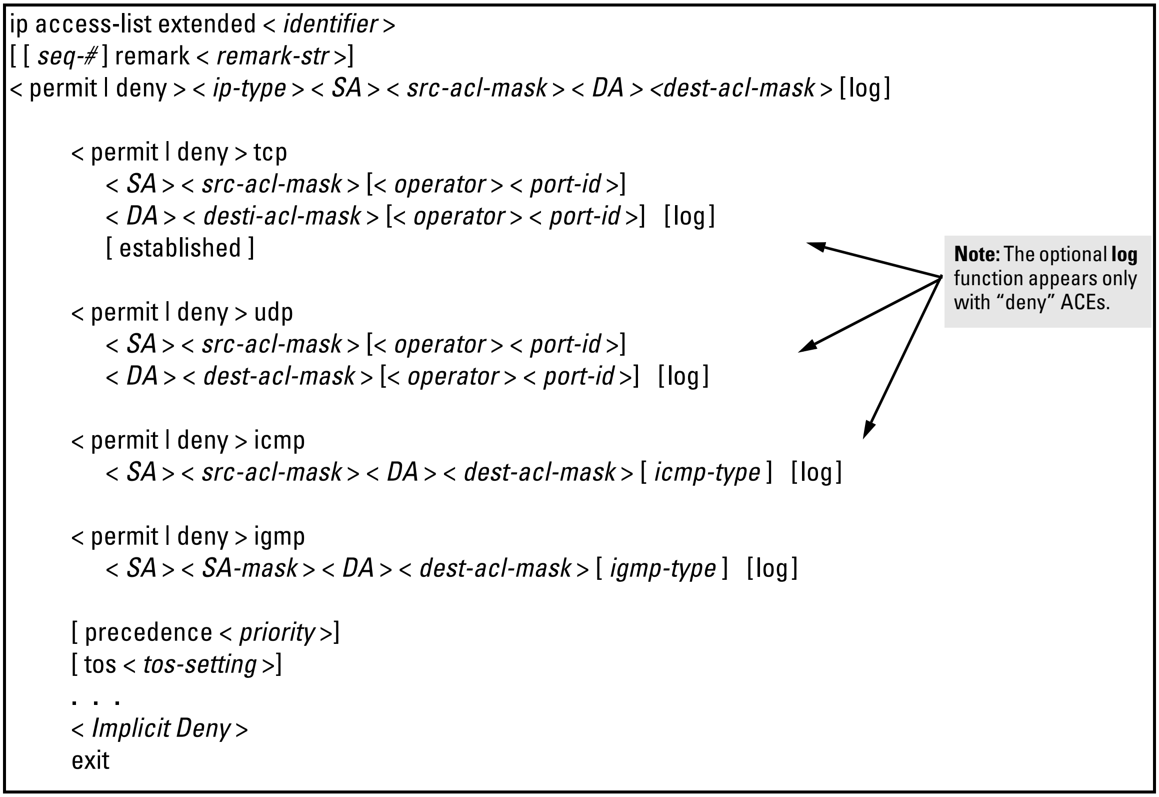
ACL List Heading with List Type and Identifier (Name or Number)
ACE Action (**permit** or **deny**)
End-of-List Marker
Source IP Address
Mask
Optional Logging Command

Figure 7-7. Example of a Displayed Standard ACL Configuration with Two ACEs

Extended ACL Configuration Structure

Individual ACEs in an extended ACL include:

- A permit/deny "type" statement
- Source and destination IP addressing
- Choice of IP criteria, including optional precedence and ToS
- Optional ACL **log** command (for **deny** entries)
- Optional remark statements

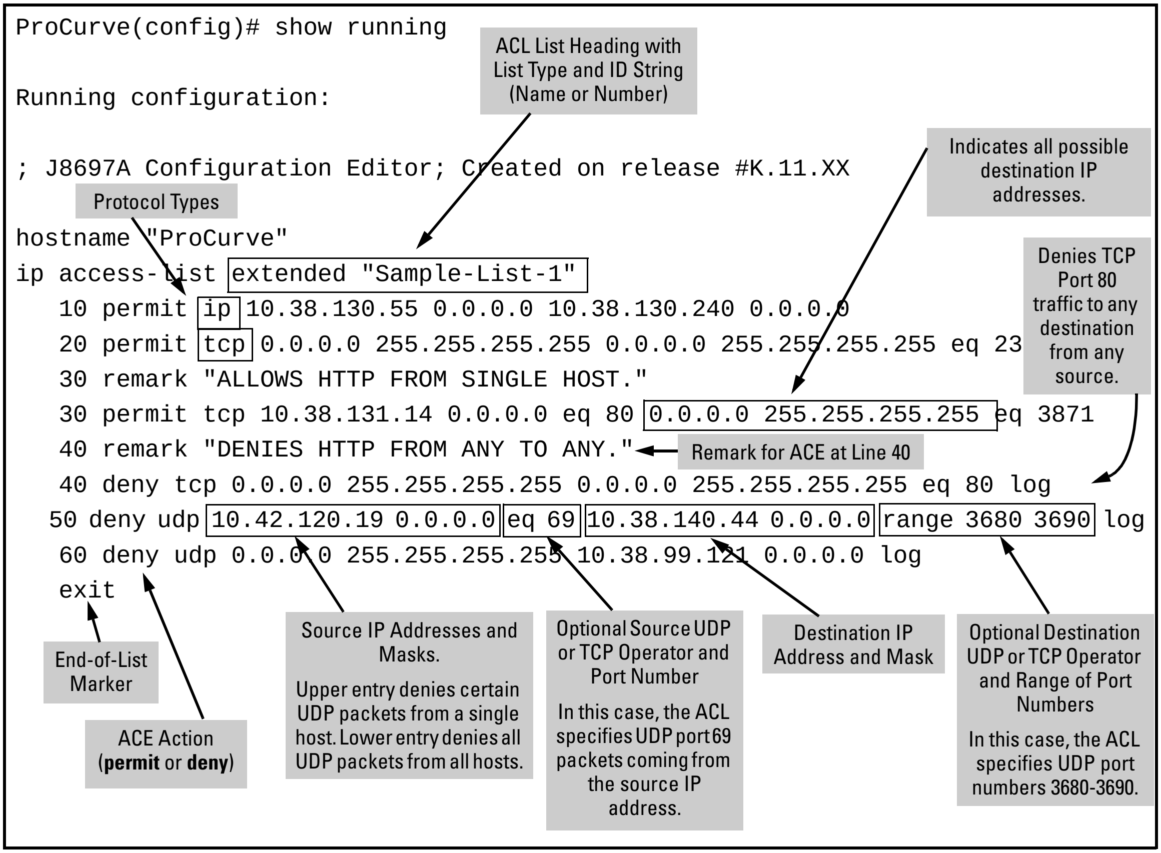


```
ip access-list extended < identifier >
[ [ seq-# ] remark < remark-str >]
< permit | deny > < ip-type > < SA > < src-acl-mask > < DA > <dest-acl-mask > [log]

      < permit | deny > tcp
         < SA > < src-acl-mask > [< operator > < port-id >]
         < DA > < desti-acl-mask > [< operator > < port-id >]   [log]
         [ established ]

      < permit | deny > udp
         < SA > < src-acl-mask > [< operator > < port-id >]
         < DA > < dest-acl-mask > [< operator > < port-id >]   [log]

      < permit | deny > icmp
         < SA > < src-acl-mask > < DA > < dest-acl-mask > [ icmp-type ]   [log]

      < permit | deny > igmp
         < SA > < SA-mask > < DA > < dest-acl-mask > [ igmp-type ]   [log]

      [ precedence < priority >]
      [ tos < tos-setting >]
      . . .
      < Implicit Deny >
      exit
```

Note: The optional **log** function appears only with "deny" ACEs.

Figure 7-8. Example of General Structure Options for an Extended ACL

For example, figure 7-9 shows how to interpret the entries in an extended ACL.



```
ProCurve(config)# show running

Running configuration:

; J8697A Configuration Editor; Created on release #K.11.XX

hostname "ProCurve"
ip access-list extended "Sample-List-1"
   10 permit ip 10.38.130.55 0.0.0.0 10.38.130.240 0.0.0.0
   20 permit tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 23
   30 remark "ALLOWS HTTP FROM SINGLE HOST."
   30 permit tcp 10.38.131.14 0.0.0.0 eq 80 0.0.0.0 255.255.255.255 eq 3871
   40 remark "DENIES HTTP FROM ANY TO ANY."
   40 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 80 log
  50 deny udp 10.42.120.19 0.0.0.0 eq 69 10.38.140.44 0.0.0.0 range 3680 3690 log
   60 deny udp 0.0.0.0 255.255.255.255 10.38.99.121 0.0.0.0 log
   exit
```

ACL List Heading with List Type and ID String (Name or Number)

Protocol Types

Indicates all possible destination IP addresses.

Denies TCP Port 80 traffic to any destination from any source.

Remark for ACE at Line 40

End-of-List Marker

ACE Action (**permit** or **deny**)

Source IP Addresses and Masks.

Upper entry denies certain UDP packets from a single host. Lower entry denies all UDP packets from all hosts.

Optional Source UDP or TCP Operator and Port Number

In this case, the ACL specifies UDP port 69 packets coming from the source IP address.

Destination IP Address and Mask

Optional Destination UDP or TCP Operator and Range of Port Numbers

In this case, the ACL specifies UDP port numbers 3680-3690.

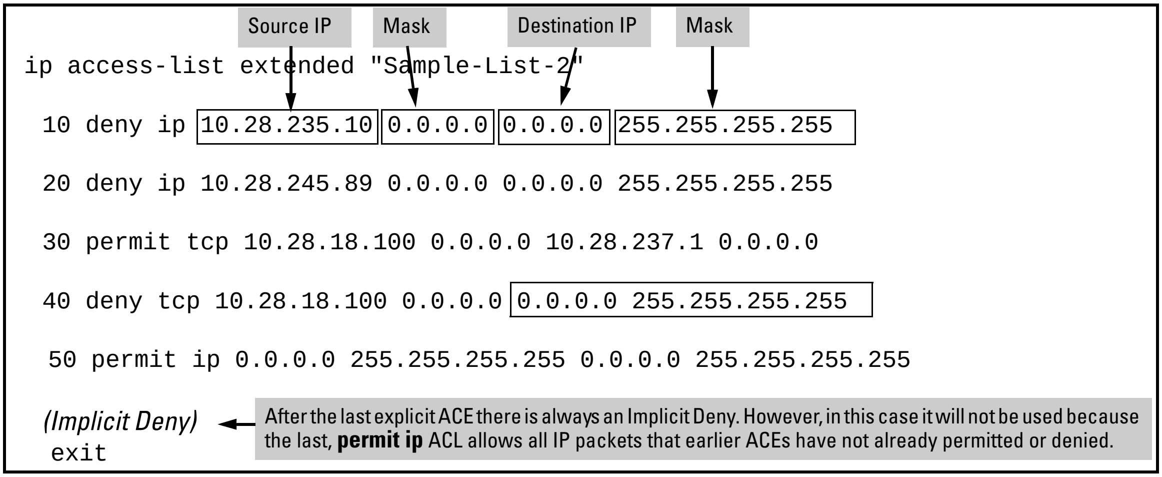
Figure 7-9. Example of a Displayed Extended ACL Configuration

ACL Configuration Factors

The Sequence of Entries in an ACL Is Significant

When the routing switch uses an ACL to determine whether to permit or deny a packet on a particular VLAN, it compares the packet to the criteria specified in the individual Access Control Entries (ACEs) in the ACL, beginning with the first ACE in the list and proceeding sequentially until a match is found. When a match is found, the routing switch applies the indicated action (permit or deny) to the packet. This is significant because, once a match is found for a packet, subsequent ACEs in the same ACL will not be used for that packet, regardless of whether they match the packet.

For example, suppose that you have applied the ACL shown in figure 7-10 to inbound traffic on VLAN 1 (the default VLAN):



```
ip access-list extended "Sample-List-2"
  10 deny ip 10.28.235.10 0.0.0.0 0.0.0.0 255.255.255.255
  20 deny ip 10.28.245.89 0.0.0.0 0.0.0.0 255.255.255.255
  30 permit tcp 10.28.18.100 0.0.0.0 10.28.237.1 0.0.0.0
  40 deny tcp 10.28.18.100 0.0.0.0 0.0.0.0 255.255.255.255
  50 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
  (Implicit Deny)
  exit
```

After the last explicit ACE there is always an Implicit Deny. However, in this case it will not be used because the last, **permit ip** ACL allows all IP packets that earlier ACEs have not already permitted or denied.

Figure 7-10. Example of a Standard ACL that Permits All Traffic Not Implicitly Denied

Table 7-7. Effect of the Above ACL on Inbound Traffic in the Assigned VLAN

Line #	Action
n/a	Shows list type (**extended**) and ID (Sample-List-2).
10	A packet from IP source address 10.28.235.10 will be denied (dropped). This ACE filters out all packets received from 10.28.235.10. As a result, IP traffic from that device will not be routed and packets from that device will not be compared against any later entries in the list.
20	A packet from IP source 10.28.245.89 will be denied (dropped). This ACE filters out all packets received from 10.28.245.89. As the result, IP traffic from that device will not be routed and packets from that device will not be compared against any later entries in the list.
30	A TCP packet from SA 10.28.18.100 with a DA of 10.28.237.1 will be permitted (forwarded). Since no earlier ACEes in the list have filtered TCP packets from 10.28.18.100 and destined for 10.28.237.1, the routing switch will use this ACE to evaluate such packets. Any packets that meet this criteria will be forwarded. (Any packets that do not meet this TCP source-destination criteria are not affected by this ACE.)
40	A TCP packet from source address 10.28.18.100 to ***any*** destination address will be denied (dropped). Since, in this example, the intent is to block TCP traffic from 10.28.18.100 to any destination ***except*** the destination stated in the ACE at line 30, this ACE must follow the ACE at line 30. (If their relative positions were exchanged, all TCP traffic from 10.28.18.100 would be dropped, including the traffic for the 10.28.18.1 destination.)
50	Any packet from any IP source address to any destination address will be permitted (forwarded). The only traffic to reach this ACE will be IP packets not specifically permitted or denied by the earlier ACEes.
n/a	The Implicit Deny is a function automatically added as the last action in all ACLs. It denies (drops) any IP traffic from any source to any destination that has not found a match with earlier entries in the ACL. In this example, the ACE at line 50 permits (forwards) any IP traffic not already permitted or denied by the earlier entries in the list, so there is no traffic remaining for action by the Implicit Deny function.
exit	Marks the end of the ACL.

Allowing for the Implied Deny Function

In any ACL having one or more ACEs there will always be a packet match. This is because the routing switch automatically applies an Implicit Deny as the last ACE in any ACL. This function is not visible in ACL listings, but is always present. (Refer to figure 7-10.) This means that if you configure the routing switch to use an ACL for filtering either inbound or outbound traffic on a VLAN, any packets not specifically permitted or denied by the explicit entries you create will be denied by the Implicit Deny action. If you want to preempt the Implicit Deny (so that traffic not specifically addressed by earlier ACEs in a given ACL will be permitted), insert an explicit **permit any** (for standard ACLs) or **permit ip any any** (for extended ACLs) as the last explicit ACE in the ACL.

A Configured ACL Has No Effect Until You Apply It to a VLAN Interface

The routing switch stores ACLs in the configuration file. Thus, until you actually assign an ACL to a VLAN interface, it is present in the configuration, but not used.

You Can Assign an ACL Name or Number to a VLAN Even if the ACL Does Not Exist in the Routing Switch's Configuration

In this case, if you subsequently create an ACL with that name or number, the routing switch automatically applies each ACE as soon as you enter it in the running-config file. Similarly, if you modify an existing ACE in an ACL you already applied to a VLAN, the routing switch automatically implements the new ACE as soon as you enter it. (See "General ACL Operating Notes" on page 7-99.) The routing switch allows a maximum of 2048 ACLs in any combination of numeric and alphanumeric names, and determines the total from the number of unique ACL names in the configuration. For example, if you configure two ACLs, but assign only one of them to a VLAN, the ACL total is two, for the two unique ACL names. If you then assign the name of a nonexistent ACL to a VLAN, the new ACL total is three, because the routing switch now has three unique ACL names in its configuration.

Using the CLI To Create an ACL

Command	Page
access-list (standard ACLs)	7-41
access-list (extended ACLs)	7-50

You can use either the routing switch CLI or an offline text editor to create an ACL. This section describes the CLI method, which is recommended for creating short ACLs. (To use the offline method, refer to "Creating or Editing ACLs Offline" on page 7-89.)

General ACE Rules

These rules apply to all ACEs you create or edit using the CLI:

- Inserting or adding an ACE to an ACL:
  - **Named ACLs:** Add an ACE to the end of a named ACE by using the **ip access-list** command to enter the Named ACL (**nacl**) context and entering the ACE without the sequence number. For example, if you wanted to add a "permit" ACL at the end of a list named "List-1" to allow traffic from the device at 10.10.10.100:

    ```
    ProCurve(config)# ip access-list standard List-1
    ProCurve(config-std-nacl)# permit host
    10.10.10.100
    ```

    Insert an ACE anywhere in a named ACL by specifying a sequence number. For example, if you wanted to insert a new ACE as line 15 between lines 10 and 20 in an existing ACL named "List-2" to deny traffic from the device at 10.10.10.77:

    ```
    ProCurve(config)# ip access-list standard List-2
    ProCurve(config-std-nacl)# 15 deny host 10.10.10.77
    ```

  - **Numbered ACLs:** Add an ACE to the end of a numbered ACL by using the **access-list < 1 - 99 | 100 - 199 >** command. For example, if you wanted to add a "permit" ACE at the end of a list identified with the number "11" to allow traffic from the device at 10.10.10.100:

    ```
    ProCurve(config)# access-list 11 permit host
    10.10.10.100
    ```

To insert an ACE *anywhere* in a numbered ACL, use the same process as described above for inserting an ACE anywhere in a *named* ACL. For example, to insert an ACE denying traffic from the host at 10.10.10.77 as line 52 in an existing ACL identified (named) with the number 11:

```
ProCurve(config)# ip access-list standard 99
ProCurve(config-std-nacl)# 52 deny host 10.10.10.77
```

Note

After a numbered ACL has been created (using **access-list < 1 - 99 | 100 - 199 >**), it can be managed as either a named or numbered ACL, as shown above.

- Deleting an ACE: Enter the ACL context and delete the sequence number for the unwanted ACE. (To view the sequence numbers of the ACEs in a list, use **show access-list <** ***acl-name-str*** **>**.)
- Duplicate ACEs are not allowed in the same ACL. Attempting to enter a duplicate ACE displays the **Duplicate access control entry** message.

Using CIDR Notation To Enter the ACL Mask

You can use CIDR (Classless Inter-Domain Routing) notation to enter ACL masks. The routing switch interprets the bits specified with CIDR notation as the IP address bits in an ACL and the corresponding IP address bits in a packet. The routing switch then converts the mask to inverse notation for ACL use.

Table 7-8. Examples of CIDR Notation for Masks

IP Address Used In an ACL with CIDR Notation	Resulting ACL Mask	Meaning
10.38.240.125/15	0.1.255.255	The leftmost 15 bits must match; the remaining bits are wildcards.
10.38.240.125/20	0.0.15.255	The leftmost 20 bits must match; the remaining bits are wildcards.
10.38.240.125/21	0.0.7.255	The leftmost 21 bits must match; the remaining bits are wildcards.
10.38.240.125/24	0.0.0.255	The leftmost 24 bits must match; the remaining bits are wildcards.
18.38.240.125/32	0.0.0.0	All bits must match.

Configuring Standard ACLs

Table 7-9. Command Summary for Standard ACLs

Action	Command(s)	Page
Create a Standard, **Named** ACL *or* Add an ACE to the End of an Existing Standard, **Named** ACL	ProCurve(config)# ip access-list standard < *name-str* > ProCurve(config-std-nacl)# < deny | permit > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] [log] [2]	7-43
Create a Standard, **Numbered** ACL or Add an ACE to the End of an Existing Standard, **Numbered** ACL	ProCurve(config)# access-list < 1-99 > < deny | permit > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >> [log][2]	7-46
Use a Sequence Number To Insert an ACE in an ACL	ProCurve(config)# ip access-list standard < *name-str* | 1-99 > ProCurve(config-std-nacl)# 1-2147483647 < deny | permit > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] [log] [2]	7-74
Use an ACE's Sequence Number To Delete the ACE from an ACL	ProCurve(config)# ip access-list standard < *name-str* | 1-99 > ProCurve(config-std-nacl)# no < 1-2147483647 >	7-77
Resequence the ACEs in an ACL	ProCurve(config)# ip access-list resequence < *name-str* | 1-99 > < 1-2147483647 > < 1-2147483646 >	7-78
Enter or Remove a Remark from an ACL	ProCurve(config)# ip access-list standard < *name-str* | 1-99 > ProCurve(config-ext-nacl)# [remark < *remark-str* > | no remark] *For numbered, standard ACLs only, the following* **remark** *commands can be substituted for the above:* ProCurve(config)# access-list < 1 - 99 > remark < remark-str > ProCurve(config)# [no] access-list < 1 - 99 > remark	7-79 7-81
Delete an ACL	ProCurve(config)# no ip access-list standard < *name-str* | 1-99 > *For numbered, standard ACLs, the following command can be substituted for the above:* ProCurve(config)# access-list < 1 - 99 > remark < remark-str >	7-72

[1] The mask can be in either dotted-decimal notation (such as 0.0.15.255) or CIDR notation (such as /20).
[2] The [log] function applies only to "deny" ACLs, and generates a message only when there is a "deny" match.

A standard ACL uses only source IP addresses in its ACEs. This type of ACE is useful when you need to:

- Permit or deny any IP traffic based on source IP address only.
- Quickly control the IP traffic from a specific address. This allows you to isolate traffic problems generated by a specific device, group of devices, or a subnet threatening to degrade network performance. This gives you an opportunity to troubleshoot without sacrificing performance for users outside of the problem area.

A *named*, standard ACL is identified by an alphanumeric string of up to 64 characters and is created by entering the Named ACL (**nacl**) context. A *numbered*, standard ACL is identified by a number in the range of 1 - 99 and is created without having to leave the global config context. Note that the CLI command syntax for creating a named ACL differs from the command syntax for creating a numbered ACL. For example, the first pair of entries below illustrate how to create (or enter) a named, standard ACL and enter an ACE. The next entry illustrates creating a numbered, standard ACL with the same ACE.

```
ProCurve(config)# ip access-list standard Test-List
ProCurve(config-std-nacl)# permit host 10.10.10.147
```

```
ProCurve(config)# ip access-list 1 permit host
10.10.10.17
```

Note that once a numbered ACL has been created, it can be accessed using the named ACL method. This is useful if it becomes necessary to edit a numbered ACL by inserting or removing individual ACEs. (Inserting or deleting an ACE is done by sequence number, and requires the Named ACL (**nacl**) context.) The routing switch allows a maximum of 2048 unique ACL identities; standard and extended combined.

Note

For a summary of standard ACL commands, refer to table 7-9 on page 7-41. For a summary of all ACL commands, refer to tables 7-1 and 7-2 on pages 7-5 and 7-6.

Configuring Named, Standard ACLs

This section describes the commands for performing the following:

- creating and/or entering the context of a named, standard ACL
- appending an ACE to the end of an existing list or entering the first ACE in a new list

For other ACL topics, refer to the following:

Topic	Page
configuring numbered, standard ACLs	7-46
configuring named, extended ACLs	7-52
configuring numbered, extended ACLs	7-64
applying or removing an ACL on a VLAN	7-71
deleting an ACL	7-72
editing an ACL	7-73
sequence numbering in ACLs	7-74
including remarks in an ACL	7-79
displaying ACL configuration data	7-83
creating or editing ACLs offline	7-89
enabling ACL "Deny" logging	7-94

Entering the "Named ACL" (nacl) Context. This command is a prerequisite to entering or editing ACEs in a named ACL.

Syntax: ip access-list standard < *name-str* >

*Places the CLI in the "Named ACL" (***nacl***) context specified by the < name-str > alphanumeric identifier. This enables entry of individual ACEs in the specified ACL. If the ACL does not already exist, this command creates it.*

< *name-str* >: *Specifies an alphanumeric identifier for the ACL. Consists of an alphanumeric string of up to 64 case-sensitive characters. Including spaces in the string requires that you enclose the string in single or double quotes. For example: "***Accounting ACL***".*

Refer also to table 7-9 on page 7-41.

Configuring ACEs in a Named, Standard ACL. Configuring ACEs is done after using the **ip access-list standard <** ***name-str*** **>** command described above to enter the "Named ACL" (nacl) context of an access list. *For a standard ACL syntax summary, refer to table 7-9 on page 7-41.*

Syntax: < deny | permit >
< any | host < *SA* > | *SA* <mask | *SA/ mask-length* >> [log]

Executing this command appends the ACE to the end of the list of ACEs in the current ACL. In the default ACL configuration, ACEs are automatically assigned consecutive sequence numbers in increments of 10 and can be renumbered using **resequence** *(page 7-78).*

Note: *To insert a new ACE between two existing ACEs, precede* **deny** *or* **permit** *with an appropriate sequence number. (Refer to "Inserting an ACE in an Existing ACL" on page 7-75.)*

< deny | permit >

*For named ACLs, used in the "Named ACL" (***nacl***) context to configure an ACE. Specifies whether the ACE denies or permits a packet matching the criteria in the ACE, as described below.*

< any | host < *SA* > | *SA* < *mask* > | *SA*/*mask-length* >

Defines the source IP address (SA) a packet must carry for a match with the ACE.

- **any** — *Allows IP packets from any SA.*
- **host <** ***SA*** **>** — *Specifies only packets having* **<** ***SA*** **>** *as the source. Use this criterion when you want to match the IP packets from a single source IP address.*
- ***SA*** **<** ***mask*** **>** *or* ***SA /mask-length*** — *Specifies packets received from either a subnet or a group of IP addresses. The mask format can be in either dotted-decimal format or CIDR format (number of significant bits). (Refer to "Using CIDR Notation To Enter the ACL Mask" on page 7-40).*

  Mask Application: *The mask is applied to the IP address in the ACL to define which bits in a packet's source IP address must exactly match the IP address configured in the ACL and which bits need not match. For example:* **10.10.10.1/24** *and* **10.10.10.1 0.0.0.255** *both define any IP address in the range of 10.10.10.(1 - 255).*

  Note: *Specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 7-26.*

[log]

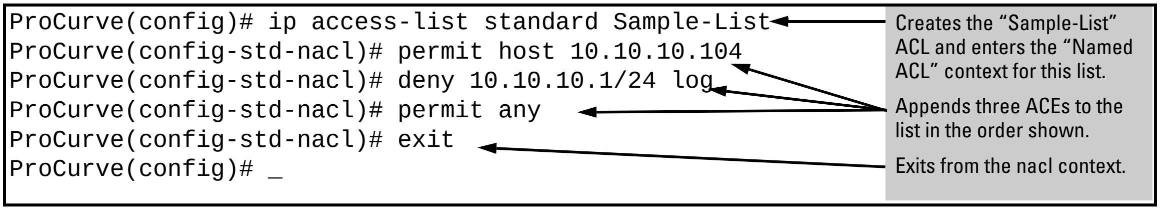
This option generates an ACL log message if:

- *The action is deny.*
- *There is a match.*
- *ACL logging is enabled on the routing switch. (Refer to "Enable ACL "Deny" Logging" on page 7-94.)*

(Use the debug command to direct ACL logging output to the current console session and/or to a Syslog server. Note that you must also use the **logging < *ip-addr* >** *command to specify the IP addresses of Syslog servers to which you want log messages sent. See also "Enable ACL "Deny" Logging" on page 7-94.)*

Example of Creating and Listing a Standard, Named ACL. This example illustrates how to create a standard, named ACL with several ACEs. This example creates an ACL that:

1. permits IP traffic from a host with the IP address of 10.10.10.104
2. creates another ACE that blocks all other traffic from the same subnet
3. allows all other IP traffic



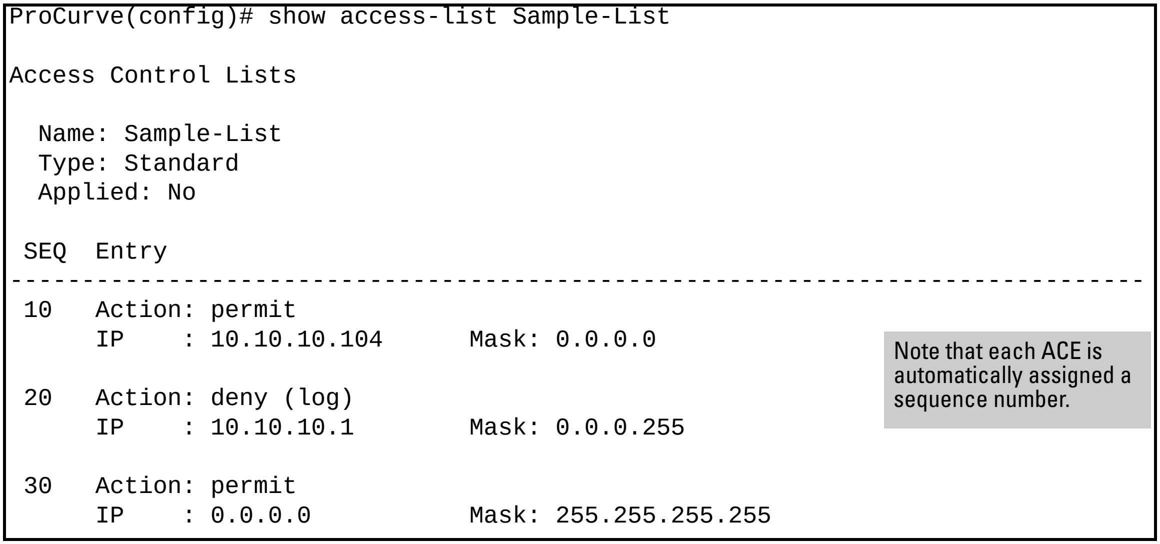
```
ProCurve(config)# ip access-list standard Sample-List
ProCurve(config-std-nacl)# permit host 10.10.10.104
ProCurve(config-std-nacl)# deny 10.10.10.1/24 log
ProCurve(config-std-nacl)# permit any
ProCurve(config-std-nacl)# exit
ProCurve(config)# _
```

Creates the "Sample-List" ACL and enters the "Named ACL" context for this list.

Appends three ACEs to the list in the order shown.

Exits from the nacl context.

Figure 7-11. Example of Commands Used To Create a Standard, Named ACL



```
ProCurve(config)# show access-list Sample-List

Access Control Lists

  Name: Sample-List
  Type: Standard
  Applied: No

 SEQ  Entry
-------------------------------------------------------------------------------
 10   Action: permit
      IP    : 10.10.10.104      Mask: 0.0.0.0

 20   Action: deny (log)
      IP    : 10.10.10.1        Mask: 0.0.0.255

 30   Action: permit
      IP    : 0.0.0.0           Mask: 255.255.255.255
```

Note that each ACE is automatically assigned a sequence number.

Figure 7-12. Screen Output Listing the "Sample-List" ACL Content

Creating Numbered, Standard ACLs

Use the following general steps to create or add to a numbered, standard ACL:

1. Create a numbered, standard ACL by entering the first ACE in the list
2. Append a new ACE to the end of an existing, standard ACL

This section describes the commands for performing these steps. For other ACL topics, refer to the following:

Topic	Page
configuring named, standard ACLs	7-43
configuring named, extended ACLs	7-52
configuring numbered, extended ACLs	7-64
applying or removing an ACL on a VLAN	7-71
deleting an ACL	7-72
editing an ACL	7-73
sequence numbering in ACLs	7-74
including remarks in an ACL	7-79
displaying ACL configuration data	7-83
creating or editing ACLs offline	7-89
enabling ACL "Deny" logging	7-94

Creating or Adding to a Standard, Numbered ACL. *This command is an alternative to using* **ip access-list standard < *name-str* >** *and does not use the "Named ACL" (***nacl***) context. For a standard ACL syntax summary, refer to table 7-9 on page 7-41.*

Syntax: access-list < 1-99 > < deny | permit >
< any | host < *SA* > | *SA* < *mask* | *SA*/*mask-length* >> [log]

Appends an ACE to the end of the list of ACEs in the current standard, numbered ACL. If the ACL does not already exist, creates both the ACL and its first ACE. In the default configuration, ACEs are automatically assigned consecutive sequence numbers in increments of 10 and can be renumbered using **resequence** *(page 7-78).*

Note: *To insert a new ACE between two existing ACEs in a standard, numbered ACL:*

a. *Use* **ip access list extended < 1 - 99 >** *to open the ACL as a named ACL.*

b. *Enter the desired sequence number along with the ACE keywords and variables you want.*

(After a numbered ACL has been created, it can be managed as either a named or numbered ACL. Refer to the "Numbered ACLs" list item on page 7-39.)

< 1-99 >

*Specifies the ACL identifier as a number. The routing switch interprets an ACL with a value in this range as a standard ACL (which filters all IP traffic on the basis of SA). (To create a standard access list with an alphanumeric name (***name-str***) instead of a number, refer to "Configuring Named, Standard ACLs" on page 7-43.)*

< deny | permit >

Specifies whether the ACE denies or permits a packet matching the criteria in the ACE, as described next.

< any | host < *SA* > | SA < *mask* | *SA*/*mask-length* >>

Defines the source IP address (SA) a packet must carry for a match with the ACE.

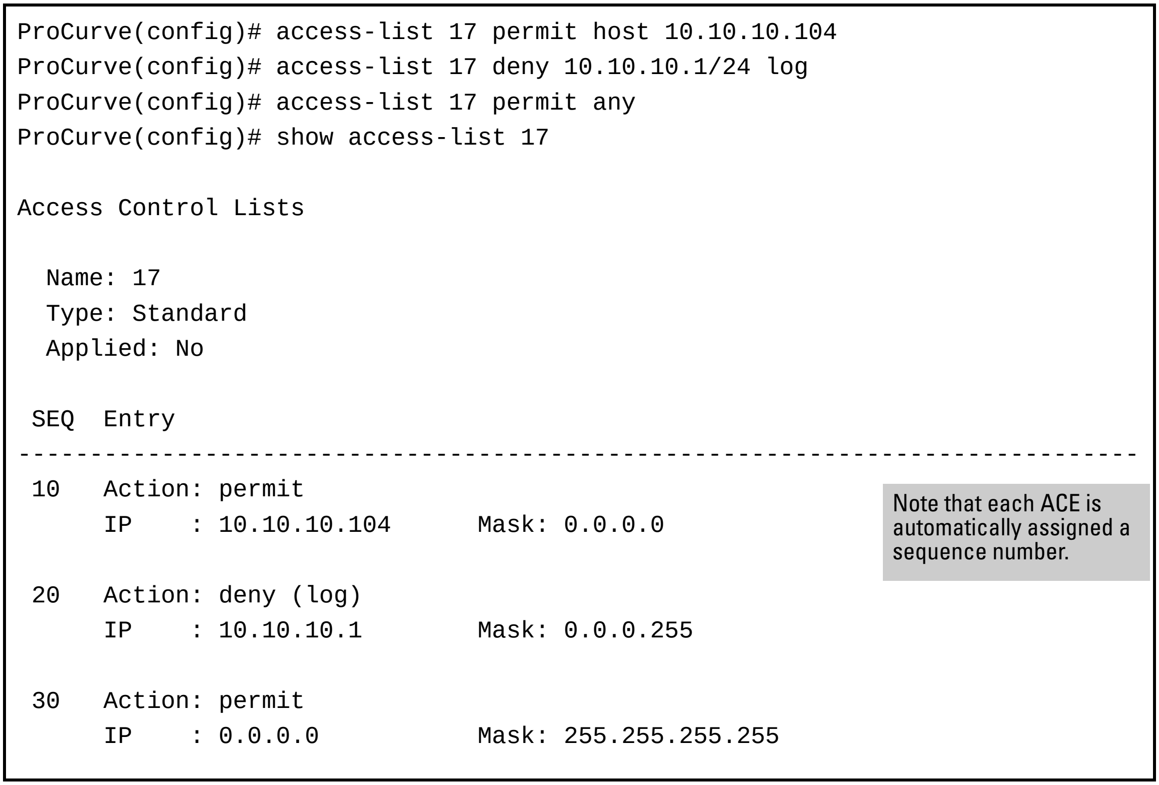
- **any** — *Allows IP packets from any SA.*
- **host <** ***SA*** **> —** *Specifies only packets having* **<** ***SA*** **>** *as the source. Use this criterion when you want to match only the IP packets from a single SA.*
- ***SA*** **<** ***mask*** **>** *or* ***SA*** **/*****mask-length*** **—** *Specifies packets received from an SA, where the SA is either a subnet or a group of IP addresses. The mask format can be in either dotted-decimal format or CIDR format (number of significant bits). (Refer to "Using CIDR Notation To Enter the ACL Mask" on page 7-40).*

SA Mask Application: *The mask is applied to the SA in the ACE to define which bits in a packet's SA must exactly match the SA configured in the ACL and which bits need not match.*

Example: **10.10.10.1/24** *and* **10.10.10.1 0.0.0.255** *both define any IP address in the range of 10.10.10.(1 - 255).*

Note: *Specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 7-26.*

Example of Creating and Viewing a Standard ACL. This example creates a standard, numbered ACL with the same ACE content as show in figure 7-11 on page 7-45.



```
ProCurve(config)# access-list 17 permit host 10.10.10.104
ProCurve(config)# access-list 17 deny 10.10.10.1/24 log
ProCurve(config)# access-list 17 permit any
ProCurve(config)# show access-list 17

Access Control Lists

  Name: 17
  Type: Standard
  Applied: No

 SEQ  Entry
------------------------------------------------------------------------------
 10   Action: permit
      IP    : 10.10.10.104      Mask: 0.0.0.0

 20   Action: deny (log)
      IP    : 10.10.10.1        Mask: 0.0.0.255

 30   Action: permit
      IP    : 0.0.0.0           Mask: 255.255.255.255
```

Note that each ACE is automatically assigned a sequence number.

Figure 7-13. Standard, Numbered ACL with the Same ACEs as the Standard, Named ACL in Figure 7-11

Configuring Extended ACLs

Table 7-10. Command Summary for Extended ACLs

Action	Command(s)	Page
Create an Extended, **Named** ACL *or* Add an ACE to the End of an Existing, Extended ACL	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-std-nacl)# < deny | permit > < ip | *ip-protocol* | *ip-protocol-nbr* > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] < any | host < *DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] [tcp | udp] < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] [*comparison-operator* < *value* >]] < any | host <*DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] [*comparison-operator* < *value* >] [established] < igmp > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] < any | host < *DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] [*igmp-packet-type*] < icmp > < any | host <*SA* > | *SA*/< mask-length > | *SA* < *mask* >>[1] < any | host < *DA* > | *DA*/< mask-length > | *DA* < *mask* >>[1] [[< 0 - 255 > [0 - 255]] | *icmp-message*] [precedence < *priority* >] [tos < *tos- setting* >] [log][2]	7-52
Create an Extended, **Numbered** ACL or Add an ACE to the End of an Existing, **Numbered** ACL	ProCurve(config)# access-list < 100-199 > < deny | permit > < *ip-options* |*tcp/udp-options* |*igmp-options* |*icmp-options* > [log][2] [precedence < *priority* >] [tos < *tos- setting* >] ***Note:*** *Uses the same IP, TCP/UDP, IGMP, and ICMP options as shown above for "Create an Extended, Named ACL".*	7-64
Insert an ACE by Assigning a Sequence Number	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-ext-nacl)# 1-2147483647 < deny | permit > *Uses the options shown above for "Create an Extended, Named ACL".*	7-75
Use Sequence Number To Delete an ACE	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-std-nacl)# no < 1-2147483647 >	7-77
Resequence the ACEs in an ACL	ProCurve(config)# ip access-list resequence < *name-str* | 100-199 > < 1-2147483647 > < 1-2147483646 >	7-78

[1] The mask can be in either dotted-decimal notation (such as 0.0.15.255) or CIDR notation (such as /20).
[2] The [log] function applies only to "deny" ACLs, and generates a message only when there is a "deny" match.

Table continues on the next page.

Action	Command(s)	Page
Enter or Remove a Remark	ProCurve(config)# ip access-list extended < *name-str* | 100-199 > ProCurve(config-ext-nacl)# [remark < *remark-str* > | no remark] *For numbered, extended ACLs only, the following* **remark** *commands can be substituted for the above:* ProCurve(config)# access-list < 100 - 199 > remark < remark-str > ProCurve(config)# [no] access-list < 100 - 199 > remark	7-79 7-81
Delete an Extended ACL	ProCurve(config)# no ip access-list extended < *name-str* | 100-199 > *For numbered, extended ACLs only, the following command can also be used:* ProCurve(config)# no access-list < 100 - 199 >	7-72

Standard ACLs use only source IP addresses for filtering criteria, extended ACLs use multiple filtering criteria. This enables you to more closely define your IP packet-filtering. Extended ACLs enable filtering on the following:

- Source and destination IP addresses (required), in one of the following options:
  - specific host IP
  - subnet or group of IP addresses
  - any IP address
- choice of any IP protocol
- optional packet-type criteria for IGMP, and ICMP traffic
- optional source and/or destination TCP or UDP port, with a further option for comparison operators and (for TCP) an option for establishing connections
- filtering for TCP traffic based on whether the subject traffic is initiating a connection ("established" option)
- optional IP precedence and ToS criteria

The routing switch allows up to 2048 ACLs in any combination of numeric and alphanumeric identifiers, and determines the total from the number of unique identifiers in the configuration. For example, configuring two ACLs results in an ACL total of two, even if neither is assigned to a VLAN. If you then assign a nonexistent ACL to a VLAN, the new ACL total is three, because the routing switch now has three unique ACL names in its configuration. (For more on ACL limits, refer to "Monitoring Shared Resources" on page 7-99.)

Configuring Named, Extended ACLs

For a match to occur with an ACE in an extended ACL, a packet must have the source and destination IP address criteria specified by the ACE, as well as any IP protocol-specific criteria included in the command.

Use the following general steps to create or add to a numbered, standard ACL:

1. Create and/or entering the context of a named, extended ACL.
2. Enter the first ACE in a new, extended ACL or append an ACE to the end of an existing, extended ACL.

This section describes the commands for performing these steps. For other ACL topics, refer to the following:

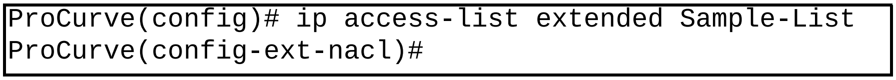
Topic	Page
configuring named, standard ACLs	7-43
configuring numbered, standard ACLs	7-46
configuring numbered, extended ACLs	7-64
applying or removing an ACL on a VLAN	7-71
deleting an ACL	7-72
editing an ACL	7-73
sequence numbering in ACLs	7-74
including remarks in an ACL	7-79
displaying ACL configuration data	7-83
creating or editing ACLs offline	7-89
enabling ACL “Deny” logging	7-94

Creating a Named, Extended ACL and/or Entering the "Named ACL" (nacl) Context. This command is a prerequisite to entering or editing ACEs in a named, extended ACL. (For a summary of the extended ACL syntax options, refer to table 7-10 on page 7-50.)

Syntax: ip access-list extended < *name-str* >

*Places the CLI in the "Named ACL" (***nacl***) context specified by the <* name-str *> alphanumeric identifier. This enables entry of individual ACEs in the specified ACL. If the ACL does not already exist, this command creates it.*

< ***name-str*** **>:** *Specifies an alphanumeric identifier for the ACL. Consists of an alphanumeric string of up to 64 case-sensitive characters. Including spaces in the string requires that you enclose the string in single or double quotes. For example: "***Accounting ACL***". You can also use this command to access an existing, numbered ACL. Refer to "Using the CLI To Edit ACLs" on page 7-73*



```
ProCurve(config)# ip access-list extended Sample-List
ProCurve(config-ext-nacl)#
```

Figure 7-14. Example of Entering the Named ACL Context

Configure ACEs in a Named, Extended ACL and/or Enter the "Named ACL" (nacl) Context. Configuring ACEs is done after using the **ip access-list standard < *name-str* >** command described on page 7-53to enter the "Named ACL" (**nacl**) context of an ACL. For an extended ACL syntax summary, refer to table 7-10 on page 7-50.

Syntax: (nacl context) < deny | permit > < ip | *ip-protocol* | *ip-protocol-nbr* >
< any | host < *SA* > | *SA/ mask-length* | *SA* < *mask* > >
< any | host < *DA* > | *DA/ mask-length* | *DA* < *mask* > >
[precedence] [tos] [log]

Appends an ACE to the end of the list of ACEs in the current ACL. In the default configuration, ACEs are automatically assigned consecutive sequence numbers in increments of 10 and can be renumbered using **resequence** *(page 7-78).*

Note: *To insert a new ACE between two existing ACEs in an extended, named ACL, precede* **deny** *or* **permit** *with an appropriate sequence number along with the ACE keywords and variables you want. (Refer to "Inserting an ACE in an Existing ACL" on page 7-75.)*

For a match to occur, a packet must have the source and destination IP addressing criteria specified in the ACE, as well as:

- *the protocol-specific criteria configured in the ACE, including any included, optional elements (described later in this section)*
- *any (optional) precedence and/or ToS settings configured in the ACE*

< deny | permit >

*For named ACLs, these keywords are used in the "Named ACL" (***nacl***) context to specify whether the ACE denies or permits a packet matching the criteria in the ACE, as described below.*

< ip | *ip-protocol* | *ip-protocol-nbr* >

Used after **deny** *or* **permit** *to specify the packet protocol type required for a match. An extended ACL must include one of the following:*

- **ip** *— any IP packet.*
- ***ip-protocol*** *— any one of the following IP protocol names:*

  ip-in-ip ipv6-in-ip gre esp ah
  ospf pim vrrp sctp tcp*
  udp* icmp* igmp*

- ***ip-protocol-nbr*** *— the IPV4 IP protocol number of an IP packet type, such as "8" for Exterior Gateway Protocol or 121 for Simple Message Protocol. (For a listing of IP protocol numbers and their corresponding protocol names, refer to the IANA "Protocol Number Assignment Services" at www.iana.com.) (Range: 0 - 255)*

** For TCP, UDP, ICMP, and IGMP, additional criteria can be specified, as described on pages 7-58 through 7-62.*

< any | host < *SA* > | *SA* < *mask* > | *SA/ mask-length*

This is the first instance of IP addressing in an extended ACE. It follows the protocol specifier and defines the source IP address (SA) a packet must carry for a match with the ACE.

- **any** *— Allows IP packets from any SA.*
- **host** < ***SA*** > *— Specifies only packets having* ***SA*** *as the SA. Use this criterion when you want to match only the IP packets from a single SA.*
- ***SA < mask >*** *or* ***SA/mask-length*** *— Specifies packets received from an SA, where the SA is either a subnet or a group of IP addresses. The mask can be in either dotted-decimal format or CIDR format (number of significant bits). Refer to "Using CIDR Notation To Enter the ACL Mask" on page 7-40.*

  SA Mask Application: *The mask is applied to the SA in the ACL to define which bits in a packet's SA must exactly match the SA configured in the ACL and which bits need not match.*

  Example: *10.10.10.1/24 and 10.10.10.1 0.0.0.255 both define any IP address in the range of 10.10.10.(1 - 255).*

  Note: *Specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 7-26.*

< any | host < *DA* > | *DA/mask*-length | *DA/* < *mask* >>

This is the second instance of IP addressing in an extended ACE. It follows the first (SA) instance, described earlier, and defines the destination IP address (DA) that a packet must carry in order to have a match with the ACE.

- **any** — *Allows routed IP packets to any DA.*
- **host** < ***DA*** > — *Specifies only packets having* ***DA*** *as the destination address. Use this criterion when you want to match only the IP packets for a single DA.*
- ***DA/mask-length*** *or* ***DA< mask >*** — *Specifies packets intended for a destination address, where the address is either a subnet or a group of IP addresses. The mask format can be in either dotted-decimal format or CIDR format (number of significant bits). Refer to "Using CIDR Notation To Enter the ACL Mask" on page 7-40.*

  DA Mask Application: *The mask is applied to the DA in the ACL to define which bits in a packet's DA must exactly match the DA configured in the ACL and which bits need not match. See also the above example and note.*

[precedence < 0 - 7 | *precedence-name* >]

This option can be used after the DA to cause the ACE to match packets with the specified IP precedence value. Values can be entered as the following IP precedence numbers or alphanumeric names:

0	*or*	routine
1	"	priority
2	"	immediate
3	"	flash
4	"	flash-override
5	"	critical
6	"	internet (for internetwork control)
7	"	network (for network control)

Note: *The precedence criteria described in this section are applied <u>in addition to</u> any other selection criteria configured in the same ACE.*

[tos < *tos-setting* >]

This option can be used after the DA to cause the ACE to match packets with the specified IP Type-of-Service (ToS) setting. ToS values can be entered as the following numeric settings or, in the case of 0, 2, 4, and 8, as alphanumeric names:

0	*or*	normal
2	“	max-reliability
4	“	max-throughput
6		
8	“	minimize-delay
10		
12		
14		

Note: *The ToS criteria in this section are applied in addition to any other criteria configured in the same ACE.*

[log]

This option can be used after the DA to generate an Event Log message if:

- *The action is* **deny**. *(Not applicable to* **permit**.*)*
- *There is a match.*
- *ACL logging is enabled. (Refer to “Enabling ACL Logging on the Routing Switch” on page 7-96.)*

Options for TCP and UDP Traffic in Extended ACLs. An ACE designed to permit or deny TCP or UDP traffic can optionally include port number criteria for either the source or destination, or both. Use of TCP criteria also allows the **established** option for controlling TCP connection traffic. (For a summary of the extended ACL syntax options, refer to table 7-10 on page 7-50.)

Syntax: < deny | permit > < tcp | udp >
< *SA* > [*comparison-operator* < *tcp/udp-src-port* >]
< *DA* >
[*comparison-operator* < *tcp-dest-port* >] [established]
[*comparison-operator* < *udp-dest-port* >]

In an extended ACL using either **tcp** *or* **udp** *as the IP packet protocol type, you can optionally use TCP or UDP source and/ or destination port numbers or ranges of numbers to further define the criteria for a match. For example:*

```
#deny tcp host 10.20.10.17 eq 23 host 10.20.10.155
 established
#permit tcp host 10.10.10.100 host 10.20.10.17
 eq telnet
#deny udp 10.30.10.1/24 host 10.20.10.17 range
 161 162
```

[*comparison-operator* < *tcp/udp-src-port* >]

To specify a TCP or UDP <u>*source*</u> *port number in an ACE, (1) select a comparison operator from the following list and (2) enter the port number or a well-known port name.*

Comparison Operators:

- **eq** < ***tcp/udp-port-nbr*** > — *"Equal To"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be equal to <* ***tcp/udp-port-nbr*** *>.*
- **gt** < ***tcp/udp-port-nbr*** > — *"Greater Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be greater than <* ***tcp/udp-port-nbr*** *>.*
- **lt** < ***tcp/udp-port-nbr*** > — *"Less Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be less than <* ***tcp/udp-port-nbr*** *>.*
- **neq** < ***tcp/udp-port-nbr*** > — *"Not Equal"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must not be equal to* ***< tcp/udp-port-nbr >****.*
- **range** ***< start-port-nbr > < end-port-nbr >*** *— For a match with the ACE entry, the TCP or UDP source-port number in a packet must be in the range <****start-port-nbr >*** ***< end-port-nbr >****.*

Port Number or Well-Known Port Name:

Use the TCP or UDP port number required by your application. The routing switch also accepts these well-known TCP or UDP port names as an alternative to their port numbers:

- ***TCP***: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet
- ***UDP***: bootpc, bootps, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To list the above names, press the **[Shift] [?]** *key combination after entering an operator. For a comprehensive listing of port numbers, visit www.iana.org/assignments/port-numbers.*

[*comparison-operator* < *tcp-dest-port* >] [established]
[*comparison-operator* < *udp-dest-port* >]

This option, if used, is entered immediately after the **< *DA* >** *entry. To specify a TCP or UDP port number, (1) select a comparison operator and (2) enter the port number or a well-known port name.*

Comparison Operators and Well-Known Port Names — *These are the same as are used with the TCP/UDP source-port options, and are listed earlier in this command description.*

[established] *— This option applies only where TCP is the configured IP protocol type. It blocks the synchronizing packet associated with establishing a TCP connection in one direction on a VLAN while allowing all other traffic for the same type of connection in the opposite direction. For example, a Telnet connect requires TCP traffic to move both ways between a host and the target device. Simply applying a Deny to inbound Telnet traffic on a VLAN would prevent Telnet sessions in either direction because responses to outbound requests would be blocked. However, by using the* **established** *option, inbound Telnet traffic arriving in response to outbound Telnet requests would be permitted, but inbound Telnet traffic trying to establish a connection would be denied.*

Options for ICMP Traffic in Extended ACLs. This option is useful where it is necessary to permit some types of ICMP traffic and deny other types, instead of simply permitting or denying all types of ICMP traffic. That is, an ACE designed to permit or deny ICMP traffic can optionally include an ICMP type and code value to permit or deny an individual type of ICMP packet while not addressing other ICMP traffic types in the same ACE. As an optional alternative, the ACE can include the name of an ICMP packet type. (For a summary of the extended ACL syntax options, refer to table 7-10 on page 7-50.)

Syntax: < deny | permit > icmp < *SA* > < *DA* > [*icmp-type* [*icmp-code*]
< deny | permit > icmp < *SA* > < *DA* > [*icmp-type-name*]

[ ] |]

In an extended ACL using **icmp** *as the packet protocol type (see above), you can optionally specify an individual ICMP packet type or packet type/code pair to further define the criteria for a match. This option, if used, is entered immediately after the destination IP address (DA) entry. The following example shows two ACEs entered in a Named ACL context:*

```
#permit icmp any any host-unknown
#permit icmp any any 3 7
```

[*icmp-type* [*icmp-code*]]

This option identifies an individual ICMP packet type as criteria for permitting or denying that type of ICMP traffic in an ACE.

- **icmp-type —** *This value is in the range of 0 - 255 and corresponds to an ICMP packet type.*
- **icmp-code —** *This value is in the range of 0 - 255 and corresponds to an ICMP code for an ICMP packet type.*

For more information on ICMP type names, visit the Internet Assigned Numbers Authority (IANA) website at **www.iana.com,** *click on "Protocol Number Assignment Services", and then go to the selections under "Internet Control Message Protocol (ICMP) Parameters".*

[*icmp-type-name*]

These name options are an alternative to the **[icmp-type [icmp-code]]** *methodology described above. For more information, visit the IANA website cited above.*

administratively-prohibited
alternate-address
conversion-error
dod-host-prohibited
dod-net-prohibited
echo
echo-reply
general-parameter-problem
host-isolated
host-precedence-unreachable
host-redirect
host-tos-redirect
host-tos-unreachable
host-unknown
host-unreachable
information-reply
information-request
mask-reply
mask-request
mobile-redirect
net-redirect
net-tos-redirect
net-tos-unreachable
net-unreachable
network-unknown
no-room-for-option
option-missing
packet-too-big
parameter-problem
port-unreachable
precedence-unreachable
protocol-unreachable
reassembly-timeout
redirect
router-advertisement
router-solicitation
source-quench
source-route-failed
time-exceeded
timestamp-reply
timestamp-request
traceroute
ttl-exceeded
unreachable

Option for IGMP in Extended ACLs. This option is useful where it is necessary to permit some types of IGMP traffic and deny other types instead of simply permitting or denying all types of IGMP traffic. That is, an ACE designed to permit or deny IGMP traffic can optionally include an IGMP packet type to permit or deny an individual type of IGMP packet while not addressing other IGMP traffic types in the same ACE. (For a summary of the extended ACL syntax options, refer to table 7-10 on page 7-50.)

Syntax: < permit | deny > igmp < *SA* > < *DA* > [*igmp*-type]

In an extended ACL using **igmp** *as the packet protocol type, you can optionally specify an individual IGMP packet type to further define the criteria for a match. This option, if used, is entered immediately after the destination IP addressing entry. The following example shows an IGMP ACE entered in the Named ACL context:*

```
ProCurve(config-ext-nacl)# permit igmp any
   any host-query
```

[*igmp*-type]

The complete list of IGMP packet-type options includes:

dvmrp
host-query
host-report
pim
trace
v2-host-report
v2-host-leave
mtrace-reply
mtrace-request
v3-host-report

For more information on IGMP packet types, visit the Internet Assigned Numbers Authority (IANA) website at **www.iana.com,** *click on "Protocol Number Assignment Services", and then go to the selections under "Internet Group Management Protocol (IGMP) Type Numbers".*

Example of a Named, Extended ACL. Suppose that you want to implement these policies on a routing switch configured for IP routing and membership in VLANs 10, 20, and 30:

A. Permit Telnet traffic from 10.10.10.44 to 10.10.20.78, deny all other IP traffic from network 10.10.10.0 (VLAN 10) to 10.10.20.0 (VLAN 20), and permit all other IP traffic from any source to any destination. (See "A" in figure 7-15, below.)

B. Permit FTP traffic from IP address 10.10.20.100 (on VLAN 20) to 10.10.30.55 (on VLAN 30). Deny FTP traffic from other hosts on network10.10.20.0 to any destination, but permit all other traffic.

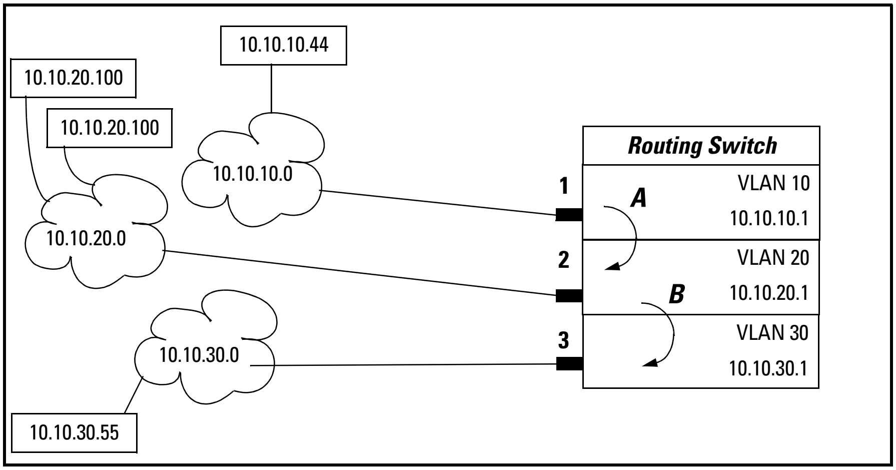
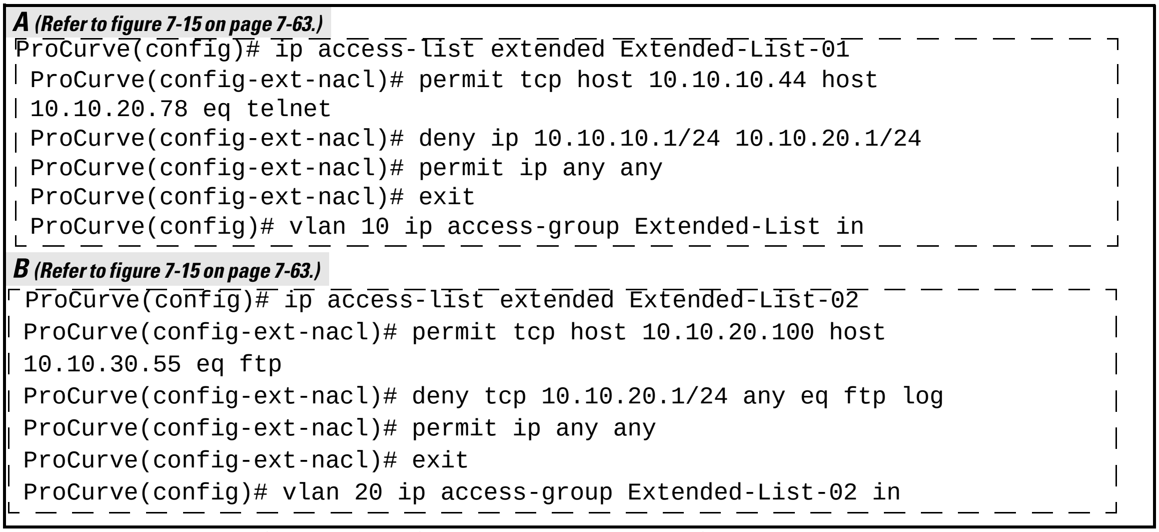


Figure 7-15. Example of an Extended ACL



A *(Refer to figure 7-15 on page 7-63.)*

```
ProCurve(config)# ip access-list extended Extended-List-01
 ProCurve(config-ext-nacl)# permit tcp host 10.10.10.44 host
 10.10.20.78 eq telnet
 ProCurve(config-ext-nacl)# deny ip 10.10.10.1/24 10.10.20.1/24
 ProCurve(config-ext-nacl)# permit ip any any
 ProCurve(config-ext-nacl)# exit
 ProCurve(config)# vlan 10 ip access-group Extended-List in
```

B *(Refer to figure 7-15 on page 7-63.)*

```
 ProCurve(config)# ip access-list extended Extended-List-02
 ProCurve(config-ext-nacl)# permit tcp host 10.10.20.100 host
 10.10.30.55 eq ftp
 ProCurve(config-ext-nacl)# deny tcp 10.10.20.1/24 any eq ftp log
 ProCurve(config-ext-nacl)# permit ip any any
 ProCurve(config-ext-nacl)# exit
 ProCurve(config)# vlan 20 ip access-group Extended-List-02 in
```

Figure 7-16. Example of Configuration Commands for Extended ACLs

Configuring Numbered, Extended ACLs

This section describes the commands for performing the following in a numbered, extended ACL:

- creating the ACL by entering the first ACE in the list
- appending a new ACE to the end of an existing ACL

For other ACL topics, refer to the following:

Topic	Page
configuring named, standard ACLs	7-43
configuring numbered, standard ACLs	7-46
configuring named, extended ACLs	7-52
applying or removing an ACL on a VLAN	7-71
deleting an ACL	7-72
editing an ACL	7-73
sequence numbering in ACLs	7-74
including remarks in an ACL	7-79
displaying ACL configuration data	7-83
creating or editing ACLs offline	7-89
enabling ACL “Deny” logging	7-94

Creating or Adding to an Extended, Numbered ACL. This command is an alternative to using **ip access-list extended < *name-str* >** and does not use the Named ACL (**nacl**) context. (For an extended ACL syntax summary, refer to table 7-10 on page 7-50.)

Syntax: access-list < 100-199 > < deny | permit > < ip | *ip-protocol* | *ip-protocol-nbr* >
< any | host < *SA* > | *SA*/*mask-length* | SA < *mask* >>
< any | host < *DA* > | *DA*/mask-length | DA < mask >>
[precedence < 0 - 7 | *precedence-name* >]
[tos < *tos-bit-setting* >
[log]

If the ACL does not already exist, this command creates the specified ACL and its first ACE. If the ACL already exists, the new ACE is appended to the end of the configured list of explicit ACEs. In the default configuration, the ACEs in an ACL will automatically be assigned consecutive sequence numbers in increments of 10 and can be renumbered with **resequence** *(page 7-78).*

Note: *To insert a new ACE between two existing ACEs in an extended, numbered ACL:*

a. Use **ip access list extended < 100 - 199 >** *to open the ACL as a named ACL.*

b. Enter the desired sequence number along with the ACE statement you want.

(Refer to the "Numbered ACLs" list item on page 7-39.)

For a match to occur, a packet must have the source and destination IP addressing criteria specified in the ACE, as well as:

- *the protocol-specific criteria configured in the ACE, including any included, optional elements (described later in this section)*
- *any (optional) precedence and/or ToS settings configured in the ACE*

< 100-199 >

Specifies the ACL ID number. The routing switch interprets a numeric ACL with a value in this range as an extended ACL.

< deny | permit >

*Specifies whether to deny (***drop***) or permit (forward) a packet that matches the criteria specified in the ACE, as described below.*

< ip | *ip-protocol* | *ip-protocol-nbr* >

Specifies the packet protocol type required for a match. An extended ACL must include one of the following:

- **ip** — *any IP packet.*
- ***ip-protocol*** — *any one of the following IP protocol names:*

ip-in-ip	**ipv6-in-ip**	**gre**	**esp**	**ah**
ospf	**pim**	**vrrp**	**sctp**	**tcp***
udp*	**icmp***	**igmp***		

- ***ip-protocol-nbr*** — *the IPv4 IP protocol number of an IP packet type, such as "8" for Exterior Gateway Protocol or 121 for Simple Message Protocol. (For a listing of IP protocol numbers and their corresponding protocol names, refer to the IANA "Protocol Number Assignment Services" at www.iana.com.) (Range: 0 - 255)*

** For TCP, UDP, ICMP, and IGMP, additional criteria can be specified, as described later in this section.*

< any | host < *SA* > | *SA/mask-length* **/** *SA* < *mask* >>

In an extended ACL, this parameter defines the source IP address (SA) that a packet must carry in order to have a match with the ACE.

- **any** — *Specifies all inbound IP packets.*
- **host** < ***SA*** > — *Specifies only inbound packets from a single IP address. Use this option when you want to match only the IP packets from one source IP address.*
- ***SA/mask-length*** *or* ***SA < mask >*** — *Specifies packets received from an SA, where the SA is either a subnet or a group of IP addresses. The mask can be in either dotted-decimal format or CIDR format with the number of significant bits. Refer to "Using CIDR Notation To Enter the ACL Mask" on page 7-40.*

SA Mask Application: *The mask is applied to the SA in the ACL to define which bits in a packet's source SA must exactly match the IP address configured in the ACL and which bits need not match.*

Example: *10.10.10.1/24 and 10.10.10.1 0.0.0.255 both define any IP address in the range of 10.10.10.(1-255).*

Note: *Specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 7-26.*

< any | host < *DA* > | *DA/mask*-length >

This is the second instance of IP addressing in an extended ACE. It follows the first (SA) instance, described earlier, and defines the destination IP address (DA) that a packet must carry in order to have a match with the ACE. The options are the same as shown for < SA *>.*

- **any** — *Allows routed IP packets to any DA.*
- **host** < ***DA*** > — *Specifies only packets having* ***DA*** *as the destination IP address. Use this criterion when you want to match only the IP packets for a single DA.*
- ***DA/mask-length*** *or* ***DA < mask >*** — *Specifies packets intended for a destination address, where the address is either a subnet or a group of IP addresses. The mask format can be in either dotted-decimal format or CIDR format (number of significant bits). Refer to "Using CIDR Notation To Enter the ACL Mask" on page 7-40.*

  DA Mask Application: *The mask is applied to the DA in the ACL to define which bits in a packet's DA must exactly match the DA configured in the ACL and which bits need not match. See also the above example and note.*

[precedence < 0 - 7 | *precedence-name* >]

This option causes the ACE to match packets with the specified IP precedence value. Values can be entered as the following IP precedence numbers or alphanumeric names:

0	*or*	routine
1	“	priority
2	“	immediate
3	“	flash
4	“	flash-override
5	“	critical
6	“	internet (for internetwork control)
7	“	network (for network control)

***Note:** The precedence criteria described in this section are applied in addition to any other selection criteria configured in the same ACE.*

[tos]

This option can be used after the DA to cause the ACE to match packets with the specified IP Type-of-Service (ToS) setting. ToS values can be entered as the following numeric settings or, in the case of 0, 2, 4, and 8, as alphanumeric names:

0	*or*	normal
2	“	max-reliability
4	“	max-throughput
6		
8	“	minimize-delay
10		
12		
14		

***Note:** The ToS criteria in this section are applied in addition to any other criteria configured in the same ACE.*

[log]

Optional; generates an Event Log message if:

- *The action is* **deny.** *(This option is not configurable for Permit.)*
- *There is a match.*
- *ACL logging is enabled on the routing switch. (Refer to “Enabling ACL Logging on the Routing Switch” on page 7-96)*

Additional Options for TCP and UDP Traffic. An ACE designed to permit or deny TCP or UDP traffic can optionally include port number criteria for either the source or destination, or both. Use of TCP criteria also allows the **established** option for controlling TCP connection traffic. (For a summary of the extended ACL syntax options, refer to table 7-10 on page 7-50.)

Syntax: access-list < 100 - 199 > < deny | permit > < tcp | udp >

< *SA* > [*comparison-operator* < *tcp/udp-src-port* >]

< *DA* > [*comparison-operator* < *tcp-dest-port* >] [established]
< *DA* > [*comparison-operator* < *udp-dest-port* >]

This source-port and destination-port TCP/UDP criteria is identical to the criteria described for TCP/UDP use in named, extended ACLs, beginning on page 7-58.

Additional Options for ICMP Traffic. This option is useful where it is necessary to permit some types of ICMP traffic and deny other types, instead of simply permitting or denying all types of ICMP traffic. That is, an ACE designed to permit or deny ICMP traffic can optionally include an ICMP type and code value to permit or deny an individual type of ICMP packet while not addressing other ICMP traffic types in the same ACE. As an optional alternative, the ACE can include the name of an ICMP packet type. (For a summary of the extended ACL syntax options, refer to table 7-10 on page 7-50.)

Syntax: access-list < 100 - 199 > < deny | permit > icmp < *SA* > < *DA* >

[[*icmp-type* [*icmp-code*]] | [*icmp-type-name*]]

The ICMP "type" and "code" criteria are identical to the criteria described for ICMP in named, extended ACLs, beginning on page 7-60.

Additional Option for IGMP. This option is useful where it is necessary to permit some types of IGMP traffic and deny other types, instead of simply permitting or denying all types of IGMP traffic. That is, an ACE designed to permit or deny IGMP traffic can optionally include an IGMP packet type to permit or deny an individual type of IGMP packet while not addressing other IGMP traffic types in the same ACE. (For a summary of the extended ACL syntax options, refer to table 7-10 on page 7-50.)

Syntax: access-list < 100 - 199 >
< deny | permit > igmp < *src-ip* > < *dest-ip* > [*igmp-type*]

The IGMP "type" criteria is identical to the criteria described for IGMP in named, extended ACLs, beginning on page 7-62.

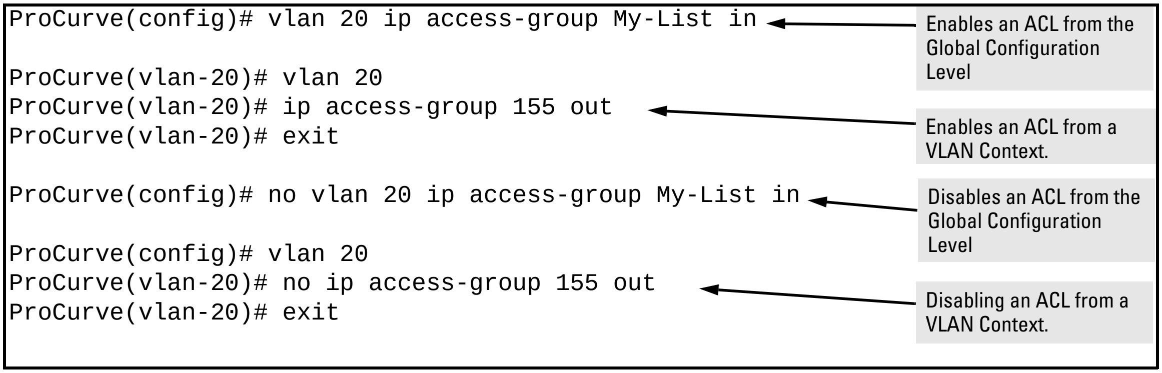
Adding or Removing an ACL Assignment On a VLAN

For a given interface, you can assign one ACL to filter inbound traffic and one ACL to filter outbound traffic. You can also use the same ACL for both inbound and outbound traffic, and for assignment to multiple VLANs. For limits and operating rules, refer to "ACL Configuration and Operating Rules" on page 7-23.

Syntax: [no] vlan < *vid* > ip access-group < *ascii-string* > < in | out >
where: < *ascii-string* > = either a ACL name or an ACL ID number.

Assigns an ACL to a VLAN. You can use either the global configuration level or the VLAN context level to assign an ACL to a VLAN or remove an ACL from a VLAN.

Note: *The routing switch allows you to assign a nonexistent ACL name or number to a VLAN. In this case, if you subsequently configure an ACL with that name or number, it will automatically become active on the assigned VLAN. Also, if you delete an assigned ACL from the routing switch without subsequently using the "***no***" form of this command to remove the assignment to a VLAN, the ACL assignment remains and will automatically activate any new ACL you create with the same identifier (name or number).*



```
ProCurve(config)# vlan 20 ip access-group My-List in

ProCurve(vlan-20)# vlan 20
ProCurve(vlan-20)# ip access-group 155 out
ProCurve(vlan-20)# exit

ProCurve(config)# no vlan 20 ip access-group My-List in

ProCurve(config)# vlan 20
ProCurve(vlan-20)# no ip access-group 155 out
ProCurve(vlan-20)# exit
```

Enables an ACL from the Global Configuration Level

Enables an ACL from a VLAN Context.

Disables an ACL from the Global Configuration Level

Disabling an ACL from a VLAN Context.

Figure 7-17. Methods for Enabling and Disabling ACLs

Deleting an ACL

Syntax: no ip access-list standard < *name-str* | 1-99 >

no ip access-list extended < *name-str* | 100-199 >

no access-list < 1 - 99 | 100 - 199 >

Removes the specified ACL from the routing switch's running-config file.

Note: *Deleting an ACL does not delete any assignment of that ACL's identifier on a specific VLAN. Creating a new ACL using an identifier that is already configured on a VLAN causes the routing switch to automatically activate that ACL. If you need to remove an ACL identifier assignment on a VLAN, refer to "Adding or Removing an ACL Assignment On a VLAN" on page 7-71*

Editing an Existing ACL

The CLI provides the capability for editing in the routing switch by using sequence numbers to insert or delete individual ACEs. An offline method is also available. This section describes using the CLI for editing ACLs. To use the offline method for editing ACLs, refer to "Creating or Editing ACLs Offline" on page 7-89.

Using the CLI To Edit ACLs

You can use the CLI to delete individual ACEs from anywhere in an ACL, append new ACEs to the end of an ACL, and insert new ACEs anywhere within an ACL.

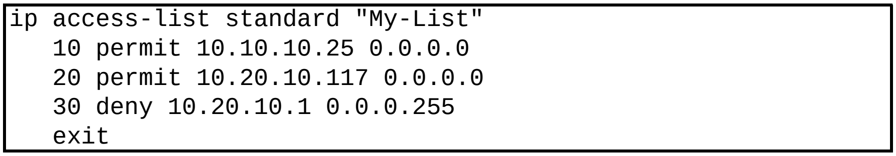
General Editing Rules

- **Named ACLs:**
  - When you enter a new ACE in a named ACL without specifying a sequence number, the routing switch inserts the ACE as the last entry in the ACL.
  - When you enter a new ACE in a named ACL and include a sequence number, the routing switch inserts the ACE according to the position of the sequence number in the current list of ACEs.
- **Numbered ACLs:** When using the **access-list < 1 - 99 | 100 - 199 >** command to create or add to a numbered ACL, each new ACE you enter is added to the end of the current list. (This command does not offer a **<** ***seq-#*** **>** option for including a sequence number to enable inserting an ACE at other points in the list.) Note, however, that once a numbered list has been created, you have the option of accessing it in the same way as a named list by using the **ip access-list < standard | extended >** command. This enables you to edit a numbered list in the same way that you would edit a named list. (See the next item in this list.)
- You can delete any ACE from any ACL (named or numbered) by using the **ip access-list** command to enter the ACL's context, and then using the **no <** ***seq-#*** **>** command (page 7-77).

- Deleting the last ACE from an ACL leaves the ACL in memory. In this case, the ACL is “empty” and cannot perform any filtering tasks. (In any ACL the Implicit Deny does not apply unless the ACL includes at least one explicit ACE.)
- When you create a new ACL, the routing switch inserts it as the last ACL in the startup-config file. (Executing **write memory** saves the running-config file to the startup-config file.)

Sequence Numbering in ACLs

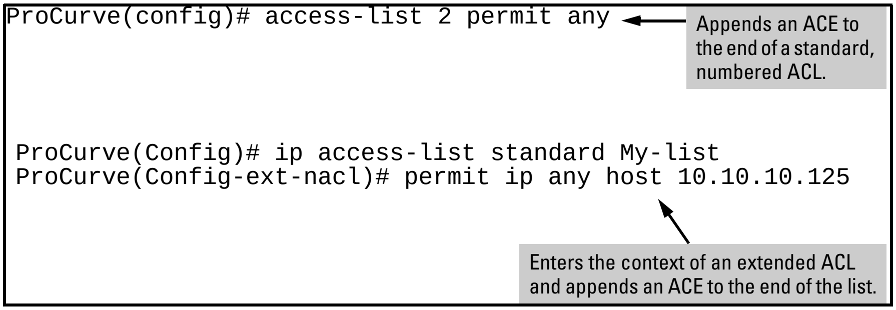
The ACEs in any ACL are sequentially numbered. In the default state, the sequence number of the first ACE in a list is “10” and subsequent ACEs are numbered in increments of 10. For example, the following show run output lists three ACEs with default numbering in a list named “My-List”:



```
ip access-list standard "My-List"
   10 permit 10.10.10.25 0.0.0.0
   20 permit 10.20.10.117 0.0.0.0
   30 deny 10.20.10.1 0.0.0.255
   exit
```

Figure 7-18. Example of the Default Sequential Numbering for ACEs

You can add an ACE to the end of a named or numbered ACL by using either **access-list** for numbered ACLs or **ip access-list** for named ACLs:



```
ProCurve(config)# access-list 2 permit any

ProCurve(Config)# ip access-list standard My-list
ProCurve(Config-ext-nacl)# permit ip any host 10.10.10.125
```

Appends an ACE to the end of a standard, numbered ACL.

Enters the context of an extended ACL and appends an ACE to the end of the list.

Figure 7-19. Examples of Adding an ACE to the end of Numbered or Named ACLs

For example, to append a fourth ACE to the end of the ACL in figure 7-18:

```
ProCurve(config)# ip access-list standard My-List
ProCurve(config-std-nacl)# permit any
ProCurve(config-std-nacl)# show run
.
.
.
ip access-list standard "My-List"
   10 permit 10.10.10.25 0.0.0.0
   20 permit 10.20.10.117 0.0.0.0
   30 deny 10.20.10.1 0.0.0.255
   40 permit 0.0.0.0 255.255.255.255
   exit
```

Figure 7-20. Example of Appending an ACE to an Existing List

Note

When using the **access-list < 1 - 99 | 100 - 199 > < permit | deny > <** ***SA*** **>** command to create an ACE for a numbered ACL, the ACE is always added to the end of the current list and given the appropriate sequence number. However, once a numbered list has been created, you can use the **ip access-list** command to open it as a named ACL and specify a nondefault sequence number, as described in the next section.

Inserting an ACE in an Existing ACL

This action uses a sequence number to specify where to insert a new ACE into an existing sequence of ACLs.

Syntax: ip access-list < standard | extended > < *name-str* | 1 - 99 | 100 - 199 >

<1-2147483647> < permit | deny > < *standard-acl-ip-criteria* > [log]
<1-2147483647> < permit | deny > < *extended-acl-ip-criteria* > [*options*]

The first command enters the "Named-ACL" context for the specified ACL. The remaining two commands insert a new ACE in a standard or extended ACL, respectively. (For details on these criteria and options, refer to table 7-1, "Command Summary for Standard ACLs" —page 7-5, and table 7-2, "Command Summary for Extended ACLs" —page 7-6.)

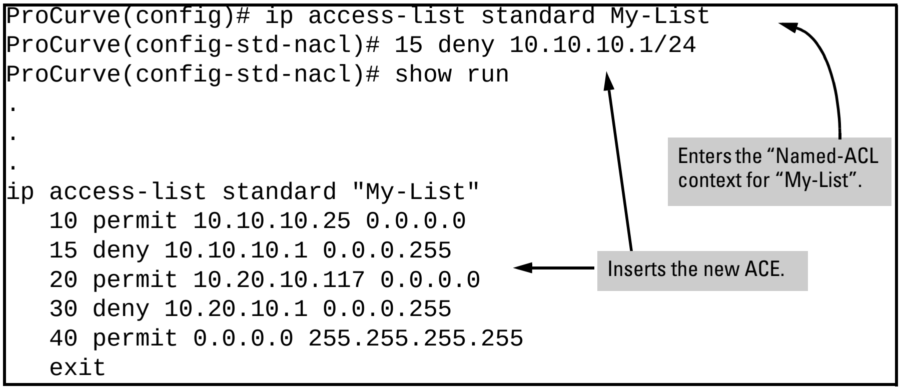
To insert a new ACE between existing ACEs in a list:

1. Use **ip access-list** to enter the "Named-ACL" (**nacl**) context of the ACE. *This applies regardless of whether the ACE was originally created as a numbered ACL or a named ACL.*

2. Begin the ACE command with a sequence number that identifies the position you want the ACE to occupy. (The sequence number range is 1-2147483647.)

3. Complete the ACE with the command syntax appropriate for the type of ACL you are editing.

For example, inserting a new ACE between the ACEs numbered 10 and 20 in figure 7-20 requires a sequence number in the range of 11-19 for the new ACE.



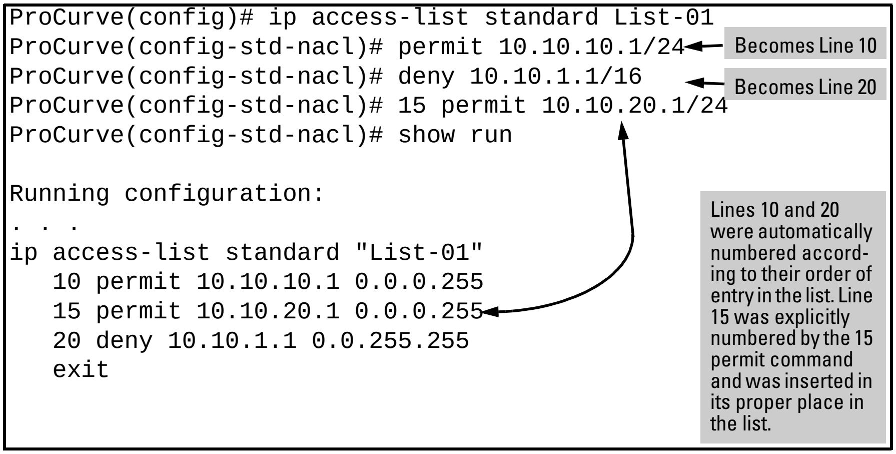
```
ProCurve(config)# ip access-list standard My-List
ProCurve(config-std-nacl)# 15 deny 10.10.10.1/24
ProCurve(config-std-nacl)# show run
.
.
.
ip access-list standard "My-List"
   10 permit 10.10.10.25 0.0.0.0
   15 deny 10.10.10.1 0.0.0.255
   20 permit 10.20.10.117 0.0.0.0
   30 deny 10.20.10.1 0.0.0.255
   40 permit 0.0.0.0 255.255.255.255
   exit
```

Enters the “Named-ACL context for “My-List”.

Inserts the new ACE.

Figure 7-21. Example of Inserting an ACE in an Existing ACL

In the following example, the first two ACEs entered become lines 10 and 20 in the list. The third ACE entered is configured with a sequence number of 15 and is inserted between lines 10 and 20.



```
ProCurve(config)# ip access-list standard List-01
ProCurve(config-std-nacl)# permit 10.10.10.1/24
ProCurve(config-std-nacl)# deny 10.10.1.1/16
ProCurve(config-std-nacl)# 15 permit 10.10.20.1/24
ProCurve(config-std-nacl)# show run

Running configuration:
. . .
ip access-list standard "List-01"
   10 permit 10.10.10.1 0.0.0.255
   15 permit 10.10.20.1 0.0.0.255
   20 deny 10.10.1.1 0.0.255.255
   exit
```

Becomes Line 10

Becomes Line 20

Lines 10 and 20 were automatically numbered according to their order of entry in the list. Line 15 was explicitly numbered by the 15 permit command and was inserted in its proper place in the list.

Figure 7-22. Example of Inserting an ACE into an Existing Sequence

Deleting an ACE from an Existing ACL

This action uses ACL sequence numbers to delete ACEs from an ACL.

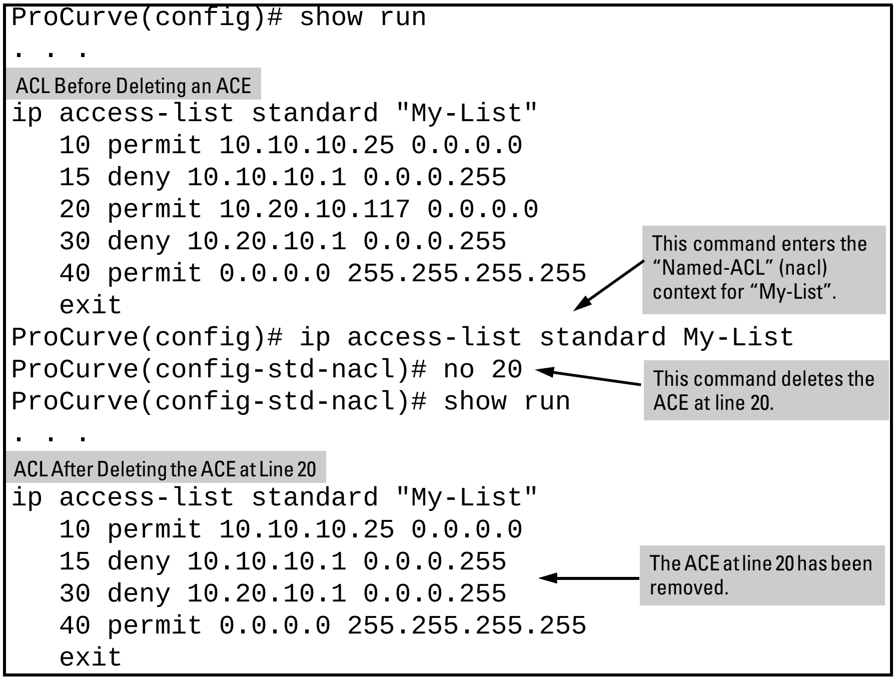
Syntax: ip access-list < standard | extended > < *name-str* | 1 - 99 | 100 - 199 >

no < *seq-#* >

The first command enters the "Named-ACL" context for the specified ACL. The **no** *command deletes the ACE corresponding to the sequence number entered. (Range: 1 - 2147483647)*

1. To find the sequence number of the ACE you want to delete, use **show run** or **show access-list <** ***name-str*** **| 1 - 99 | 100-199 >** to view the ACL.
2. Use **ip access-list** to enter the "Named-ACL" (**nacl**) context of the ACE. *This applies regardless of whether the ACE was originally created as a numbered ACL or a named ACL.*
3. In the "Named-ACL" context, type **no** and enter the sequence number of the ACE you want to delete.

Figure 7-23 illustrates the process for deleting an ACE from a list:



```
ProCurve(config)# show run
. . .
ACL Before Deleting an ACE
ip access-list standard "My-List"
   10 permit 10.10.10.25 0.0.0.0
   15 deny 10.10.10.1 0.0.0.255
   20 permit 10.20.10.117 0.0.0.0
   30 deny 10.20.10.1 0.0.0.255
   40 permit 0.0.0.0 255.255.255.255
   exit
ProCurve(config)# ip access-list standard My-List
ProCurve(config-std-nacl)# no 20
ProCurve(config-std-nacl)# show run
. . .
ACL After Deleting the ACE at Line 20
ip access-list standard "My-List"
   10 permit 10.10.10.25 0.0.0.0
   15 deny 10.10.10.1 0.0.0.255
   30 deny 10.20.10.1 0.0.0.255
   40 permit 0.0.0.0 255.255.255.255
   exit
```

This command enters the "Named-ACL" (nacl) context for "My-List".

This command deletes the ACE at line 20.

The ACE at line 20 has been removed.

Figure 7-23. Example of Deleting an ACE from Any ACL

Resequencing the ACEs in an ACL

This action reconfigures the starting sequence number for ACEs in an ACL, and resets the numeric interval between sequence numbers for ACEs configured in the ACL.

Syntax: ip access-list resequence < name-str | 1 - 99 | 100 - 199 > < *starting-seq-#* > < *interval* >

Resets the sequence numbers for all ACEs in the ACL.
< *starting-seq-#* > : *Specifies the sequence number for the first ACE in the list. (Default: 10; Range: 1 - 2147483647)*

< *interval* > : *Specifies the interval between sequence numbers for the ACEs in the list. (Default: 10; Range: 1 - 2147483647)*

1. To view the current sequence numbering in an ACE, use **show run** or **show access-list < *name-str* | 1 - 99 | 100-199 >**.
2. Use the command syntax (above) to change the sequence numbering.

This example resequences the “My-List” ACL at the bottom of figure 7-23 so that the list begins with line 100 and uses a sequence interval of 100.

```
ProCurve(config)# show run
. . .
ip access-list standard "My-List"
   10 permit 10.10.10.25 0.0.0.0
   15 deny 10.10.10.1 0.0.0.255
   30 deny 10.20.10.1 0.0.0.255
   40 permit 0.0.0.0 255.255.255.255
   exit
. . .
ProCurve(config)# ip access-list resequence My-List 100 100
ProCurve(config)# show run
. . .
ip access-list standard "My-List"
   100 permit 10.10.10.25 0.0.0.0
   200 deny 10.10.10.1 0.0.0.255
   300 deny 10.20.10.1 0.0.0.255
   400 permit 0.0.0.0 255.255.255.255
   exit
```

Figure 7-24. Example of Viewing and Resequencing an ACL

Attaching a Remark to an ACE

A remark is numbered in the same way as an ACE, and uses the same sequence number as the ACE to which it refers. This operation requires that the remark for a given ACE be entered prior to entering the ACE itself.

Syntax: access-list < 1 - 99 | 100 - 199 > remark < *remark-str* >

This syntax appends a remark to the end of a numbered ACL and automatically assigns a sequence number to the remark. The next command entry should be the ACE to which the remark belongs. (The new ACE will automatically be numbered with the same sequence number as was used for the preceding remark.

Syntax: ip access-list < standard | extended > < *name-str* | 1-99 | 100-199 >
[*seq-#*] remark < *remark-str* >
no < *seq-#* > remark

This syntax applies to both named and numbered ACLs. Without an optional sequence number, the remark is appended to the end of the list and automatically assigned a sequence number. When entered with an optional sequence number, the remark is inserted in the list according to the numeric precedence of the sequence number. The no form of the command deletes the indicated remark, but does not affect the related ACE.

To associate a remark with a specific ACE, enter the remark first, and then enter the ACE.

- *Entering a remark without a sequence number and then entering an ACE without a sequence number results in the two entries being automatically paired with the same sequence number and appended to the end of the current ACL.*
- *Entering a remark with a sequence number and then entering an ACE with the same sequence number results in the two entries being paired together and positioned in the list according to the sequence number they share.*

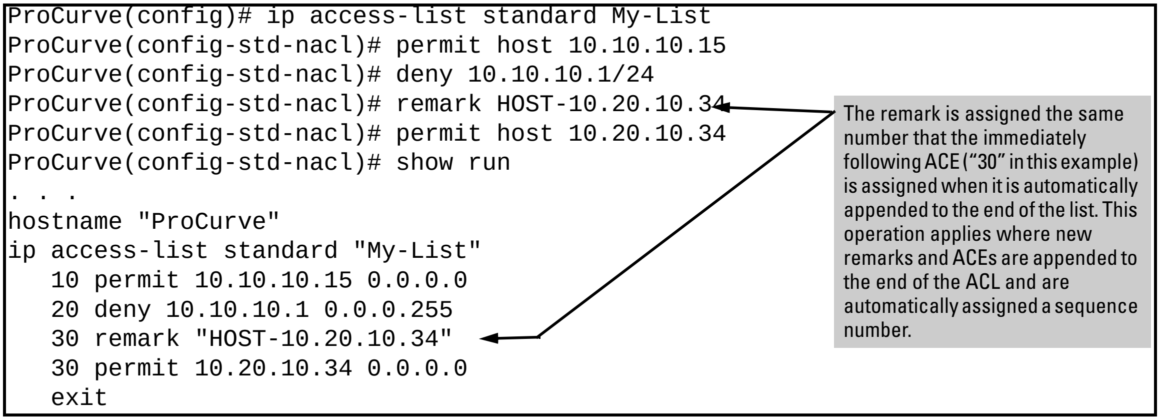
Note

After a numbered ACL has been created (using **access-list < 1 - 99 | 100 - 199 >**), it can be managed as either a named or numbered ACL. For example, in an existing ACL with a numeric identifier of "115", either of the following command sets adds an ACE denying IP traffic from any IP source to a host at 10.10.10.100:

```
ProCurve(config)# access-list 115 deny ip host
10.10.10.100
```

```
ProCurve(config)# ip access-list extended 115
ProCurve(config-ext-nacl)# deny ip any 10.10.10.100
```

Appending Remarks and Related ACEs to the End of an ACL. To include a remark for an ACE that will be appended to the end of the current ACL, enter the remark first, then enter the related ACE. This results in the remark and the subsequent ACE having the same sequence number. For example, to add remarks using the "Named-ACL" (**nacl**) context:



```
ProCurve(config)# ip access-list standard My-List
ProCurve(config-std-nacl)# permit host 10.10.10.15
ProCurve(config-std-nacl)# deny 10.10.10.1/24
ProCurve(config-std-nacl)# remark HOST-10.20.10.34
ProCurve(config-std-nacl)# permit host 10.20.10.34
ProCurve(config-std-nacl)# show run
. . .
hostname "ProCurve"
ip access-list standard "My-List"
   10 permit 10.10.10.15 0.0.0.0
   20 deny 10.10.10.1 0.0.0.255
   30 remark "HOST-10.20.10.34"
   30 permit 10.20.10.34 0.0.0.0
   exit
```

The remark is assigned the same number that the immediately following ACE ("30" in this example) is assigned when it is automatically appended to the end of the list. This operation applies where new remarks and ACEs are appended to the end of the ACL and are automatically assigned a sequence number.

Figure 7-25. Example of Appending a Remark and Its Related ACE to the End of an ACL

(You can also perform the operation illustrated in figure 7-25 by using the numbered, **access-list < 1 - 99 | 100 - 199 >** syntax shown at the beginning of this section.)

Inserting Remarks and Related ACEs Within an Existing List. To insert an ACE with a remark within an ACL by specifying a sequence number, insert the numbered remark first, then, using the same sequence number, insert the ACE. (This operation applies only to ACLs accessed using the "Named-ACL" (**nacl**) context.) For example:

```
ProCurve(config-std-nacl)# 15 remark "HOST 10.10.10.21"
ProCurve(config-std-nacl)# 15 permit host 10.10.10.21
ProCurve(config-std-nacl)# show run
ip access-list standard "My-List"
   10 permit 10.10.10.15 0.0.0.0
   15 remark "HOST 10.10.10.21"
   15 permit 10.10.10.21 0.0.0.0
   20 deny 10.10.10.1 0.0.0.255
   30 remark "HOST-10.20.10.34"
   30 permit 10.20.10.34 0.0.0.0
   exit
```

Inserting a remark/ACE pair with the same sequence number requires that the remark (with the desired sequence number) be inserted **before** the ACE with the same number.

Inserting a Remark for an ACE that Already Exists in an ACL. If a sequence number is already assigned to an ACE in a list, you cannot insert a remark by assigning it to the same number. (To configure a remark with the same number as a given ACE, the remark must be configured first.) To assign a remark to the same number as an existing ACE:

1. Delete the ACE.
2. Configure the remark with the number you want assigned to the pair.
3. Re-Enter the deleted ACE with the number used to enter the remark.

Removing a Remark from an Existing ACE. If you want to remove a remark, but want to retain the ACE, do the following:

1. Use the Named ACL context to enter the ACL.
2. Note the sequence number and content of the ACE having a remark you want to remove.
3. Delete the ACE.
4. Using the same sequence number, re-enter the ACE.

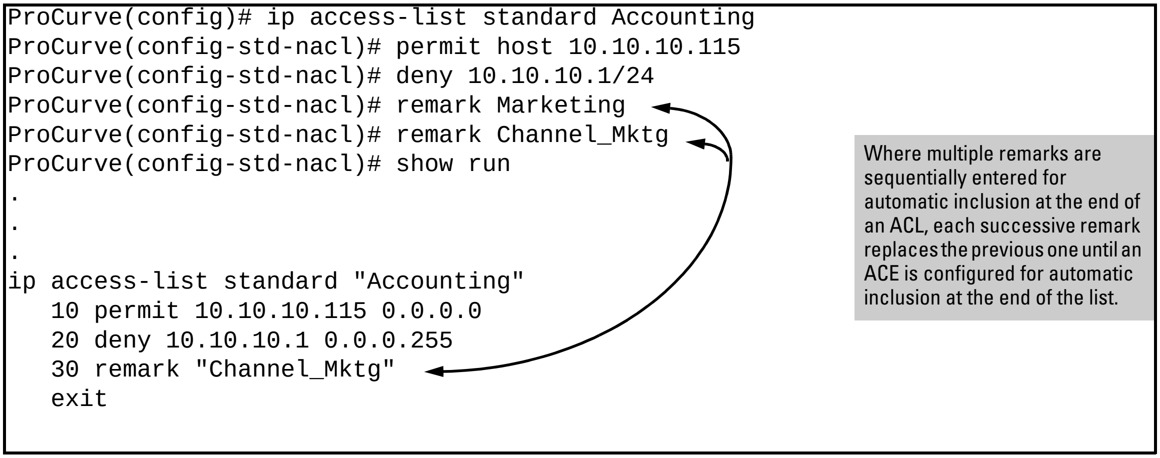
Operating Notes for Remarks

- The **resequence** command ignores “orphan” remarks that do not have an ACE counterpart with the same sequence number. For example, if:
  - a remark numbered “55” exists in an ACE
  - there is no ACE numbered “55” in the same ACL
  - **resequence** is executed on an ACL

  then the remark retains “55” as its sequence number and will be placed in the renumbered version of the ACL according to that sequence number.

- Entering an unnumbered remark followed by a numbered ACE, or the reverse, creates an “orphan” remark. The unnumbered entry will be assigned a sequence number that is an increment from the last ACE in the list. The numbered entry will then be placed sequentially in the list according to the sequence number used.

- Configuring two remarks without either sequence numbers or an intervening, unnumbered ACE results in the second remark overwriting the first.



```
ProCurve(config)# ip access-list standard Accounting
ProCurve(config-std-nacl)# permit host 10.10.10.115
ProCurve(config-std-nacl)# deny 10.10.10.1/24
ProCurve(config-std-nacl)# remark Marketing
ProCurve(config-std-nacl)# remark Channel_Mktg
ProCurve(config-std-nacl)# show run
.
.
.
ip access-list standard "Accounting"
   10 permit 10.10.10.115 0.0.0.0
   20 deny 10.10.10.1 0.0.0.255
   30 remark "Channel_Mktg"
   exit
```

Where multiple remarks are sequentially entered for automatic inclusion at the end of an ACL, each successive remark replaces the previous one until an ACE is configured for automatic inclusion at the end of the list.

Figure 7-26. Example of Overwriting One Remark with Another

Displaying ACL Configuration Data

ACL Commands	Function	Page
show access-list	Displays a brief listing of all ACLs on the routing switch.	7-83
show access-list config	Display the type, identifier, and content of all ACLs configured in the routing switch.	7-84
show access-list vlan < *vid* >	List the name and type of ACLs assigned to a particular VLAN on the routing switch.	7-85
show access-list < *acl-id* >	Display detailed content information for a specific ACL.	7-86
show config show running	**show config** includes configured ACLs and assignments existing in the startup-config file. **show running** includes configured ACLs and assignments existing in the running-config file.	

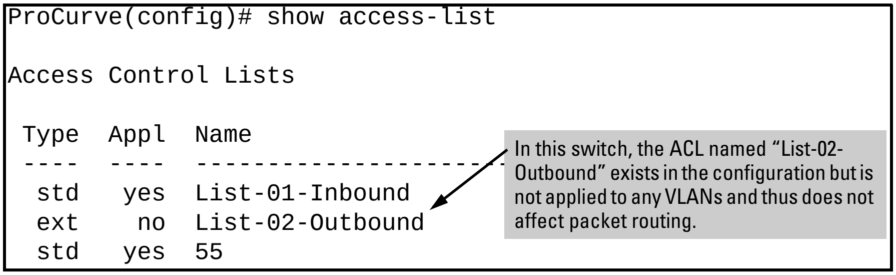
Display an ACL Summary

This command lists the configured ACLs, regardless of whether they are assigned to any VLANs.

Syntax: show access-list

List a summary table of the name, type, and application status of all ACLs configured on the routing switch.

For example:



```
ProCurve(config)# show access-list

Access Control Lists

 Type  Appl  Name
 ----  ----  --------------------
  std   yes  List-01-Inbound
  ext    no  List-02-Outbound
  std   yes  55
```

In this switch, the ACL named "List-02-Outbound" exists in the configuration but is not applied to any VLANs and thus does not affect packet routing.

Figure 7-27. Example of a Summary Table of Access lists

Term	Meaning
Type	Shows whether the listed ACL is **std** (Standard; source-address only) or **ext** (Extended; protocol, source, and destination data).
Appl	Shows whether the listed ACL has been applied to a VLAN (**yes**/**no**).
Name	Shows the identifier (name or number) assigned to each ACL configured in the routing switch.

Display the Content of All ACLs on the Routing Switch

This command lists the configuration details for every ACL in the running-config file, regardless of whether any are actually assigned to filter traffic on specific VLANs.

Syntax: show access-list config

List the configured syntax for all ACLs currently configured on the routing switch.

Note

Notice that you can use the output from this command for input to an offline text file in which you can edit, add, or delete ACL commands. Refer to “Creating or Editing ACLs Offline” on page 7-89.

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, with two ACLs configured in the routing switch, you will see results similar to the following:

```
ProCurve(config)# show access-list config

ip access-list standard "List-43"
   10 deny 10.28.236.77 0.0.0.0
   20 deny 10.29.140.107 0.0.0.0
   30 permit 0.0.0.0 255.255.255.255
   exit
ip access-list extended "111"
   10 permit tcp 10.30.133.27 0.0.0.0 0.0.0.0 255.255.255.255
   20 permit tcp 10.30.155.101 0.0.0.0 0.0.0.0 255.255.255.255
   30 deny ip 10.30.133.1 0.0.0.0 0.0.0.0 255.255.255.255 log
   40 deny ip 10.30.155.1 0.0.0.255 0.0.0.0 255.255.255.255
   exit
```

Figure 7-28. Example of an ACL Configured Syntax Listing

Display the ACL Assignments for a VLAN

This command briefly lists the identification and type(s) of ACLs currently assigned to a particular VLAN in the running-config file. (The routing switch allows up to two ACL assignments per VLAN; one inbound and one outbound.)

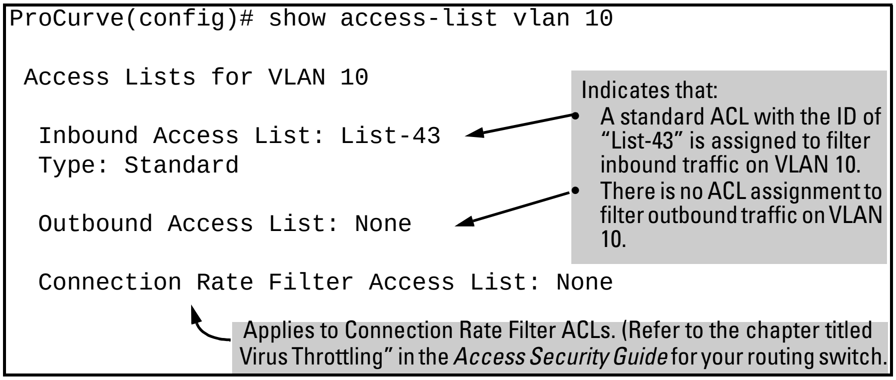
Syntax: show access-list vlan < *vid* >

List the ACLs assigned to a VLAN in the running config file.

Note

This information also appears in the **show running** display. If you execute **write memory** after configuring an ACL, it also appears in the **show config** display.

For example, if you assigned a standard ACL with an ACL-ID of "List-43" to filter inbound traffic on VLAN 10, you could quickly verify this assignment as follows:



```
ProCurve(config)# show access-list vlan 10

 Access Lists for VLAN 10

  Inbound Access List: List-43
  Type: Standard

  Outbound Access List: None

  Connection Rate Filter Access List: None
```

Indicates that:
- A standard ACL with the ID of "List-43" is assigned to filter inbound traffic on VLAN 10.
- There is no ACL assignment to filter outbound traffic on VLAN 10.

Applies to Connection Rate Filter ACLs. (Refer to the chapter titled Virus Throttling" in the *Access Security Guide* for your routing switch.

Figure 7-29. Example of Listing the ACL Assignments for a VLAN

Displaying the Content of a Specific ACL

This command displays a specific ACL configured in the running config file in an easy-to-read tabular format.

Note

This information also appears in the **show running** display. If you execute **write memory** after configuring an ACL, it also appears in the **show config** display.

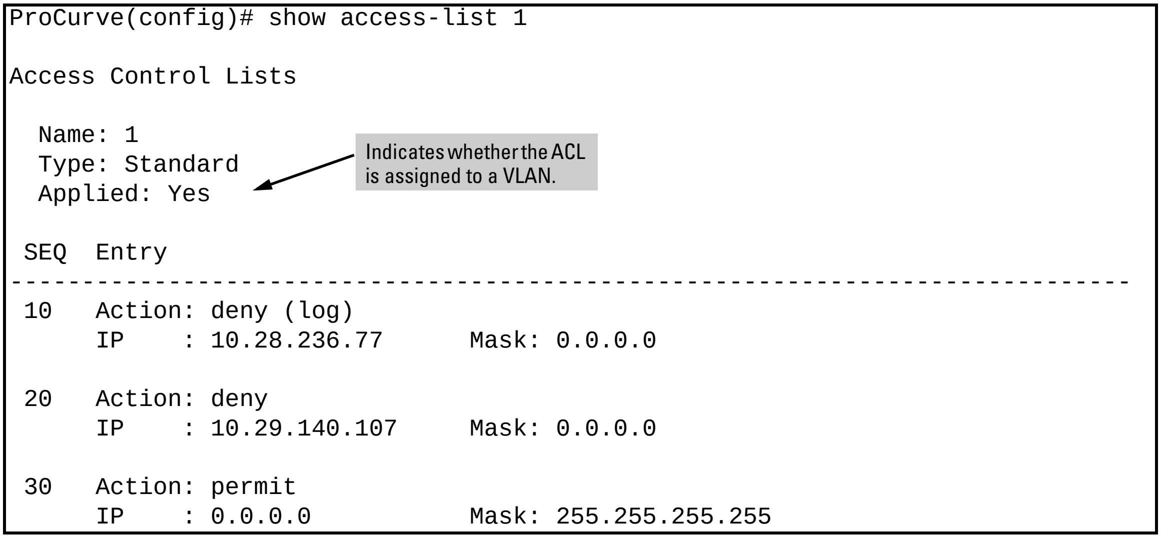
Syntax: show access-list < *acl-id* >

Display detailed information on the content of a specific ACL configured in the running-config file.

For example, suppose you configured the following two ACLs in the routing switch:

ACL ID	ACL Type	Desired Action
1	Standard	• Deny IP traffic from 18.28.236.77 and 18.29.140.107. • Permit IP traffic from all other sources.
105	Extended	• Permit any TCP traffic from 18.30.133.27 to any destination. • Deny any other IP traffic from 18.30.133.(1-255). • Permit all other IP traffic from any source to any destination.

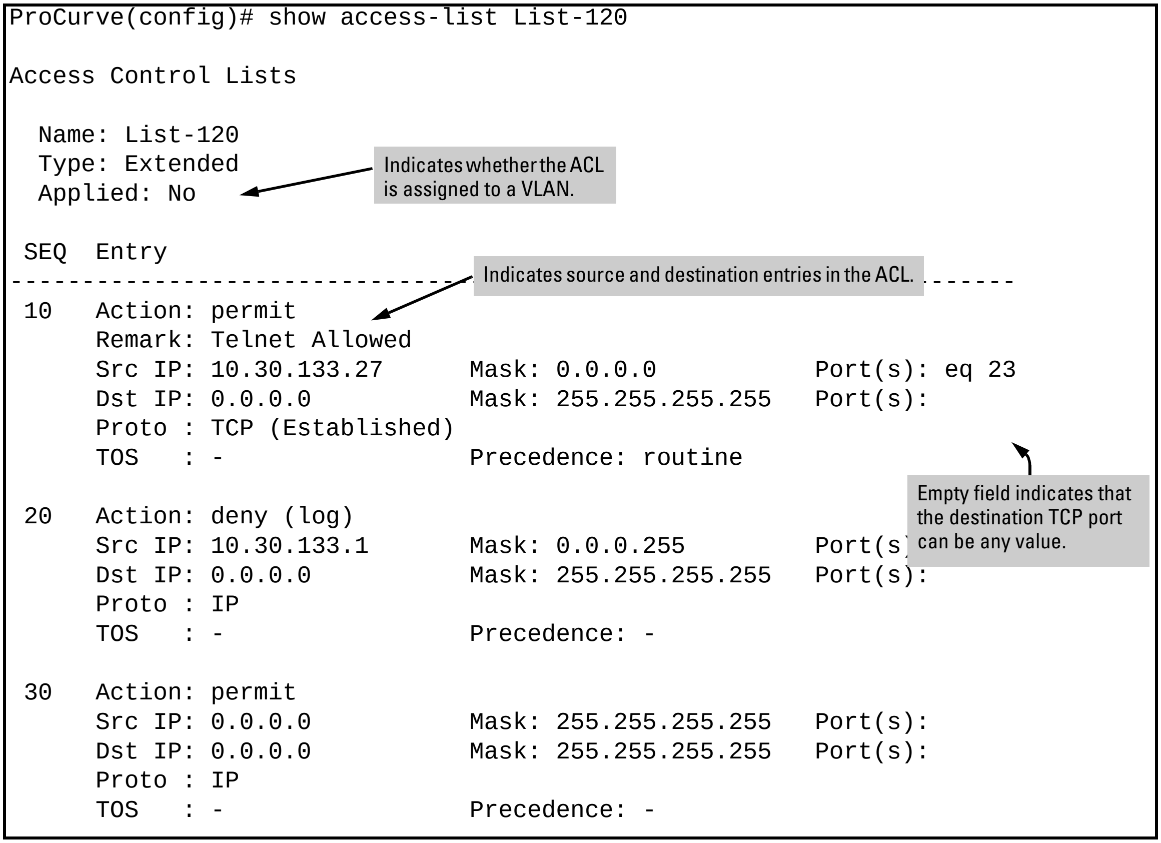
Inspect the ACLs as follows:



```
ProCurve(config)# show access-list 1

Access Control Lists

  Name: 1
  Type: Standard
  Applied: Yes

 SEQ  Entry
-------------------------------------------------------------------------------
 10   Action: deny (log)
      IP    : 10.28.236.77      Mask: 0.0.0.0

 20   Action: deny
      IP    : 10.29.140.107     Mask: 0.0.0.0

 30   Action: permit
      IP    : 0.0.0.0           Mask: 255.255.255.255
```

Indicates whether the ACL is assigned to a VLAN.

Figure 7-30. Example of a Listing a Standard ACL



```
ProCurve(config)# show access-list List-120

Access Control Lists

  Name: List-120
  Type: Extended
  Applied: No

 SEQ  Entry
----------------------------------------------------------------------
 10   Action: permit
      Remark: Telnet Allowed
      Src IP: 10.30.133.27      Mask: 0.0.0.0           Port(s): eq 23
      Dst IP: 0.0.0.0           Mask: 255.255.255.255   Port(s):
      Proto : TCP (Established)
      TOS   : -                 Precedence: routine

 20   Action: deny (log)
      Src IP: 10.30.133.1       Mask: 0.0.0.255         Port(s
      Dst IP: 0.0.0.0           Mask: 255.255.255.255   Port(s):
      Proto : IP
      TOS   : -                 Precedence: -

 30   Action: permit
      Src IP: 0.0.0.0           Mask: 255.255.255.255   Port(s):
      Dst IP: 0.0.0.0           Mask: 255.255.255.255   Port(s):
      Proto : IP
      TOS   : -                 Precedence: -
```

Indicates whether the ACL is assigned to a VLAN.

Indicates source and destination entries in the ACL.

Empty field indicates that the destination TCP port can be any value.

Figure 7-31. Examples of Listings Showing the Content of Standard and Extended ACLs

Table 7-11. Descriptions of Data Types Included in Show Access-List < *acl-id* > Output

Field	Description
Name	The ACL identifier. Can be a number from 1 to 199, or a name.
Type	Standard or Extended. The former uses only source IP addressing. The latter uses both source and destination IP addressing and also allows TCP or UDP port specifiers.
Applied	“Yes” means the ACL has been applied to a VLAN. “No” means the ACL exists in the routing switch configuration, but has not been applied to any VLANs, and is therefore not in use.
SEQ	The sequential number of the Access Control Entry (ACE) in the specified ACL.
Entry	Lists the content of the ACEs in the selected ACL.
Action	Permit (forward) or deny (drop) a packet when it is compared to the criteria in the applicable ACE and found to match. Includes the optional **log** option, if used, in **deny** actions.
Remark	Displays any optional remark text configured for the selected ACE.
IP	**Used for Standard ACLs:** The source IP address to which the configured mask is applied to determine whether there is a match with a packet.
Src IP	**Used for Extended ACLs:** Same as above.
Dst IP	**Used for Extended ACLs:** The source and destination IP addresses to which the corresponding configured masks are applied to determine whether there is a match with a packet.
Mask	The mask configured in an ACE and applied to the corresponding IP address in the ACE to determine whether a packet matches the filtering criteria.
Proto	Used only in extended ACLs to specify the packet protocol type to filter. Must be either IP, TCP, or UDP. For TCP protocol selections, includes the **established** option, if configured.
Port(s)	Used only in extended ACLs to show any TCP or UDP operator and port number(s) included in the ACE.
TOS	Used only in extended ACLs to indicate Type-of-Service setting, if any.
Precedence	Used only in extended ACLs to indicate the IP precedence setting, if any.

Display All ACLs and Their Assignments in the Routing Switch Startup-Config File and Running-Config File

The **show config** and **show running** commands include in their listings any configured ACLs and any ACL assignments to VLANs. Refer to figure 7-9 (page 7-36) for an example. Remember that **show config** lists the startup-config file and **show running** lists the running-config file.

Creating or Editing ACLs Offline

The section titled "Editing an Existing ACL" on page 7-73 describes how to use the CLI to edit an ACL, and is most applicable in cases where the ACL is short or there is only a minor editing task to perform. The offline method provides a useful alternative to using the CLI for creating or extensively editing a large ACL.This section describes how to:

- move an existing ACL to a TFTP server
- use a text (.txt) file format to create a new ACL or edit an existing ACL offline
- use TFTP to load an offline ACL into the routing switch's running-config

For longer ACLs that may be difficult or time-consuming to accurately create or edit in the CLI, you can use the offline method described in this section.

Creating or Editing an ACL Offline

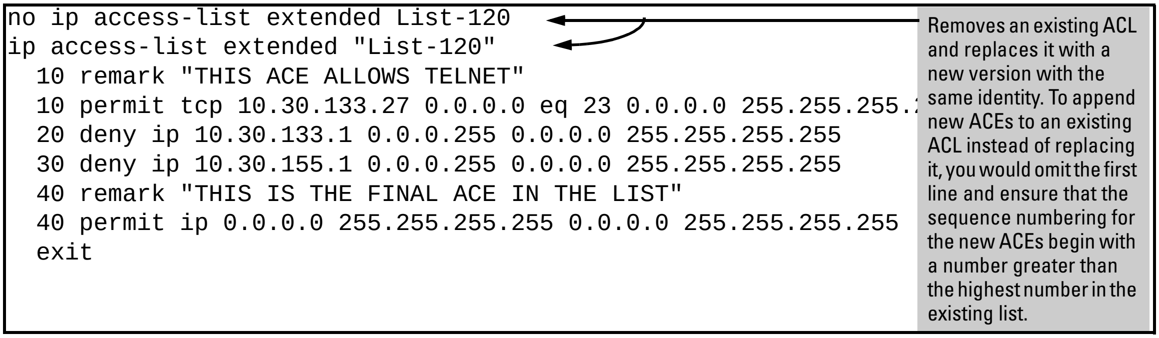
The Offline Process

1. Begin by doing one of the following:
   - To edit one or more existing ACLs, use **copy command-output tftp** to copy the current version of the ACL configuration to a file in your TFTP server. For example, to copy the ACL configuration to a file named **acl-02.txt** in the TFTP directory on a server at 10.28.227.2:

     ```
     ProCurve# copy command-output 'show access-list
     config' tftp 10.28.227.2 acl02.txt pc
     ```
   - To create a new ACL, just open a text (.txt) file in the appropriate directory on a TFTP server accessible to the routing switch.
2. Use a text editor to create or edit the ACL(s) in the ***.txt** ASCII file format.

   If you are replacing an ACL on the routing switch with a new ACL that uses the same number or name syntax, begin the command file with a **no ip access-list** command to remove the earlier version of the ACL from the routing switch's running-config file. Otherwise, the routing switch will append the new ACEs in the ACL you download to the existing ACL. For example, if you planned to use the **copy** command to *replace* ACL "List-120", you would place this command at the beginning of the edited file:

   ```
   no ip access-list extended List-120
   ```



```
no ip access-list extended List-120
ip access-list extended "List-120"
  10 remark "THIS ACE ALLOWS TELNET"
  10 permit tcp 10.30.133.27 0.0.0.0 eq 23 0.0.0.0 255.255.255.
  20 deny ip 10.30.133.1 0.0.0.255 0.0.0.0 255.255.255.255
  30 deny ip 10.30.155.1 0.0.0.255 0.0.0.0 255.255.255.255
  40 remark "THIS IS THE FINAL ACE IN THE LIST"
  40 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
  exit
```

Removes an existing ACL and replaces it with a new version with the same identity. To append new ACEs to an existing ACL instead of replacing it, you would omit the first line and ensure that the sequence numbering for the new ACEs begin with a number greater than the highest number in the existing list.

Figure 7-32. Example of an Offline ACL File Designed To Replace An Existing ACL

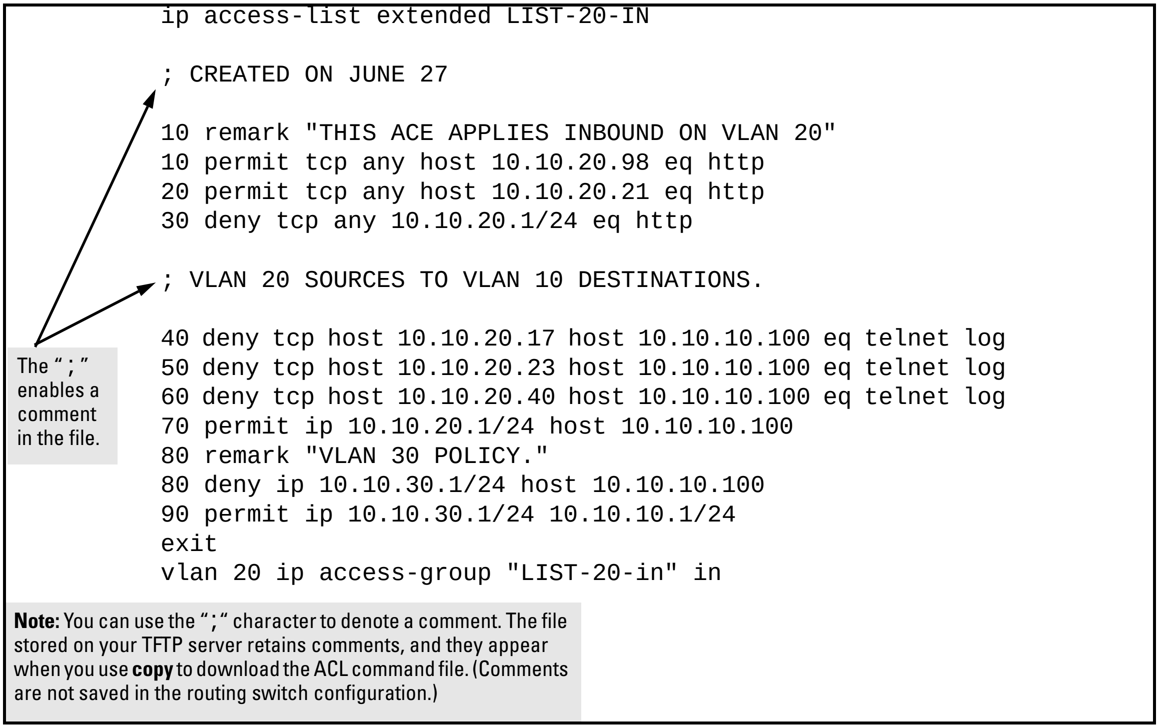
3. Use **copy tftp command-file** to download the file as a list of commands to the routing switch.

Example of Using the Offline Process

For example, suppose that you wanted to create an extended ACL to fulfill the following requirements (Assume a subnet mask of 255.255.255.0 and a TFTP server at 10.10.10.1.):

- ID: "LIST-20-IN"
- Deny Telnet access to a server at 10.10.10.100 on VLAN 10 from these three IP addresses on VLAN 20 (with ACL logging):
  - 10.10.20.17
  - 10.10.20.23
  - 10.10.20.40
- Allow any access to the server from all other addresses on VLAN 20:
- Permit internet access to these two IP address on VLAN 20, but deny access to all other addresses on VLAN 20 (without ACL logging).
  - 10.10.20.98
  - 10.10.20.21
- Deny all other traffic from VLAN 20 to VLAN 10.
- Deny all traffic from VLAN 30 (10.10.30.0) to the server at 10.10.10.100 on VLAN 10 (without ACL logging), but allow any other traffic from VLAN 30 to VLAN 10.
- Deny all other inbound traffic to VLAN 20. (Hint: The Implicit Deny can achieve this objective.)

1. You would create a **.txt** file with the content shown in figure 7-34.



```
ip access-list extended LIST-20-IN

; CREATED ON JUNE 27

10 remark "THIS ACE APPLIES INBOUND ON VLAN 20"
10 permit tcp any host 10.10.20.98 eq http
20 permit tcp any host 10.10.20.21 eq http
30 deny tcp any 10.10.20.1/24 eq http

; VLAN 20 SOURCES TO VLAN 10 DESTINATIONS.

40 deny tcp host 10.10.20.17 host 10.10.10.100 eq telnet log
50 deny tcp host 10.10.20.23 host 10.10.10.100 eq telnet log
60 deny tcp host 10.10.20.40 host 10.10.10.100 eq telnet log
70 permit ip 10.10.20.1/24 host 10.10.10.100
80 remark "VLAN 30 POLICY."
80 deny ip 10.10.30.1/24 host 10.10.10.100
90 permit ip 10.10.30.1/24 10.10.10.1/24
exit
vlan 20 ip access-group "LIST-20-in" in
```

The ";" enables a comment in the file.

Note: You can use the ";" character to denote a comment. The file stored on your TFTP server retains comments, and they appear when you use **copy** to download the ACL command file. (Comments are not saved in the routing switch configuration.)

Figure 7-33. Example of a .txt File Designed for Creating an ACL

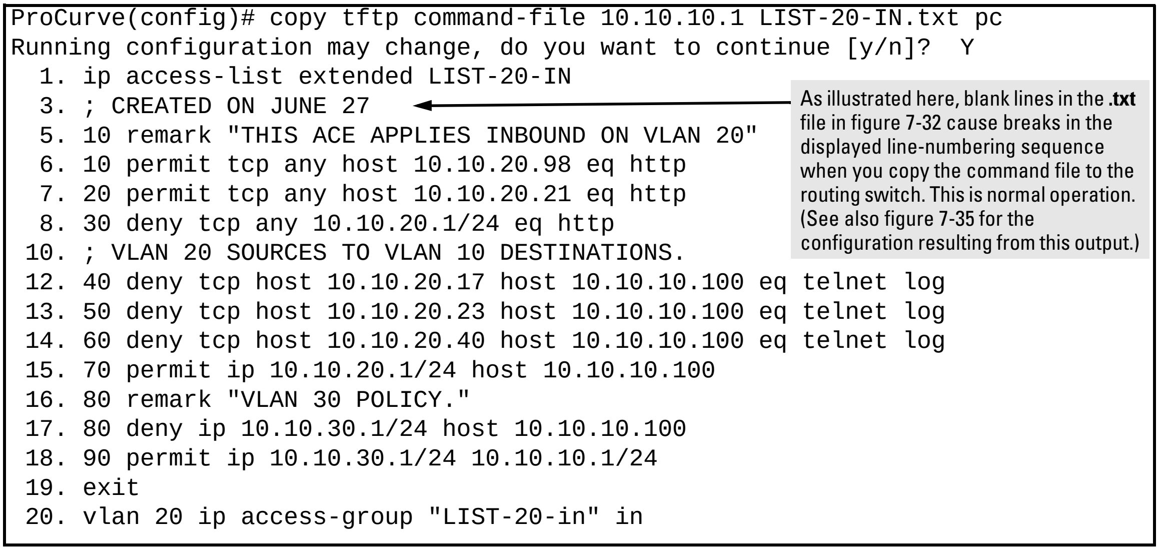
2. After you copy the above .txt file to a TFTP server the routing switch can access, you would then execute the following command:

   copy tftp command-file 10.10.10.1 LIST-20-IN.txt pc

   In this example, the CLI would show the following output to indicate that the ACL was successfully downloaded to the routing switch:

Note

If a transport error occurs, the routing switch does not execute the command and the ACL is not configured.



```
ProCurve(config)# copy tftp command-file 10.10.10.1 LIST-20-IN.txt pc
Running configuration may change, do you want to continue [y/n]?  Y
  1. ip access-list extended LIST-20-IN
  3. ; CREATED ON JUNE 27
  5. 10 remark "THIS ACE APPLIES INBOUND ON VLAN 20"
  6. 10 permit tcp any host 10.10.20.98 eq http
  7. 20 permit tcp any host 10.10.20.21 eq http
  8. 30 deny tcp any 10.10.20.1/24 eq http
 10. ; VLAN 20 SOURCES TO VLAN 10 DESTINATIONS.
 12. 40 deny tcp host 10.10.20.17 host 10.10.10.100 eq telnet log
 13. 50 deny tcp host 10.10.20.23 host 10.10.10.100 eq telnet log
 14. 60 deny tcp host 10.10.20.40 host 10.10.10.100 eq telnet log
 15. 70 permit ip 10.10.20.1/24 host 10.10.10.100
 16. 80 remark "VLAN 30 POLICY."
 17. 80 deny ip 10.10.30.1/24 host 10.10.10.100
 18. 90 permit ip 10.10.30.1/24 10.10.10.1/24
 19. exit
 20. vlan 20 ip access-group "LIST-20-in" in
```

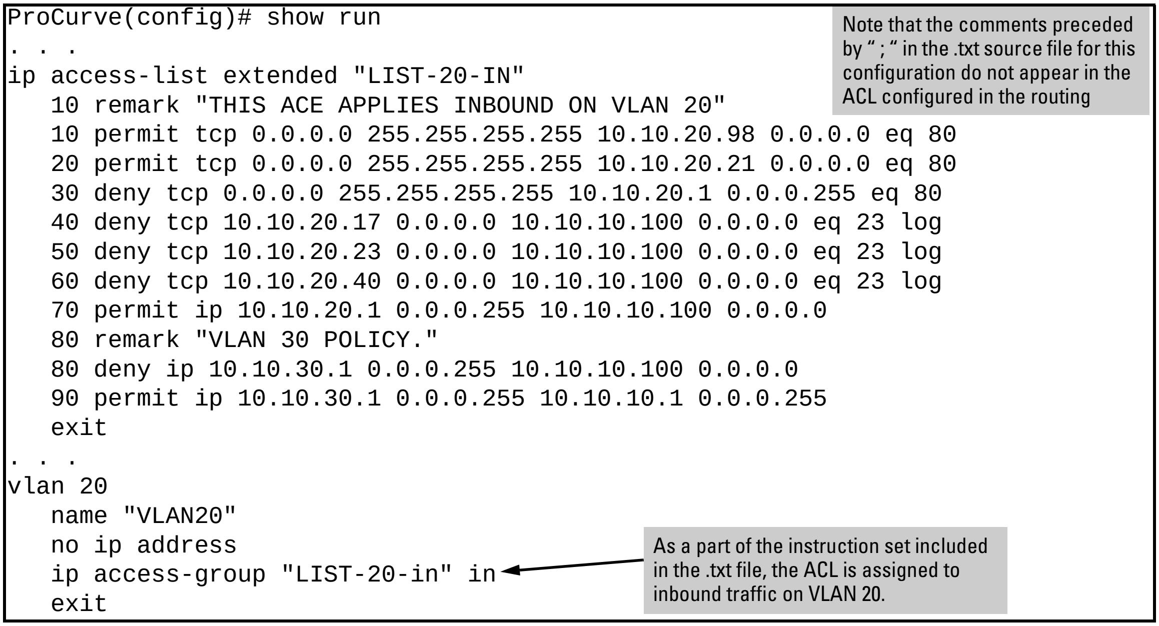
As illustrated here, blank lines in the **.txt** file in figure 7-32 cause breaks in the displayed line-numbering sequence when you copy the command file to the routing switch. This is normal operation. (See also figure 7-35 for the configuration resulting from this output.)

Figure 7-34. Example of Using "copy tftp command-file" To Configure an ACL in the Routing Switch

3. In this example, the command to assign the ACL to a VLAN was included in the .txt command file. If this is not done in your applications, then the next step is to manually assign the new ACL to the intended VLAN.

   vlan < *vid* > ip access-group < *identifier* > in

4. You can then use the **show run** or **show access-list config** command to inspect the routing switch configuration to ensure that the ACL was properly downloaded.



```
ProCurve(config)# show run
. . .
ip access-list extended "LIST-20-IN"
   10 remark "THIS ACE APPLIES INBOUND ON VLAN 20"
   10 permit tcp 0.0.0.0 255.255.255.255 10.10.20.98 0.0.0.0 eq 80
   20 permit tcp 0.0.0.0 255.255.255.255 10.10.20.21 0.0.0.0 eq 80
   30 deny tcp 0.0.0.0 255.255.255.255 10.10.20.1 0.0.0.255 eq 80
   40 deny tcp 10.10.20.17 0.0.0.0 10.10.10.100 0.0.0.0 eq 23 log
   50 deny tcp 10.10.20.23 0.0.0.0 10.10.10.100 0.0.0.0 eq 23 log
   60 deny tcp 10.10.20.40 0.0.0.0 10.10.10.100 0.0.0.0 eq 23 log
   70 permit ip 10.10.20.1 0.0.0.255 10.10.10.100 0.0.0.0
   80 remark "VLAN 30 POLICY."
   80 deny ip 10.10.30.1 0.0.0.255 10.10.10.100 0.0.0.0
   90 permit ip 10.10.30.1 0.0.0.255 10.10.10.1 0.0.0.255
   exit
. . .
vlan 20
   name "VLAN20"
   no ip address
   ip access-group "LIST-20-in" in
   exit
```

Note that the comments preceded by " ; " in the .txt source file for this configuration do not appear in the ACL configured in the routing

As a part of the instruction set included in the .txt file, the ACL is assigned to inbound traffic on VLAN 20.

Figure 7-35. Example of Verifying the .txt File Download to the Routing Switch

5. If the configuration appears satisfactory, save it to the startup-config file:

   ```
   ProCurve(config)# write memory
   ```

Enable ACL "Deny" Logging

ACL logging enables the routing switch to generate a message when IP traffic meets the criteria for a match with an ACE that results in an explicit "deny" action. You can use ACL logging to help:

- Test your network to ensure that your ACL configuration is detecting and denying the traffic you do not want forwarded
- Receive notification when the routing switch detects attempts to forward traffic you have designed your ACLs to reject (deny)

The routing switch sends ACL messages to Syslog and optionally to the current console, Telnet, or SSH session. You can use **logging < >** to configure up to six Syslog server destinations.

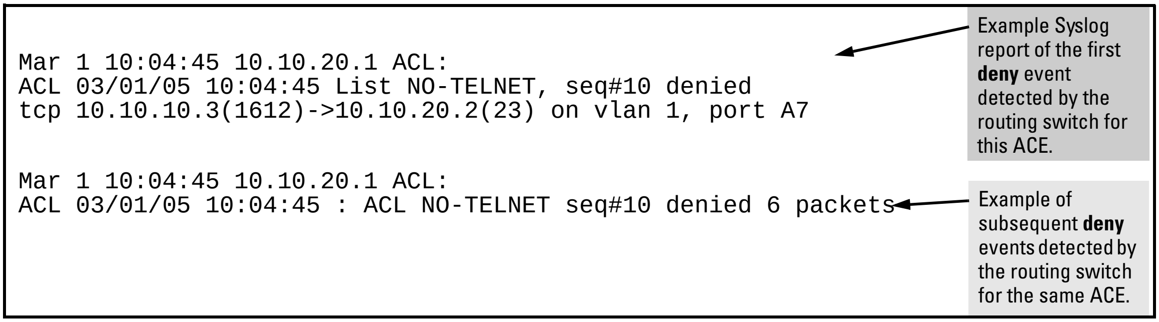
Requirements for Using ACL Logging

- The routing switch configuration must include an ACL (1) assigned to a static VLAN and (2) containing an ACE configured with the **deny** action and the **log** option.
- To screen routed packets with destination IP addresses outside of the routing switch, IP routing must be enabled.
- For ACL logging to a Syslog server:
  - The server must be accessible to the routing switch and identified in the running configuration.
  - The logging facility must be enabled for Syslog.
  - Debug must be configured to:
    - support ACL messages
    - send debug messages to the desired debug destination

  These requirements are described in more detail under "Enabling ACL Logging on the Routing Switch" on page 7-96.

ACL Logging Operation

When the routing switch detects a packet match with an ACE and the ACE includes both the **deny** action and the optional **log** parameter, an ACL log message is sent to the designated debug destination. The first time a packet matches an ACE with **deny** and **log** configured, the message is sent immediately to the destination and the routing switch starts a wait-period of approximately five minutes. (The exact duration of the period depends on how the packets are internally routed.) At the end of the collection period, the routing switch sends a single-line summary of any additional "deny" matches for that ACE (and any other "deny" ACEs for which the routing switch detected a match). If no further log messages are generated in the wait-period, the routing switch suspends the timer and resets itself to send a message as soon as a new "deny" match occurs. The data in the message includes the information illustrated in figure 7-36.



```
Mar 1 10:04:45 10.10.20.1 ACL:
ACL 03/01/05 10:04:45 List NO-TELNET, seq#10 denied
tcp 10.10.10.3(1612)->10.10.20.2(23) on vlan 1, port A7

Mar 1 10:04:45 10.10.20.1 ACL:
ACL 03/01/05 10:04:45 : ACL NO-TELNET seq#10 denied 6 packets
```

Example Syslog report of the first **deny** event detected by the routing switch for this ACE.

Example of subsequent **deny** events detected by the routing switch for the same ACE.

Figure 7-36. Content of a Message Generated by an ACL-Deny Action

Enabling ACL Logging on the Routing Switch

1. If you are using a Syslog server, use the **logging <** ***ip-addr*** **>** command to configure the Syslog server IP address(es). Ensure that the routing switch can access any Syslog server(s) you specify.
2. Use **logging facility syslog** to enable the logging for Syslog operation.
3. Use the **debug destination** command to configure one or more log destinations. (Destination options include **logging**, **session**, and **windshell**. For more information on debug, refer to "Debug and Syslog Messaging Operation" in appendix C, "Troubleshooting", in the *Management and Configuration Guide* for your routing switch.)
4. Use **debug acl** or **debug all** to configure the debug operation to include ACL messages.
5. Configure one or more ACLs with the **deny** action and the **log** option.

For example, suppose that you want to configure the following operation:

- On VLAN 10 configure an extended ACL with an ACL-ID of "NO-TELNET" to deny Telnet traffic from IP address 10.10.10.3 to any destination.
- Configure the routing switch to send an ACL log message to the current console session and to a Syslog server at IP address 10.10.20.3 on VLAN 20 if the routing switch detects a packet match denying a Telnet attempt from 10.10.10.3.

(This example assumes that IP routing is already configured on the routing switch.)

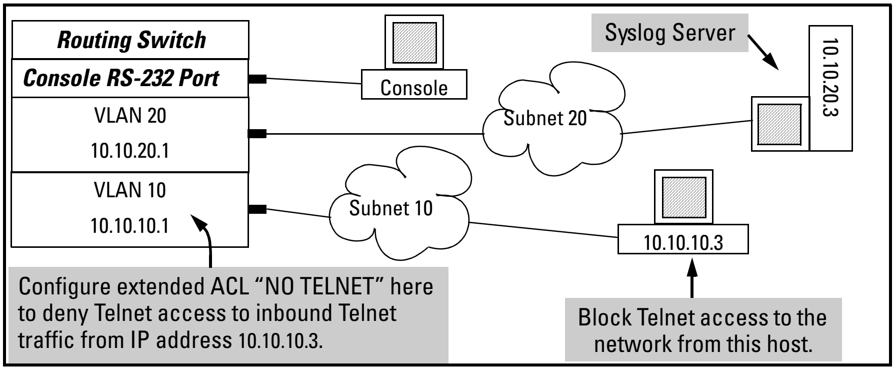
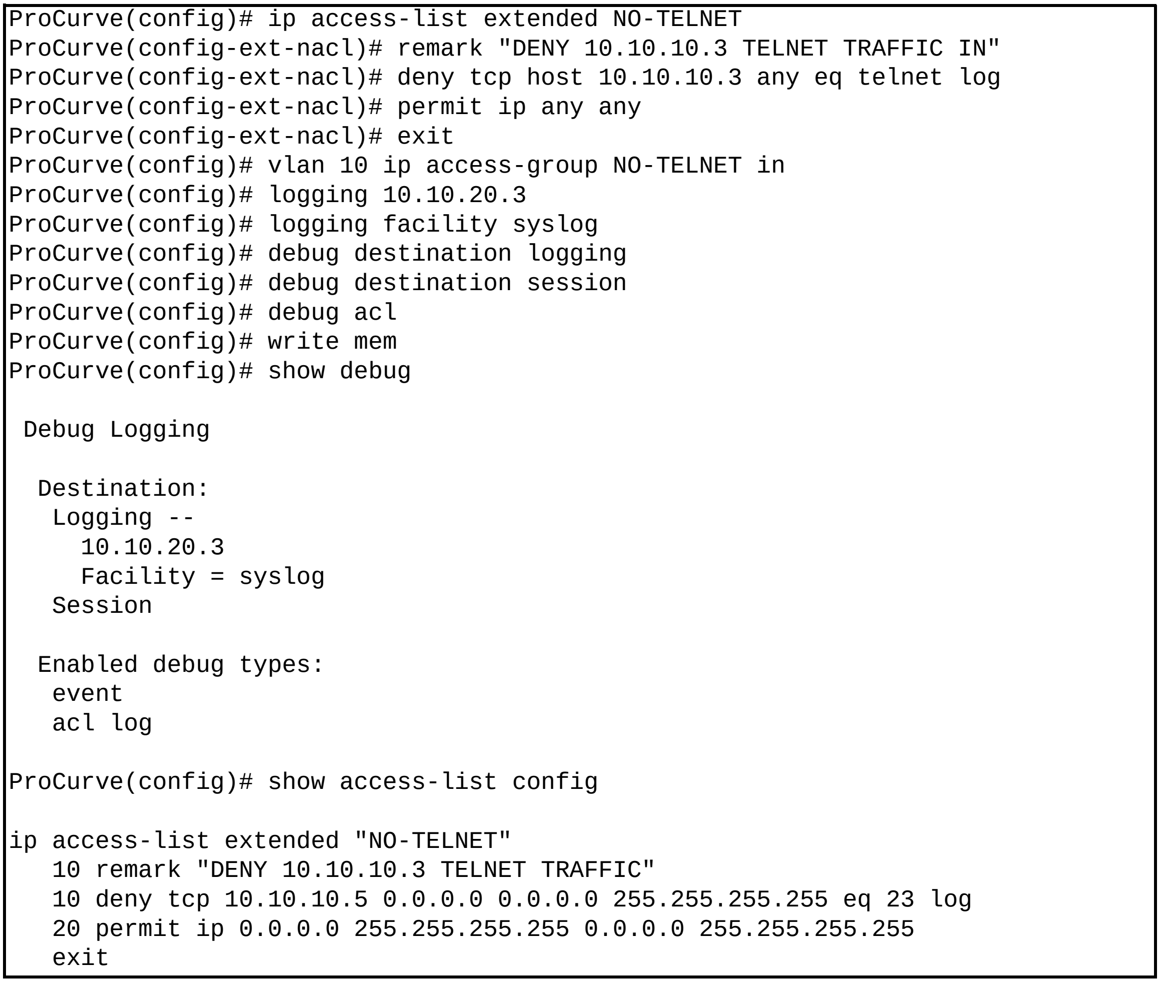

Routing Switch
Console RS-232 Port
VLAN 20
10.10.20.1
VLAN 10
10.10.10.1
Console
Syslog Server
10.10.20.3
Subnet 20
Subnet 10
10.10.10.3
Configure extended ACL "NO TELNET" here to deny Telnet access to inbound Telnet traffic from IP address 10.10.10.3.
Block Telnet access to the network from this host.

Figure 7-37. Example of an ACL Log Application



```
ProCurve(config)# ip access-list extended NO-TELNET
ProCurve(config-ext-nacl)# remark "DENY 10.10.10.3 TELNET TRAFFIC IN"
ProCurve(config-ext-nacl)# deny tcp host 10.10.10.3 any eq telnet log
ProCurve(config-ext-nacl)# permit ip any any
ProCurve(config-ext-nacl)# exit
ProCurve(config)# vlan 10 ip access-group NO-TELNET in
ProCurve(config)# logging 10.10.20.3
ProCurve(config)# logging facility syslog
ProCurve(config)# debug destination logging
ProCurve(config)# debug destination session
ProCurve(config)# debug acl
ProCurve(config)# write mem
ProCurve(config)# show debug

 Debug Logging

  Destination:
   Logging --
     10.10.20.3
     Facility = syslog
   Session

  Enabled debug types:
   event
   acl log

ProCurve(config)# show access-list config

ip access-list extended "NO-TELNET"
   10 remark "DENY 10.10.10.3 TELNET TRAFFIC"
   10 deny tcp 10.10.10.5 0.0.0.0 0.0.0.0 255.255.255.255 eq 23 log
   20 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

Figure 7-38. Commands for Applying an ACL with Logging to Figure 7-37

Operating Notes for ACL Logging

- The ACL logging feature generates a message only when packets are explicitly denied as the result of a match, and not when explicitly permitted or implicitly denied. To help test ACL logging, configure the last entry in an ACL as an explicit **deny** statement with a **log** statement included, and apply the ACL to an appropriate VLAN.

- Logging enables you to selectively test specific devices or groups. However, excessive logging can affect routing switch performance. For this reason, ProCurve recommends that you remove the logging option from ACEs for which you do not have a present need. Also, avoid configuring logging where it does not serve an immediate purpose. (Note that ACL logging is not designed to function as an accounting method.) See also "Apparent Failure To Log All 'Deny' Matches" in the section titled "ACL Problems", found in appendix C, "Troubleshooting" of the *Management and Configuration Guide* for your routing switch.

- When configuring logging, you can reduce excessive resource use by configuring the appropriate ACEs to match with specific hosts instead of entire subnets.

General ACL Operating Notes

ACLs do not provide DNS hostname support. ACLs cannot be configured to screen hostname traffic between the routing switch and a DNS.

Protocol Support . ACL criteria does not include use of MAC information or QoS.

ACLs Do Not Affect Serial Port Access. ACLs do not apply to the routing switch's serial port.

When the ACL Configuration Includes TCP or UDP Options, the Routing Switch Operates in "Strict" TCP and UDP Mode for Increased Control. The routing switch compares all TCP and UDP packets against the ACLs. (In the ProCurve 9300m and 9404sl Routing Switches, the Strict TCP and Strict UDP modes are optional and must be specifically invoked.)

Replacing or Adding To an Active ACL Policy. If you assign an ACL to a VLAN and subsequently add or replace ACEs in that ACL, each new ACE becomes active when you enter it.

ACL Screening of Traffic Generated by the Routing Switch. Outbound ACLs on a routing switch do not screen traffic (such as broadcasts, Telnet, Ping, and ICMP replies) *generated by the routing switch itself*. Note that ACLs do screen this type of traffic when other devices generate it. Similarly, ACLs can screen responses from other devices to unscreened traffic the routing switch generates.

Minimum Number of ACEs in an ACL. Any ACL must include at least one ACE to enable traffic screening. A numbered ACL cannot be created without at least one ACE. A named ACL can be created "empty"; that is, without any ACEs. However in an empty ACL applied to a VLAN, the Implicit Deny function does not operate, and the ACL has no effect on traffic.

Monitoring Shared Resources. ACLs share internal routing switch resources with several other features. The routing switch provides ample resources for all features. However, if the internal resources become fully subscribed, additional ACLs cannot be applied until the necessary resources are released from other applications. For information on determining current resource availability and usage, refer to appendix E, "Monitoring Resources" in the *Management and Configuration Guide* for your switch.

—This page intentionally unused—

Stack Management for the Series 3500yl Switches and the 6200yl Switch

Contents

Introduction to Stack Management on Series 3500yl Switches and the 6200yl Switch

This feature is available on the 3500yl and 6200yl switches, but not on the 5400zl switches.

ProCurve Stack Management (*stacking*) enables you to use a single IP address and standard network cabling to manage a group of up to 16 total switches in the same IP subnet (broadcast domain). Using stacking, you can:

- Reduce the number of IP addresses needed in your network.
- Simplify management of small workgroups or wiring closets while scaling your network to handle increased bandwidth demand.
- Eliminate any specialized cables for stacking connectivity and remove the distance barriers that typically limit your topology options when using other stacking technologies.
- Add switches to your network without having to first perform IP addressing tasks.

Stacking Support on ProCurve Switches

As of October 2005, the following ProCurve switches include stacking:

- ProCurve Series 6400cl
- ProCurve Series 6200yl
- ProCurve Switch 6108
- ProCurve Series 4200vl
- ProCurve Series 4100gl
- ProCurve Series 3500yl
- ProCurve Series 3400cl
- ProCurve Series 2600
- ProCurve Series 2800
- ProCurve Series 2500
- ProCurve Switch 8000M[1, 2]
- ProCurve Switch 4000M[1, 2]
- ProCurve Switch 2424M1[1, 2]
- ProCurve Switch 2400M[1, 2]
- ProCurve Switch 1600M[1, 2]

[1] Requires software release C.08.03 or later, which is included with the 8000M, 4000M, 2424M, and 1600M models as of July, 2000. Release C.08.03 or a later version is also available on the ProCurve Networking web site at **www.procurve.com**. (Click on **Software updates**.)

[2] Discontinued product.

Note

Stacking and meshing cannot both be enabled at the same time on a Series 3500yl switch or a 6200yl switch.

In the default configuration, stacking in the "candidate" state is enabled on the Series 3500yl switches and on the 6200yl switch.

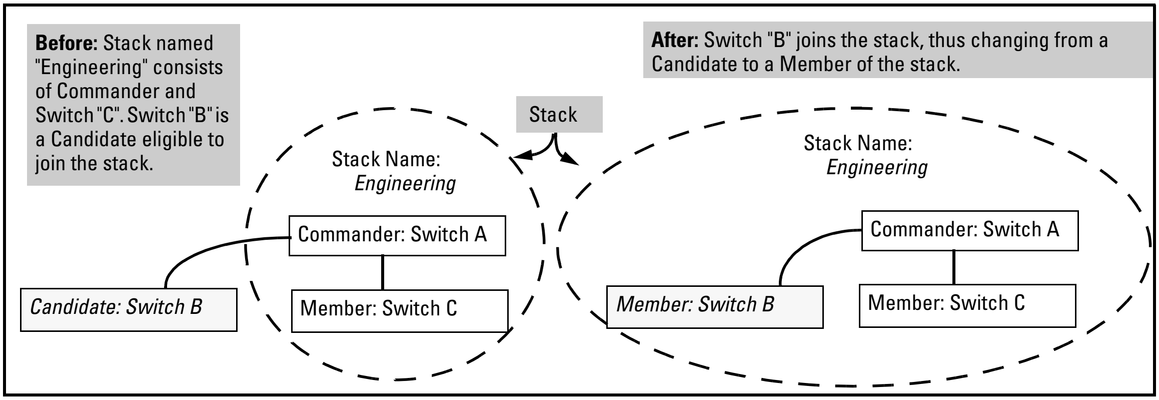
Summary of Stacking Features

Feature	Default	Menu	CLI	Web
view stack status				
view status of a single switch	n/a	page 8-26 thru page 8-28	page 8-31	Refer to Online Help
view candidate status	n/a	↑	page 8-31	↑
view status of commander and its stack	n/a		page 8-32	
view status of all stacking-enabled switches in the ip subnet	n/a		page 8-32	
configure stacking				
enable/disable candidate Auto-Join	enabled/Yes	page 8-15	page 8-37	
"push" a candidate into a stack	n/a	page 8-15	page 8-37	
configure a switch to be a commander	n/a	page 8-13	page 8-33	
"push" a member into another stack	n/a	page 8-24	page 8-39	
remove a member from a stack	n/a	page 8-21	page 8-40 or page 8-41	
"pull" a candidate into a stack	n/a	page 8-17	page 8-36	
"pull" a member from another stack	n/a	page 8-19	page 8-38	
convert a commander or member to a member of another stack	n/a	page 8-24	page 8-39	
access member switches for configuration and traffic monitoring	n/a	page 8-23	page 8-42	
disable stacking	enabled	page 8-15	page 8-44	
transmission interval	60 seconds	page 8-13	page 8-44	

Components of ProCurve Stack Management

Table 8-1. Stacking Definitions

Stack	Consists of a Commander switch and any Member switches belonging to that Commander's stack.
Commander	A switch that has been manually configured as the controlling device for a stack. When this occurs, the switch's stacking configuration appears as **Commander**.
Candidate	A switch that is ready to join (become a Member of) a stack through either automatic or manual methods. A switch configured as a Candidate is not in a stack.
Member	A switch that has joined a stack and is accessible from the stack Commander.


Before: Stack named "Engineering" consists of Commander and Switch "C". Switch "B" is a Candidate eligible to join the stack.

After: Switch "B" joins the stack, thus changing from a Candidate to a Member of the stack.

Figure 8-1. Illustration of a Switch Moving from Candidate to Member

General Stacking Operation

After you configure one switch to operate as the Commander of a stack, additional switches can join the stack by either automatic or manual methods. After a switch becomes a Member, you can work through the Commander switch to further configure the Member switch as necessary for all of the additional software features available in the switch.

The Commander switch serves as the in-band entry point for access to the Member switches. For example, the Commander's IP address becomes the path to all stack Members and the Commander's Manager password controls access to all stack Members.

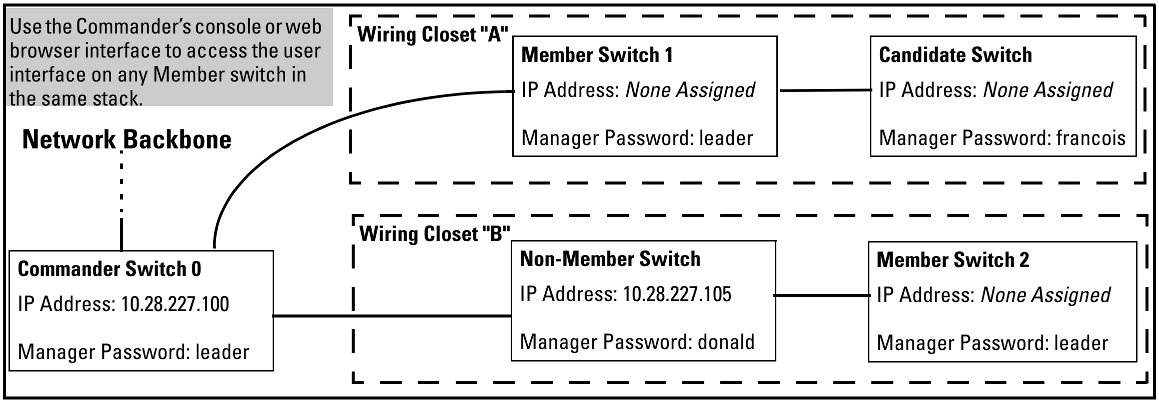

Use the Commander's console or web browser interface to access the user interface on any Member switch in the same stack.
Network Backbone
Wiring Closet "A"
Member Switch 1
IP Address: None Assigned
Manager Password: leader
Candidate Switch
IP Address: None Assigned
Manager Password: francois
Wiring Closet "B"
Commander Switch 0
IP Address: 10.28.227.100
Manager Password: leader
Non-Member Switch
IP Address: 10.28.227.105
Manager Password: donald
Member Switch 2
IP Address: None Assigned
Manager Password: leader

Figure 8-2. Example of Stacking with One Commander Controlling Access to Wiring Closet Switches

Interface Options. You can configure stacking through the switch's menu interface, CLI, or the web browser interface. For information on how to use the web browser interface to configure stacking, see the online Help for the web browser interface.

Web Browser Interface Window for Commander Switches. The web browser interface window for a Commander switch differs in appearance from the same window for non-commander switches.

Operating Rules for Stacking

General Rules

- Stacking is an optional feature (enabled in the default configuration) and can easily be disabled. Stacking has no effect on the normal operation of the switch in your network.
- A stack requires one Commander switch. (Only one Commander allowed per stack.)
- All switches in a particular stack must be in the same IP subnet (broadcast domain). A stack cannot cross a router.
- A stack accepts up to 16 switches (numbered 0-15), including the Commander (always numbered 0).
- The stacking feature supports up to 100 switches in the same IP subnet (broadcast domain), however, a switch can belong to only one stack.
- If multiple VLANs are configured, stacking uses only the primary VLAN on any switch. In the factory-default configuration, the DEFAULT_VLAN is the primary VLAN. (See "Stacking Operation with Multiple VLANs Configured" on page 8-44 and "The Primary VLAN" on page 2-45.)
- Stacking allows intermediate devices that do not support stacking. This enables you to include switches that are distant from the Commander.

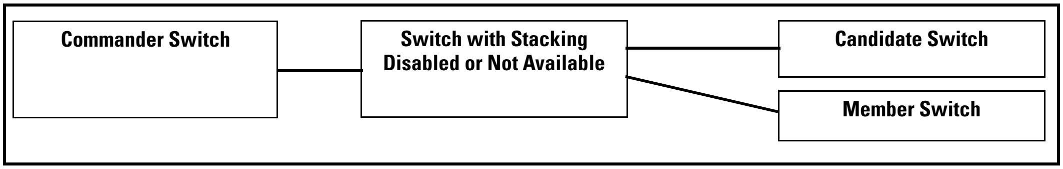


Figure 8-3. Example of a Non-Stacking Device Used in a Stacking Environment

Specific Rules

Table 8-2. Specific Rules for Commander, Candidate, and Member Switch

	IP Addressing and Stack Name	Number Allowed Per Stack	Passwords	SNMP Communities
Commander	**IP Addr:** Requires an assigned IP address and mask for access via the network. **Stack Name:** Required	Only one Commander switch is allowed per stack.	The Commander's Manager and Operator passwords are assigned to any switch becoming a Member of the stack. If you change the Commander's passwords, the Commander propagates the new passwords to all stack Members.	Standard SNMP community operation. The Commander also operates as an SNMP proxy to Members for all SNMP communities configured in the Commander.
Candidate	**IP Addr:** Optional. Configuring an IP address allows access via Telnet or web browser interface while the switch is not a stack member. In the factory default configuration the switch automatically acquires an IP address if your network includes DHCP service. **Stack Name:** N/A	n/a	Passwords optional. If the Candidate becomes a stack Member, it assumes the Commander's Manager and Operator passwords. If a candidate has a password, it cannot be automatically added to a stack. In this case, if you want the Candidate in a stack, you must manually add it to the stack.	Uses standard SNMP community operation if the Candidate has its own IP addressing.
Member	**IP Addr:** Optional. Configuring an IP address allows access via Telnet or web browser interface without going through the Commander switch. This is useful, for example, if the stack Commander fails and you need to convert a Member switch to operate as a replacement Commander. **Stack Name:** N/A	Up to 15 Members per stack.	When the switch joins the stack, it automatically assumes the Commander's Manager and Operator passwords and discards any passwords it may have had while a Candidate. **Note:** If a Member leaves a stack for any reason, it retains the passwords assigned to the stack Commander at the time of departure from the stack.	Belongs to the same SNMP communities as the Commander (which serves as an SNMP proxy to the Member for communities to which the Commander belongs). To join other communities that *exclude* the Commander, the Member must have its own IP address. Loss of stack membership means loss of membership in any community that is configured only in the Commander. See "SNMP Community Operation in a Stack" on page 8-43.

Note

In the default stack configuration, the Candidate **Auto Join** parameter is enabled, but the Commander **Auto Grab** parameter is disabled. This prevents Candidates from automatically joining a stack prematurely or joining the wrong stack (if more than one stack Commander is configured in a subnet or broadcast domain). If you plan to install more than one stack in a subnet, HP recommends that you leave **Auto Grab** disabled on all Commander switches and manually add Members to their stacks. Similarly, if you plan to install a stack in a subnet (broadcast domain) where stacking-capable switches are not intended for stack membership, you should set the **Stack State** parameter (in the Stack Configuration screen) to **Disabled** on those particular switches.

Configuring Stack Management

Overview of Configuring and Bringing Up a Stack

This process assumes that:

- All switches you want to include in a stack are connected to the same subnet (broadcast domain).
- If VLANs are enabled on the switches you want to include in the stack, then the ports linking the stacked switches must be on the primary VLAN in each switch (which, in the default configuration, is the default VLAN). If the primary VLAN is tagged, then each switch in the stack must use the same VLAN ID (VID) for the primary VLAN. (Refer to "The Primary VLAN" on page 2-45, and "Stacking Operation with Multiple VLANs Configured" on page 8-44.)
- *If you are including a ProCurve Switch 8000M, 4000M, 2424M, 2400M, or 1600M in a stack, you must first update all such devices to software version C.08.03 or later.* (You can get a copy of the latest software version from the ProCurve Networking web site and/or copy it from one switch to another. For downloading instructions, see appendix A, "File Transfers", in the *Management and Configuration Guide* for your switch.)

Options for Configuring a Commander and Candidates. Depending on how Commander and Candidate switches are configured, Candidates can join a stack either automatically or by a Commander manually adding ("pulling") them into the stack. In the default configuration, a Candidate joins only when *manually* pulled by a Commander. You can reconfigure a Commander to *automatically* pull in Candidates that are in the default stacking configuration. You can also reconfigure a Candidate switch to either "push" itself into a particular Commander's stack, convert the Candidate to a Commander (for a stack that does not already have a Commander), or to operate as a standalone switch without stacking. The following table shows your control options for adding Members to a stack.

Table 8-3. Stacking Configuration Guide

Join Method[1]	**Commander (IP Addressing Required)**	**Candidate (IP Addressing Optional)**	
	Auto Grab	**Auto Join**	**Passwords**
Automatically add Candidate to Stack (Causes the first 15 eligible, discovered switches in the subnet to automatically join a stack.)	**Yes**	**Yes** _(default)_	No _(default)_*
Manually add Candidate to Stack (Prevent automatic joining of switches you don't want in the stack)	**No** _(default)_	**Yes** _(default)_	Optional*
	Yes	**No**	Optional*
	Yes	**Yes** _(default)_ or **No**	Configured
Prevent a switch from being a Candidate	**N/A**	**Disabled**	Optional

*The Commander's Manager and Operator passwords propagate to the candidate when it joins the stack.

The easiest way to *automatically* create a stack is to:

1. Configure a switch as a Commander.
2. Configure IP addressing and a stack name on the Commander.
3. Set the Commander's **Auto Grab** parameter to **Yes**.
4. Connect Candidate switches (in their factory default configuration) to the network.

This approach automatically creates a stack of up to 16 switches (including the Commander). However this replaces manual control with an automatic process that may bring switches into the stack that you did not intend to include. With the Commander's **Auto Grab** parameter set to **Yes**, *any switch* conforming to all four of the following factors automatically becomes a stack Member:

- Default stacking configuration (**Stack State** set to **Candidate**, and **Auto Join** set to **Yes**)

- Same subnet (broadcast domain) and default VLAN as the Commander (If VLANs are used in the stack environment, see "Stacking Operation with a Tagged VLAN" on page 8-44.)

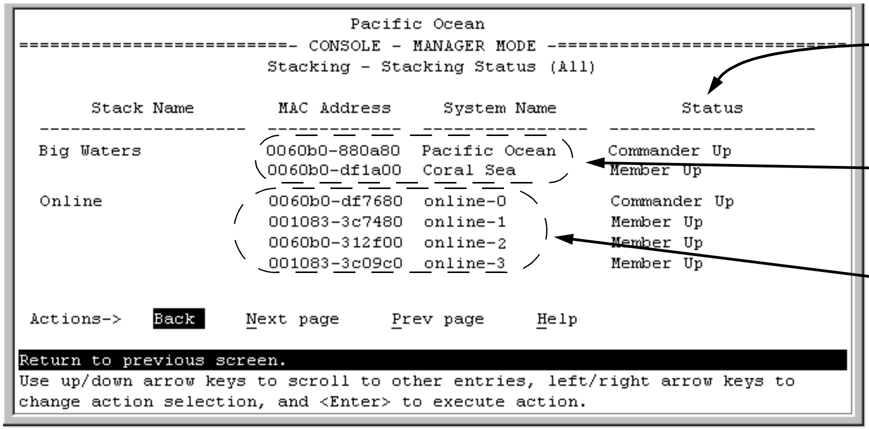
- No Manager password

- 14 or fewer stack members at the moment

General Steps for Creating a Stack

This section describes the general stack creation process. For the detailed configuration processes, see pages 8-13 through 8-36 for the menu interface and pages 8-29 through 8-41 for the CLI.

1. Determine the naming conventions for the stack. You will need a stack name. Also, to help distinguish one switch from another in the stack, you can configure a unique system name for each switch. Otherwise, the system name for a switch appearing in the Stacking Status screen appears as the stack name plus an automatically assigned switch number. For example:



```
                              Pacific Ocean
=========================- CONSOLE - MANAGER MODE -===========================
                          Stacking - Stacking Status (All)

      Stack Name          MAC Address     System Name           Status
  --------------------   -------------  ---------------  --------------------
  Big Waters             0060b0-880a80  Pacific Ocean    Commander Up
                         0060b0-df1a00  Coral Sea        Member Up

  Online                 0060b0-df7680  online-0         Commander Up
                         001083-3c7480  online-1         Member Up
                         0060b0-312f00  online-2         Member Up
                         001083-3c09c0  online-3         Member Up


 Actions->   Back     Next page     Prev page     Help

Return to previous screen.
Use up/down arrow keys to scroll to other entries, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

For status descriptions, see the table on page 8-45.

Stack with unique system name for each switch.

Stack named "Online" with no previously configured system names assigned to individual switches.

Figure 8-4. Using the System Name to Help Identify Individual Switches

2. Configure the Commander switch. Doing this first helps to establish consistency in your stack configuration, which can help prevent startup problems.

   - A stack requires one Commander switch. If you plan to implement more than one stack in a subnet (broadcast domain), the easiest way to avoid unintentionally adding a Candidate to the wrong stack is to manually control the joining process by leaving the Commander's **Auto Grab** parameter set to **No** (the default).
   - The Commander assigns its Manager and Operator passwords to any Candidate switch that joins the stack.
   - The Commander's SNMP community names apply to members.

3. For automatically or manually pulling Candidate switches into a stack, you can leave such switches in their default stacking configuration. If you need to access Candidate switches through your network before they join the stack, assign IP addresses to these devices. Otherwise, IP addressing is optional for Candidates and Members. (Note that once a Candidate becomes a member, you can access it through the Commander to assign IP addressing or make other configuration changes.)

4. Make a record of any Manager passwords assigned to the switches (intended for your stack) that are not currently members. (You will use these passwords to enable the protected switches to join the stack.)

5. If you are using VLANs in the stacking environment, you must use the default VLAN for stacking links. For more information, see "Stacking Operation with a Tagged VLAN" on page 8-44.

6. Ensure that all switches intended for the stack are connected to the same subnet (broadcast domain). As soon as you connect the Commander, it will begin discovering the available Candidates in the subnet.

   - If you configured the Commander to automatically add Members (**Auto Grab** = **Yes**), the first fifteen discovered Candidates meeting both of the following criteria will automatically join the stack:
     – **Auto Join** parameter set to **Yes** (the default)
     – Manager password not configured
   - If you configured the Commander to manually add Members (**Auto Grab** set to **No**—the default), you can begin the process of selecting and adding the desired Candidates.

7. Ensure that all switches intended for the stack have joined.

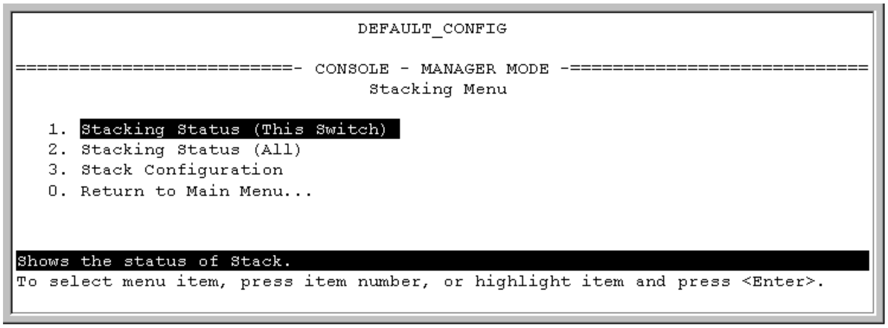
8. If you need to do specific configuration or monitoring tasks on a Member, use the console interface on the Commander to access the Member.

Using the Menu Interface To View Stack Status and Configure Stacking

Using the Menu Interface To View and Configure a Commander Switch

1. Configure an IP address and subnet mask on the Commander switch. (Refer to the *Management and Configuration Guide* for your switch.)

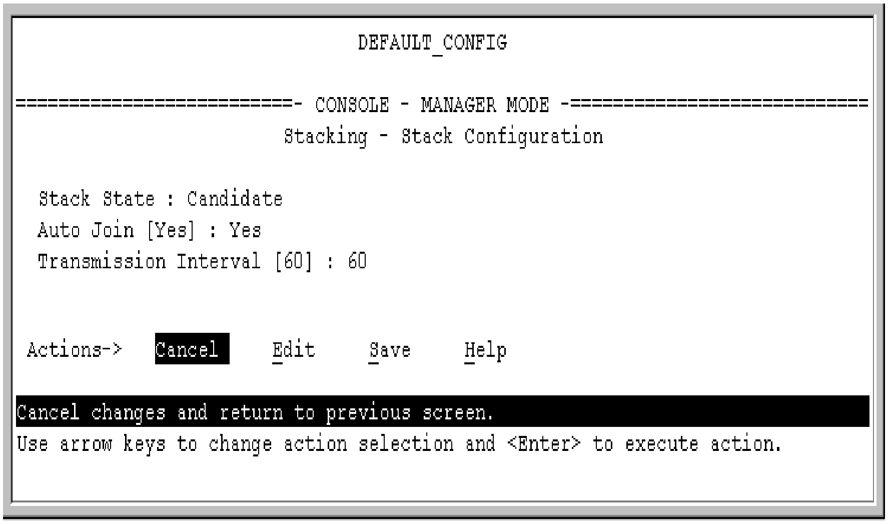
2. Display the Stacking Menu by selecting **Stacking** in the Main Menu.



```
                                DEFAULT_CONFIG

==========================- CONSOLE - MANAGER MODE -============================
                                 Stacking Menu

   1. Stacking Status (This Switch)
   2. Stacking Status (All)
   3. Stack Configuration
   0. Return to Main Menu...


Shows the status of Stack.
To select menu item, press item number, or highlight item and press <Enter>.
```

Figure 8-5. The Default Stacking Menu

3. Display the Stack Configuration menu by pressing **[3]** to select **Stack Configuration**.

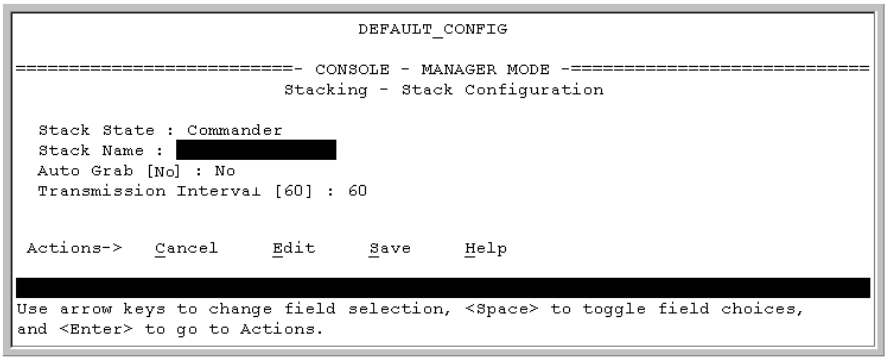


```
                                DEFAULT_CONFIG

==========================- CONSOLE - MANAGER MODE -============================
                          Stacking - Stack Configuration

  Stack State : Candidate
  Auto Join [Yes] : Yes
  Transmission Interval [60] : 60


 Actions->  Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 8-6. The Default Stack Configuration Screen

4. Move the cursor to the Stack State field by pressing **[E]** (for **Edit**). Then use the Space bar to select the **Commander** option.

5. Press the downarrow key to display the Commander configuration fields in the Stack Configuration screen.



```
                                 DEFAULT_CONFIG

==========================- CONSOLE - MANAGER MODE -============================
                         Stacking - Stack Configuration

  Stack State : Commander
  Stack Name :
  Auto Grab [No] : No
  Transmission Interval [60] : 60


 Actions->   Cancel     Edit     Save     Help


Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

Figure 8-7. The Default Commander Configuration in the Stack Configuration Screen

6. Enter a unique stack name (up to 15 characters; no spaces) and press the downarrow key.

7. Ensure that the Commander has the desired **Auto Grab** setting, then press the downarrow key:

   - **No** (the default) prevents automatic joining of Candidates that have their **Auto Join** set to **Yes**.
   - **Yes** enables the Commander to automatically take a Candidate into the stack as a Member if the Candidate has **Auto Join** set to **Yes (**the default Candidate setting) and does not have a previously configured password.

8. Accept or change the transmission interval (default: 60 seconds), then press **[Enter]** to return the cursor to the **Actions** line.

9. Press **[S]** (for **Save**) to save your configuration changes and return to the Stacking menu.

Your Commander switch should now be ready to automatically or manually acquire Member switches from the list of discovered Candidates, depending on your configuration choices.

Using the Menu To Manage a Candidate Switch

Using the menu interface, you can perform these actions on a Candidate switch:

- Add ("push") the Candidate into an existing stack
- Modify the Candidate's stacking configuration (**Auto Join** and **Transmission Interval**)
- Convert the Candidate to a Commander
- Disable stacking on the Candidate so that it operates as a standalone switch

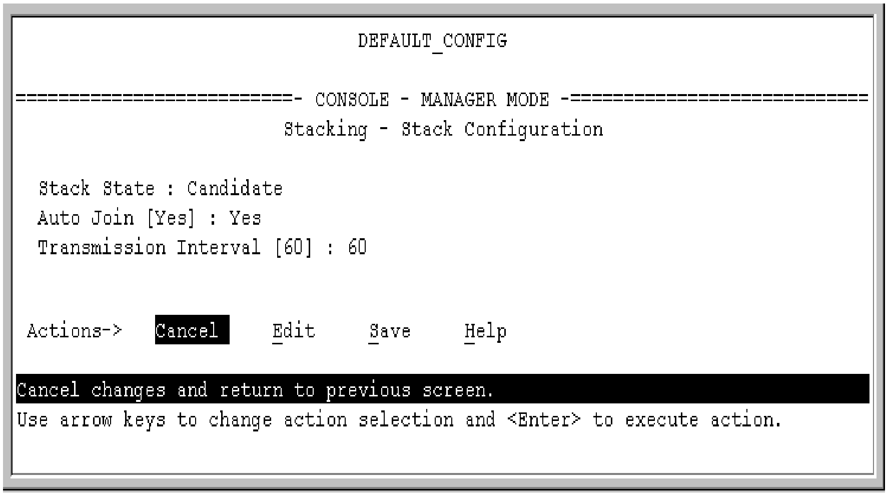
In its default stacking configuration, a Candidate switch can either automatically join a stack or be manually added ("pulled") into a stack by a Commander, depending on the Commander's **Auto Grab** setting. The following table lists the Candidate's configuration options:

Table 8-4. Candidate Configuration Options in the Menu Interface

Parameter	Default Setting	Other Settings
Stack State	Candidate	Commander, Member, or Disabled
Auto Join	Yes	No
Transmission Interval	60 Seconds	Range: 1 to 300 seconds

Using the Menu To "Push" a Switch Into a Stack, Modify the Switch's Configuration, or Disable Stacking on the Switch. Use Telnet or the web browser interface to access the Candidate if it has an IP address. Otherwise, use a direct connection from a terminal device to the switch's console port. (For information on how to use the web browser interface, see the online Help provided for the browser.)

1. Display the Stacking Menu by selecting **Stacking** in the console Main Menu.
2. Display the Stack Configuration menu by pressing **[3]** to select **Stack Configuration**.



```
                              DEFAULT_CONFIG

=========================- CONSOLE - MANAGER MODE -===========================
                        Stacking - Stack Configuration

  Stack State : Candidate
  Auto Join [Yes] : Yes
  Transmission Interval [60] : 60


 Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 8-8. The Default Stack Configuration Screen

3. Move the cursor to the Stack State field by pressing **[E]** (for **Edit**).

4. Do one of the following:

   - To disable stacking on the Candidate, use the Space bar to select the **Disabled** option, then go to step 5.

     Note: Using the menu interface to disable stacking on a Candidate removes the Candidate from all stacking menus.

   - To insert the Candidate into a specific Commander's stack:

     i. Use the space bar to select Member.

     ii. Press **[Tab]** once to display the **Commander MAC Address** parameter, then enter the MAC address of the desired Commander.

   - To change **Auto Join** or **Transmission Interval**, use **[Tab]** to select the desired parameter, and:

     – To change **Auto Join**, use the Space bar.

     – To change **Transmission Interval**, type in the new value in the range of 1 to 300 seconds.

     Note: All switches in the stack must be set to the same transmission interval to help ensure proper stacking operation. HP recommends that you leave this parameter set to the default 60 seconds.

   Then go to step 5.

5. press **[Enter]** to return the cursor to the **Actions** line.

6. Press **[S]** (for **Save**) to save your configuration changes and return to the Stacking menu.

Using the Commander To Manage The Stack

The Commander normally operates as your stack manager and point of entry into other switches in the stack. This typically includes:

- Adding new stack members
- Moving members between stacks
- Removing members from a stack
- Accessing stack members for individual configuration changes and traffic monitoring

The Commander also imposes its passwords on all stack members and provides SNMP community membership to the stack. (See "SNMP Community Operation in a Stack" on page 8-43.)

Using the Commander's Menu To Manually Add a Candidate to a Stack. In the default configuration, you must manually add stack Members from the Candidate pool. Reasons for a switch remaining a Candidate instead of becoming a Member include any of the following:

- **Auto Grab** in the Commander is set to **No** (the default).
- **Auto Join** in the Candidate is set to **No**.

  Note: When a switch leaves a stack and returns to Candidate status, its **Auto Join** parameter resets to **No** so that it will not immediately rejoin a stack from which it has just departed.
- A Manager password is set in the Candidate.
- The stack is full.

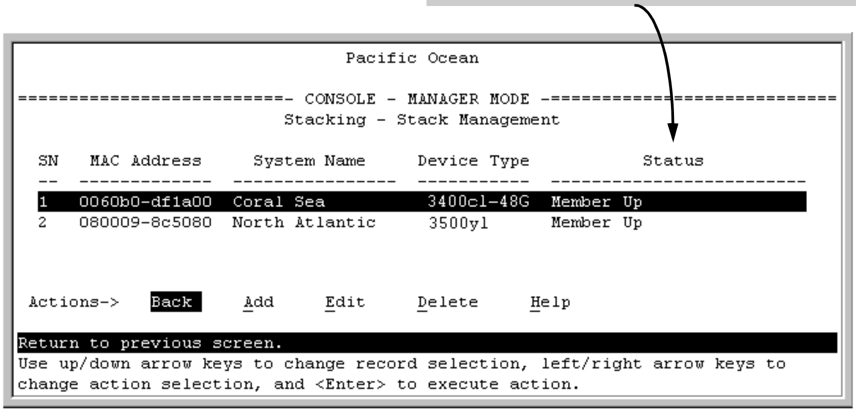
Unless the stack is already full, you can use the Stack Management screen to manually convert a Candidate to a Member. If the Candidate has a Manager password, you will need to use it to make the Candidate a Member of the stack.

1. To add a Member, start at the Main Menu and select:

   9. Stacking...

   4. Stack Management

   You will then see the Stack Management screen:

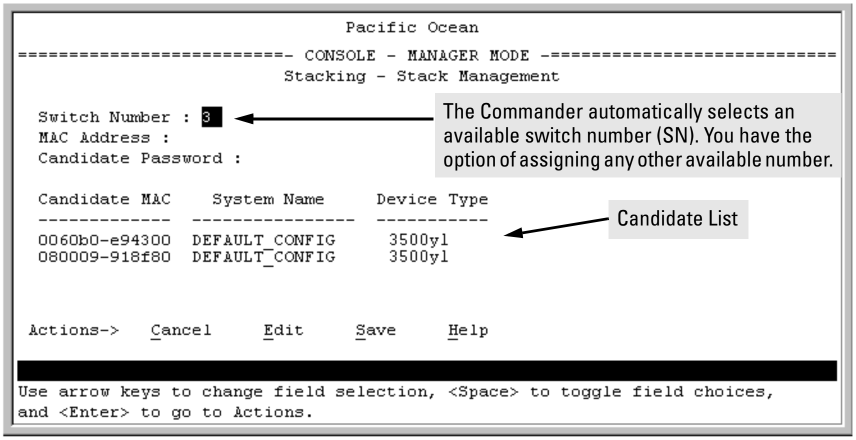


For status descriptions, see the table on page 8-45.

```
                                 Pacific Ocean

==========================- CONSOLE - MANAGER MODE -============================
                          Stacking - Stack Management

  SN   MAC Address     System Name     Device Type          Status
  --  -------------  ---------------  -----------  -------------------------
  1   0060b0-df1a00  Coral Sea         3400cl-48G  Member Up
  2   080009-8c5080  North Atlantic    3500yl      Member Up



 Actions->    Back     Add      Edit      Delete     Help

Return to previous screen.
Use up/down arrow keys to change record selection, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Figure 8-9. Example of the Stack Management Screen

2. Press **[A]** (for **Add**) to add a Candidate. You will then see this screen listing the available Candidates:



```
                                 Pacific Ocean
==========================- CONSOLE - MANAGER MODE -============================
                          Stacking - Stack Management

  Switch Number : 3
  MAC Address :
  Candidate Password :

  Candidate MAC    System Name     Device Type
  -------------  ---------------  -----------
  0060b0-e94300  DEFAULT_CONFIG     3500yl
  080009-918f80  DEFAULT_CONFIG     3500yl


 Actions->   Cancel     Edit     Save     Help


Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

The Commander automatically selects an available switch number (SN). You have the option of assigning any other available number.

Candidate List

Figure 8-10. Example of Candidate List in Stack Management Screen

3. Either accept the displayed switch number or enter another available number. (The range is 0 - 15, with 0 reserved for the Commander.)

4. Use the downarrow key to move the cursor to the MAC Address field, then type the MAC address of the desired Candidate from the Candidate list in the lower part of the screen.

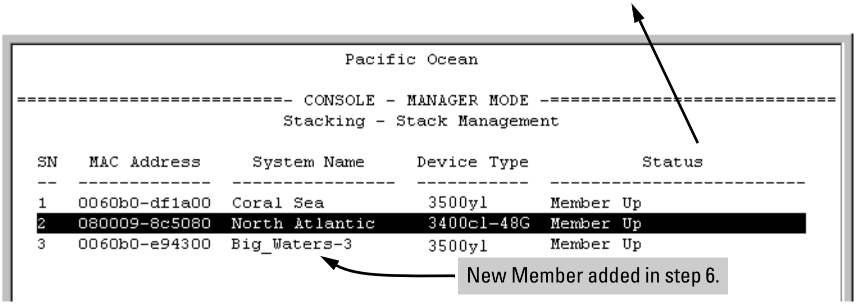
5. Do one of the following:

- If the desired Candidate has a Manager password, press the downarrow key to move the cursor to the Candidate Password field, then type the password.
- If the desired Candidate does not have a password, go to step 6.

6. Press **[Enter]** to return to the Actions line, then press **[S]** (for **Save**) to complete the Add process for the selected Candidate. You will then see a screen similar to the one in figure 8-11, below, with the newly added Member listed.

   Note: If the message **Unable to add stack member: Invalid Password** appears in the console menu's Help line, then you either omitted the Candidate's Manager password or incorrectly entered the Manager password.

For status descriptions, see the table on page 8-45.



```
                              Pacific Ocean

==========================- CONSOLE - MANAGER MODE -=============================
                         Stacking - Stack Management

  SN   MAC Address     System Name      Device Type           Status
  --   -------------   ----------------  -----------   -------------------------
  1    0060b0-df1a00   Coral Sea          3500yl       Member Up
  2    080009-8c5080   North Atlantic     3400cl-48G   Member Up
  3    0060b0-e94300   Big_Waters-3       3500yl       Member Up
```

New Member added in step 6.

Figure 8-11. Example of Stack Management Screen After New Member Added

Using the Commander's Menu To Move a Member From One Stack to Another. Where two or more stacks exist in the same subnet (broadcast domain), you can easily move a Member of one stack to another stack if the destination stack is not full. (If you are using VLANs in your stack environment, see "Stacking Operation with a Tagged VLAN" on page 8-44.) This procedure is nearly identical to manually adding a Candidate to a stack (page 8-17). (If the stack from which you want to move the Member has a Manager password, you will need to know the password to make the move.)

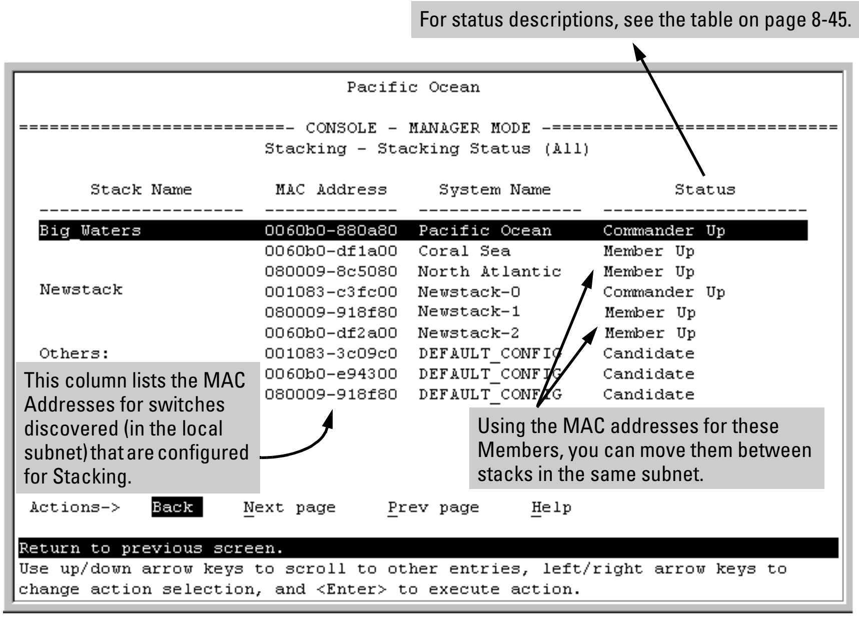
1. To move a Member from one stack to another, go to the Main Menu of the Commander in the destination stack and display the Stacking Menu by selecting

   9. Stacking...

2. To learn or verify the MAC address of the Member you want to move, display a listing of all Commanders, Members, and Candidates in the subnet by selecting:

2. Stacking Status (All)

You will then see the Stacking Status (All) screen:



For status descriptions, see the table on page 8-45.

```
                               Pacific Ocean

========================- CONSOLE - MANAGER MODE -============================
                         Stacking - Stacking Status (All)

     Stack Name        MAC Address     System Name          Status
  -------------------- -------------  ---------------  --------------------
  Big_Waters           0060b0-880a80  Pacific Ocean    Commander Up
                       0060b0-df1a00  Coral Sea        Member Up
                       080009-8c5080  North Atlantic   Member Up
  Newstack             001083-c3fc00  Newstack-0       Commander Up
                       080009-918f80  Newstack-1       Member Up
                       0060b0-df2a00  Newstack-2       Member Up
  Others:              001083-3c09c0  DEFAULT_CONFIG   Candidate
                       0060b0-e94300  DEFAULT_CONFIG   Candidate
                       080009-918f80  DEFAULT_CONFIG   Candidate

 Actions->    Back     Next page     Prev page     Help

Return to previous screen.
Use up/down arrow keys to scroll to other entries, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

This column lists the MAC Addresses for switches discovered (in the local subnet) that are configured for Stacking.

Using the MAC addresses for these Members, you can move them between stacks in the same subnet.

Figure 8-12. Example of How the Stacking Status (All) Screen Helps You Find Member MAC Addresses

3. In the Stacking Status (All) screen, find the Member switch that you want to move and note its MAC address, then press **[B]** (for **Back**) to return to the Stacking Menu.

4. Display the Commander's Stack Management screen by selecting

   4. Stack Management

   (For an example of this screen, see figure 8-9 on page 8-18.)

5. Press **[A]** (for **Add**) to add the Member. You will then see a screen listing any available candidates. (See figure 8-10 on page 8-18.) Note that you will not see the switch you want to add because it is a Member of another stack and not a Candidate.)

6. Either accept the displayed switch number or enter another available number. (The range is 0 - 15, with 0 reserved for the Commander.)

7. Use the downarrow key to move the cursor to the MAC Address field, then type the MAC address of the desired Member you want to move from another stack.

8. Do one of the following:

   - If the stack containing the Member you are moving has a Manager password, press the downarrow key to select the Candidate Password field, then type the password.
   - If the stack containing the Member you want to move does not have a password, go to step 9.

9. Press **[Enter]** to return to the Actions line, then press **[S]** (for **Save**) to complete the Add process for the selected Member. You will then see a screen similar to the one in figure 8-9 on page 8-18, with the newly added Member listed.

Note: If the message **Unable to add stack member: Invalid Password** appears in the console menu's Help line, then you either omitted the Manager password for the stack containing the Member or incorrectly entered the Manager password.

You can "push" a Member from one stack to another by going to the Member's interface and entering the MAC address of the destination stack Commander in the Member's **Commander MAC Address** field. Using this method moves the Member to another stack without a need for knowing the Manager password in that stack, but also blocks access to the Member from the original Commander.

Using the Commander's Menu To Remove a Stack Member. These rules affect removals from a stack:

- When a Candidate becomes a Member, its **Auto Join** parameter is automatically set to **No**. This prevents the switch from automatically rejoining a stack as soon as you remove it from the stack.
- When you use the Commander to remove a switch from a stack, the switch rejoins the Candidate pool for your IP subnet (broadcast domain), with **Auto Join** set to **No**.
- When you remove a Member from a stack, it frees the previously assigned switch number (**SN**), which then becomes available for assignment to another switch that you may subsequently add to the stack. The default switch number used for an add is the lowest unassigned number in the Member range (1 - 15; 0 is reserved for the Commander).

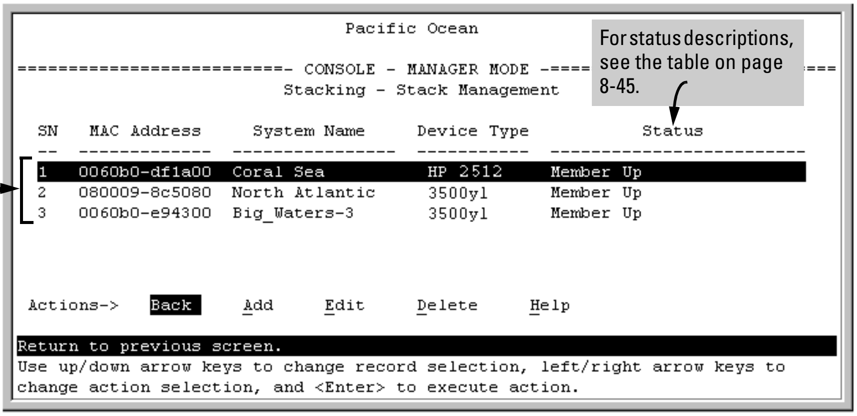
To remove a Member from a stack, use the Stack Management screen.

1. From the Main Menu, select:

   9. Stacking...

   4. Stack Management

   You will then see the Stack Management screen:



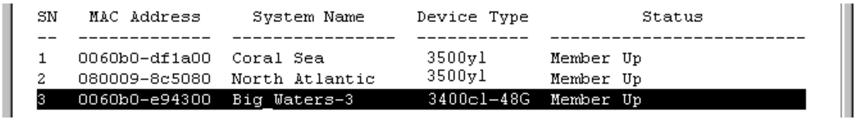
```
                              Pacific Ocean

=========================- CONSOLE - MANAGER MODE -====                    ===
                        Stacking - Stack Management

  SN   MAC Address     System Name     Device Type          Status
  --   -------------  ----------------  -----------  -------------------------
  1    0060b0-df1a00  Coral Sea          HP 2512     Member Up
  2    080009-8c5080  North Atlantic     3500yl      Member Up
  3    0060b0-e94300  Big_Waters-3       3500yl      Member Up



 Actions->   Back     Add     Edit     Delete     Help

Return to previous screen.
Use up/down arrow keys to change record selection, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

For status descriptions, see the table on page 8-45.

Stack Member List

Figure 8-13. Example of Stack Management Screen with Stack Members Listed

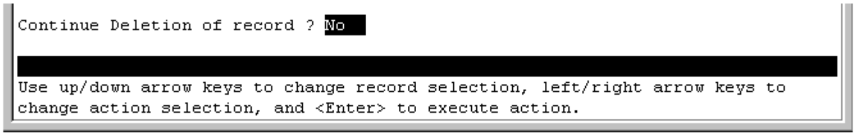
2. Use the downarrow key to select the Member you want to remove from the stack.



```
 SN   MAC Address     System Name     Device Type          Status
 --   -------------  ----------------  -----------  -------------------------
 1    0060b0-df1a00  Coral Sea          3500yl      Member Up
 2    080009-8c5080  North Atlantic     3500yl      Member Up
 3    0060b0-e94300  Big_Waters-3       3400cl-48G  Member Up
```

Figure 8-14. Example of Selecting a Member for Removal from the Stack

3. Type **[D]** (for **Delete**) to remove the selected Member from the stack. You will then see the following prompt:



```
Continue Deletion of record ? No

Use up/down arrow keys to change record selection, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Figure 8-15. The Prompt for Completing the Deletion of a Member from the Stack

4. To continue deleting the selected Member, press the Space bar once to select **Yes** for the prompt, then press **[Enter]** to complete the deletion. The Stack Management screen updates to show the new stack Member list.

Using the Commander To Access Member Switches for Configuration Changes and Monitoring Traffic

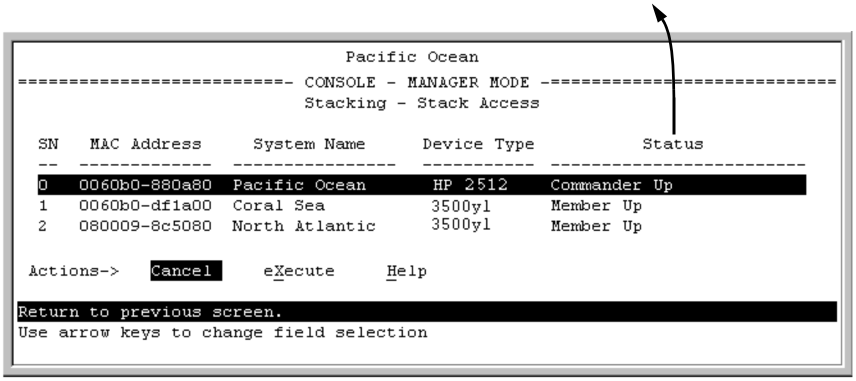
After a Candidate becomes a stack Member, you can use that stack's Commander to access the Member's console interface for the same configuration and monitoring that you would do through a Telnet or direct-connect access.

1. From the Main Menu, select:

   9. Stacking...
   5. Stack Access

   You will then see the Stack Access screen:

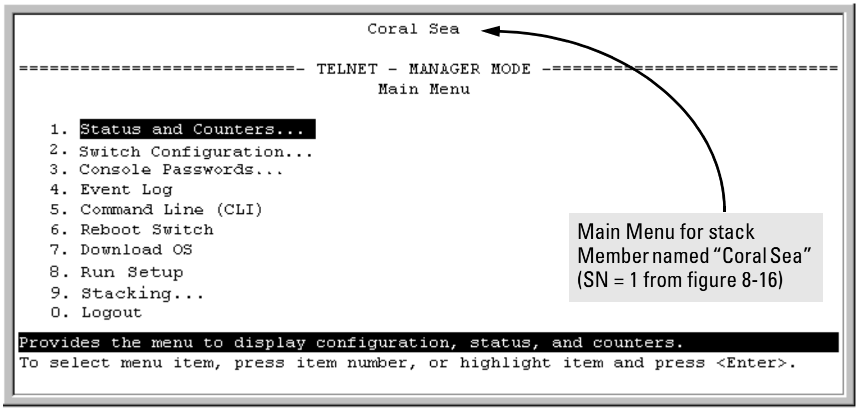
For status descriptions, see the table on page 8-45.



```
                              Pacific Ocean
==========================- CONSOLE - MANAGER MODE -============================
                          Stacking - Stack Access

  SN   MAC Address      System Name      Device Type          Status
  --   -------------   ----------------  -----------   -------------------------
  0    0060b0-880a80   Pacific Ocean      HP 2512      Commander Up
  1    0060b0-df1a00   Coral Sea         3500yl        Member Up
  2    080009-8c5080   North Atlantic    3500yl        Member Up

 Actions->   Cancel     eXecute     Help

Return to previous screen.
Use arrow keys to change field selection
```

Figure 8-16. Example of the Stack Access Screen

Use the down arrow key to select the stack Member you want to access, then press **[X]** (for **eXecute**) to display the console interface for the selected Member. For example, if you selected switch number 1 (system name: **Coral Sea**) in figure 8-16 and then pressed **[X]**, you would see the Main Menu for the switch named Coral Sea.



```
                                    Coral Sea

==========================- TELNET - MANAGER MODE -=============================
                                    Main Menu

     1. Status and Counters...
     2. Switch Configuration...
     3. Console Passwords...
     4. Event Log
     5. Command Line (CLI)
     6. Reboot Switch
     7. Download OS
     8. Run Setup
     9. Stacking...
     0. Logout

Provides the menu to display configuration, status, and counters.
To select menu item, press item number, or highlight item and press <Enter>.
```

Main Menu for stack Member named "Coral Sea" (SN = 1 from figure 8-16)

Figure 8-17. The eXecute Command Displays the Console Main Menu for the Selected Stack Member

2. You can now make configuration changes and/or view status data for the selected Member in the same way that you would if you were directly connected or telnetted into the switch.

3. When you are finished accessing the selected Member, do the following to return to the Commander's Stack Access screen:

   a. Return to the Member's Main Menu.

   b. Press **[0]** (for Logout), then **[Y]** (for Yes).

   c. Press **[Return]**.

   You should now see the Commander's Stack Access screen. (For an example, see figure 8-16 on page 8-23.)

Converting a Commander or Member to a Member of Another Stack

When moving a commander, the following procedure returns the stack members to Candidate status (with Auto-Join set to "**No**") and converts the stack Commander to a Member of another stack. When moving a member, the procedure simply pulls a Member out of one stack and pushes it into another.

1. From the Main Menu of the switch you want to move, select

   9. Stacking

2. To determine the MAC address of the destination Commander, select

   2. Stacking Status (All)

3. Press **[B]** (for **Back**) to return to the Stacking Menu.

4. To display Stack Configuration menu for the switch you are moving, select

   3. Stack Configuration

5. Press **[E]** (for **Edit**) to select the Stack State parameter.

6. Use the Space bar to select **Member**, then press [↓] to move to the **Commander MAC Address** field.

7. Enter the MAC address of the destination Commander and press **[Enter]**.

8. Press **[S]** (for **Save**).

Monitoring Stack Status

Using the stacking options in the menu interface for any switch in a stack, you can view stacking data for that switch or for all stacks in the subnet (broadcast domain). (If you are using VLANs in your stack environment, see "Stacking Operation with a Tagged VLAN" on page 8-44.) This can help you in such ways as determining the stacking configuration for individual switches, identifying stack Members and Candidates, and determining the status of individual switches in a stack. See table 8-5 on page 8-25.

Table 8-5. Stack Status Environments

Screen Name	Commander	Member	Candidate
Stack Status (This Switch)	• Commander's stacking configuration • Data on stack Members: – Switch Number – MAC Address – System Name – Device Type – Status	• Member's stacking configuration • Member Status • Data identifying Member's Commander: – Commander Status – Commander IP Address – Commander MAC Address	Candidate's stacking configuration
Stack Status (All)	Lists devices by stack name or Candidate status (if device is not a stack Member). Includes: • Stack Name • MAC Address • System Name • Status	Same as for Commander.	Same as for Commander.

Using Any Stacked Switch To View the Status for All Switches with Stacking Enabled. This procedure displays the general status of all switches in the IP subnet (broadcast domain) that have stacking enabled.

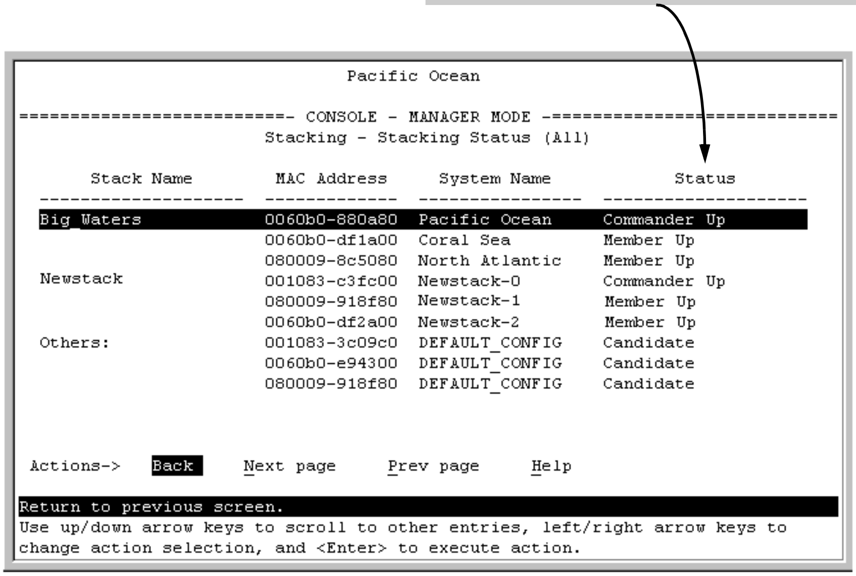
1. Go to the console Main Menu for any switch configured for stacking and select:

   9. Stacking ...

   2. Stacking Status (All)

   You will then see a Stacking Status screen similar to the following:

For status descriptions, see the table on page 8-45.



```
                              Pacific Ocean

==========================- CONSOLE - MANAGER MODE -============================
                        Stacking - Stacking Status (All)

     Stack Name          MAC Address      System Name            Status
  --------------------  -------------  ----------------  --------------------
  Big_Waters             0060b0-880a80  Pacific Ocean     Commander Up
                         0060b0-df1a00  Coral Sea         Member Up
                         080009-8c5080  North Atlantic    Member Up
  Newstack               001083-c3fc00  Newstack-0        Commander Up
                         080009-918f80  Newstack-1        Member Up
                         0060b0-df2a00  Newstack-2        Member Up
  Others:                001083-3c09c0  DEFAULT_CONFIG    Candidate
                         0060b0-e94300  DEFAULT_CONFIG    Candidate
                         080009-918f80  DEFAULT_CONFIG    Candidate



 Actions->   Back     Next page     Prev page     Help

Return to previous screen.
Use up/down arrow keys to scroll to other entries, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Figure 8-18. Example of Stacking Status for All Detected Switches Configured for Stacking

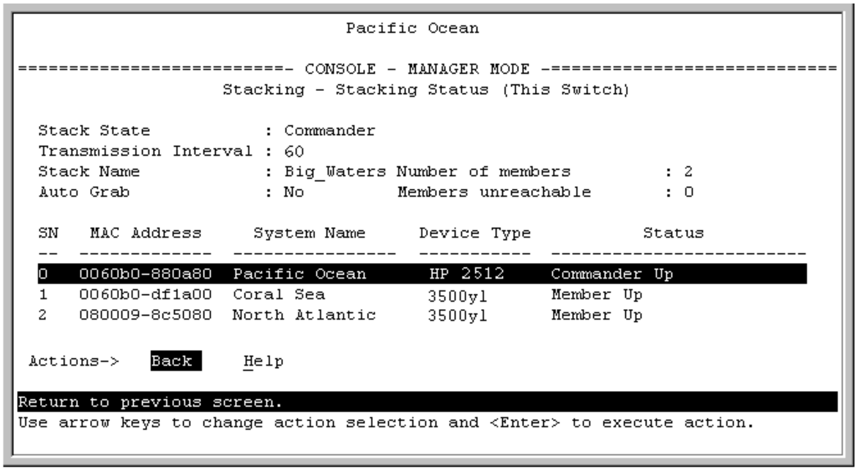
Viewing Commander Status. This procedure displays the Commander and stack configuration, plus information identifying each stack member.

To display the status for a Commander, go to the console Main Menu for the switch and select:

9. Stacking ...

1. Stacking Status (This Switch)

You will then see the Commander's Stacking Status screen:



```
                              Pacific Ocean

==========================- CONSOLE - MANAGER MODE -============================
                   Stacking - Stacking Status (This Switch)

  Stack State           : Commander
  Transmission Interval : 60
  Stack Name            : Big_Waters Number of members          : 2
  Auto Grab             : No         Members unreachable        : 0

  SN   MAC Address     System Name      Device Type          Status
  --  -------------  ----------------  -----------  -------------------------
  0   0060b0-880a80  Pacific Ocean      HP 2512     Commander Up
  1   0060b0-df1a00  Coral Sea          3500yl      Member Up
  2   080009-8c5080  North Atlantic     3500yl      Member Up

 Actions->   Back     Help

Return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 8-19. Example of the Commander's Stacking Status Screen

Viewing Member Status. This procedure displays the Member's stacking information plus the Commander's status, IP address, and MAC address.

To display the status for a Member:

1. Go to the console Main Menu of the Commander switch and select

   9. Stacking ...

   5. Stack Access

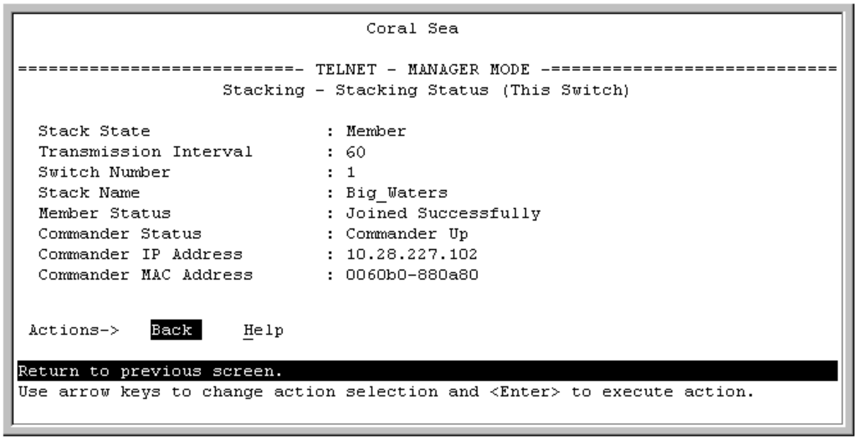
2. Use the downarrow key to select the Member switch whose status you want to view, then press **[X]** (for **eXecute**). You will then see the Main Menu for the selected Member switch.

3. In the Member's Main Menu screen, select

   9. Stacking ...

   1. Stacking Status (This Switch)

   You will then see the Member's Stacking Status screen:



```
                                  Coral Sea

===========================- TELNET - MANAGER MODE -============================
                    Stacking - Stacking Status (This Switch)

  Stack State                  : Member
  Transmission Interval        : 60
  Switch Number                : 1
  Stack Name                   : Big_Waters
  Member Status                : Joined Successfully
  Commander Status             : Commander Up
  Commander IP Address         : 10.28.227.102
  Commander MAC Address        : 0060b0-880a80


 Actions->  Back    Help

Return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 8-20. Example of a Member's Stacking Status Screen

Viewing Candidate Status. This procedure displays the Candidate's stacking configuration.

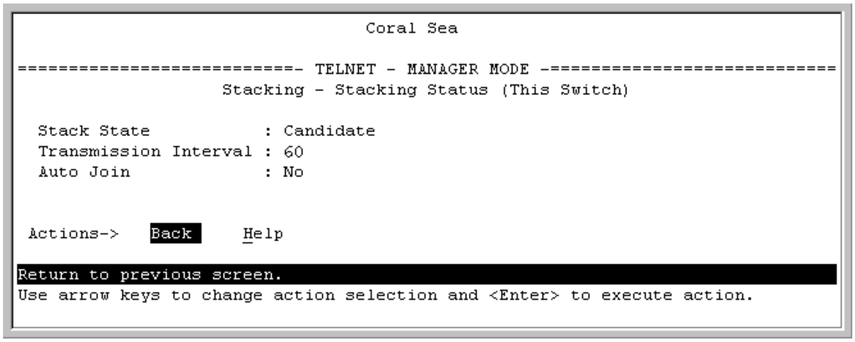
To display the status for a Candidate:

1. Use Telnet (if the Candidate has a valid IP address for your network) or a direct serial port connection to access the menu interface Main Menu for the Candidate switch and select

   9. Stacking ...

   1. Stacking Status (This Switch)

   You will then see the Candidate's Stacking Status screen:



```
                                  Coral Sea

===========================- TELNET - MANAGER MODE -============================
                    Stacking - Stacking Status (This Switch)

  Stack State           : Candidate
  Transmission Interval : 60
  Auto Join             : No


 Actions->  Back    Help

Return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 8-21. Example of a Candidate's Stacking Screen

Using the CLI To View Stack Status and Configure Stacking

The CLI enables you to do all of the stacking tasks available through the menu interface.)

Table 8-6. CLI Commands for Configuring Stacking on a Switch

CLI Command	Operation
show stack [candidates | view | all]	**Commander:** Shows Commander's stacking configuration and lists the stack members and their individual status. **Member:** Lists Member's stacking configuration and status, and the status and the IP address and subnet mask of the stack Commander. Options: **candidates:** (Commander only) Lists stack Candidates. **view:** (Commander only) Lists current stack Members and their individual status. **all:** Lists all stack Commanders, Members and Candidates, with their individual status.
[no] stack	**Any Stacking-Capable Switch:** Enables or disables stacking on the switch. **Default:** Stacking Enabled
[no] stack commander *<stack name>*	**Candidate or Commander:** Converts a Candidate to a Commander or changes the stack name of an existing commander. "**No**" form eliminates named stack and returns Commander and stack Members to Candidate status with **Auto Join** set to **No**. "**No**" form prevents the switch from being discovered as a stacking-capable switch. **Default:** Switch Configured as a Candidate
[no] stack auto-grab	**Commander:** Causes Commander to automatically add to its stack any discovered Candidate in the subnet that does not have a Manager password and has **Auto-Join** set to **Yes**. **Default:** Disabled **Note:** If the Commander's stack already has 15 members, the Candidate cannot join until an existing member leaves the stack.

CLI Command	Operation
[no] stack member *<switch-num>* mac-address *<mac-addr>* [password *<password-str>*]	**Commander:** Adds a Candidate to stack membership. "No" form removes a Member from stack membership. To easily determine the MAC address of a Candidate, use the **show stack candidates** command. To determine the MAC address of a Member you want to remove, use the **show stack view** command.The password (*password-str*) is required only when adding a Candidate that has a Manager password.
telnet *<1..15>* *Used In:* Commander Only	**Commander:** Uses the **SN** (switch number— assigned by the stack Commander) to access the console interface (menu interface or CLI) of a stack member. To view the list of **SN** assignments for a stack, execute the **show stack** command in the Commander's CLI.
[no] stack join *<mac-addr>*	**Candidate:** Causes the Candidate to join the stack whose Commander has the indicated MAC address. "No" form is used in a Member to remove it from the stack of the Commander having the specified address. **Member:** "Pushes" the member to another stack whose Commander has the indicated MAC address.
[no] stack auto-join	**Candidate:** Enables Candidate to automatically join the stack of any Commander in the IP subnet that has **Auto Grab** enabled, or disables **Auto-Join** in the candidate. **Default: Auto Join** enabled. **Note:** If the Candidate has a Manager password or if the available stack(s) already have the maximum of 15 Members, the automatic join will not occur.
stack transmission-interval	**All Stack Members:** specifies the interval in seconds for transmitting stacking discovery packets. **Default:** 60 seconds

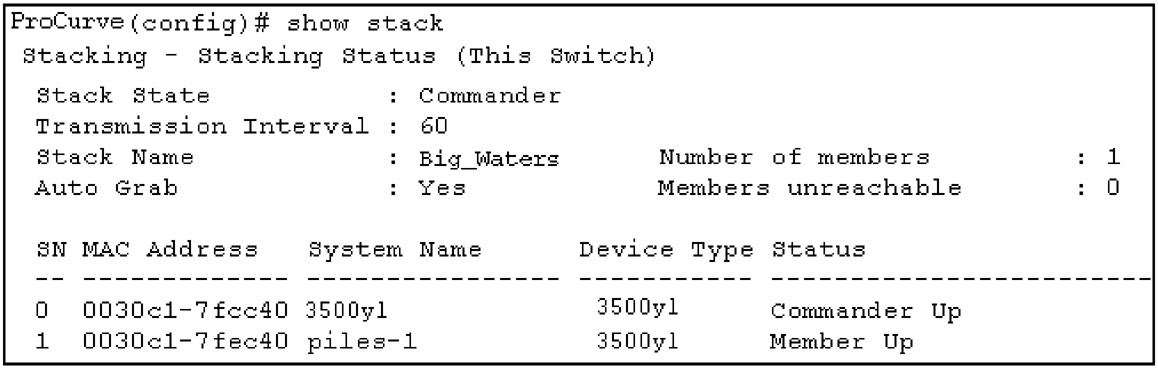
Using the CLI To View Stack Status

You can list the stack status for an individual switch and for other switches that have been discovered in the same subnet.

Syntax: show stack [candidates | view | all]

Viewing the Status of an Individual Switch. The following example illustrates how to use the CLI in a to display the stack status for that switch. In this case, the switch is in the default stacking configuration.

Syntax: show stack



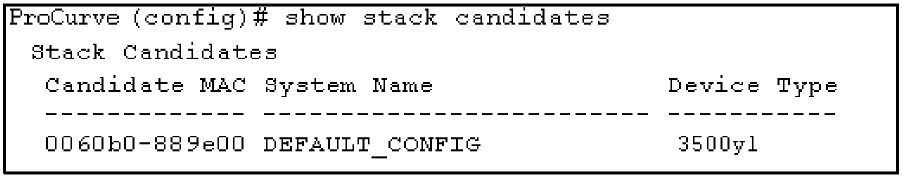
```
ProCurve(config)# show stack
 Stacking - Stacking Status (This Switch)
  Stack State            : Commander
  Transmission Interval  : 60
  Stack Name             : Big_Waters      Number of members        : 1
  Auto Grab              : Yes             Members unreachable      : 0

  SN MAC Address   System Name      Device Type Status
  -- ------------- ---------------- ----------- ------------------------
  0  0030c1-7fcc40 3500yl            3500yl     Commander Up
  1  0030c1-7fec40 piles-1           3500yl     Member Up
```

Figure 8-22. Example of Using the Show Stack Command To List the Stacking Configuration for an Individual Switch

Viewing the Status of Candidates the Commander Has Detected.

This example illustrates how to list stack candidates the Commander has discovered in the ip subnet (broadcast domain).

Syntax: show stack candidates

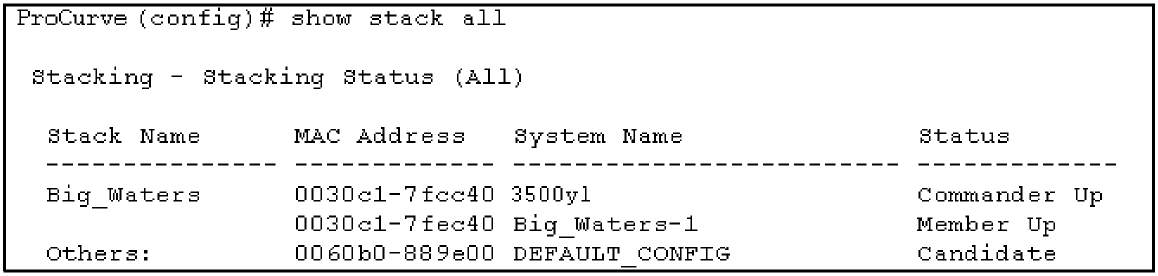


```
ProCurve (config)# show stack candidates
 Stack Candidates
  Candidate MAC System Name                Device Type
  ------------- ------------------------- -----------
  0060b0-889e00 DEFAULT_CONFIG              3500yl
```

Figure 8-23. Example of Using the Show Stack Candidates Command To List Candidates

Viewing the Status of all Stack-Enabled Switches Discovered in the IP Subnet. The next example lists all the stack-configured switches discovered in the IP subnet. Because the switch on which the **show stack all** command was executed is a candidate, it is included in the “Others” category.

Syntax: show stack all

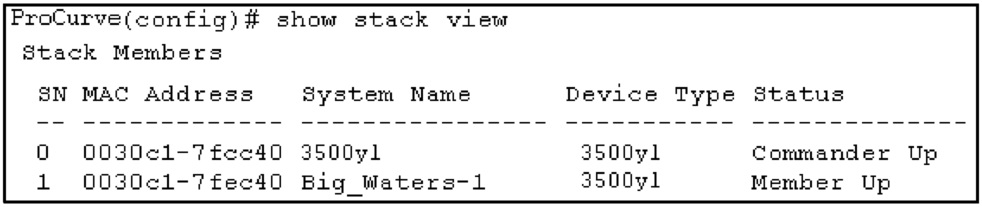


```
ProCurve (config)# show stack all

 Stacking - Stacking Status (All)

  Stack Name      MAC Address   System Name               Status
  --------------- ------------- ------------------------- -------------
  Big_Waters      0030c1-7fcc40 3500yl                    Commander Up
                  0030c1-7fec40 Big_Waters-1              Member Up
  Others:         0060b0-889e00 DEFAULT_CONFIG            Candidate
```

Figure 8-24. Result of Using the Show Stack All Command To List Discovered Switches in the IP Subnet

Viewing the Status of the Commander and Current Members of the Commander’s Stack. The next example lists all switches in the stack of the selected switch.

Syntax: show stack view



```
ProCurve(config)# show stack view
 Stack Members

  SN MAC Address   System Name      Device Type Status
  -- ------------- ---------------- ----------- --------------
  0  0030c1-7fcc40 3500yl            3500yl     Commander Up
  1  0030c1-7fec40 Big_Waters-1      3500yl     Member Up
```

Figure 8-25. Example of the Show Stack View Command To List the Stack Assigned to the Selected Commander

Using the CLI To Configure a Commander Switch

You can configure any stacking-enabled switch to be a Commander as long as the intended stack name does not already exist on the broadcast domain. (When you configure a Commander, you automatically create a corresponding stack.)

Before you begin configuring stacking parameters:

1. Configure IP addressing on the switch intended for stack commander and, if not already configured, on the primary VLAN. (For more on configuring IP addressing, refer to the *Management and Configuration Guide* for your switch.)

Note

The primary VLAN must have an IP address in order for stacking to operate properly. For more on the primary VLAN, see “The Primary VLAN” on page 2-45.

2. Configure a Manager password on the switch intended for commander. (The Commander’s Manager password controls access to stack Members.) For more on passwords, see the local manager and operator password information in the *Access Security Guide* for your switch.

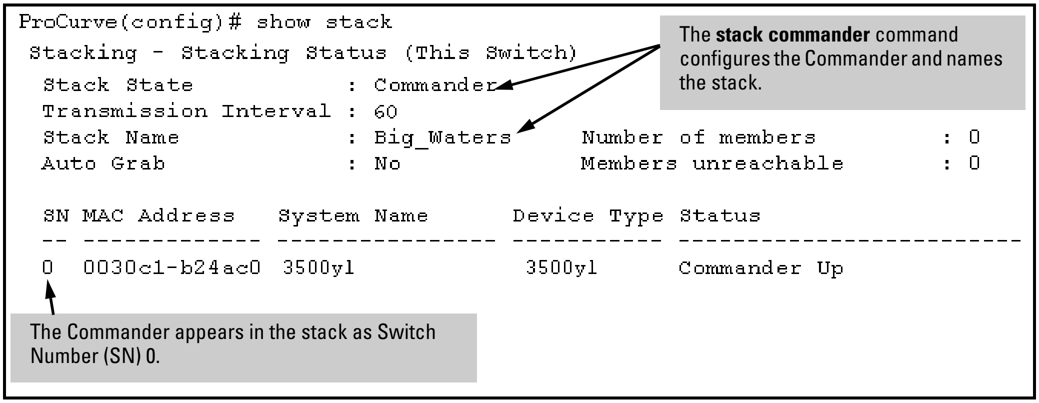
Configure the Stack Commander. Assigning a stack name to a switch makes it a Commander and automatically creates a stack.

Syntax: stack commander < *name-str* >

This example creates a Commander switch with a stack name of **Big_Waters**. (Note that if stacking was previously disabled on the switch, this command also enables stacking.)

```
ProCurve(config)# stack commander Big_Waters
```

As the following **show stack** display shows, the Commander switch is now ready to add members to the stack.



```
ProCurve(config)# show stack
 Stacking - Stacking Status (This Switch)
  Stack State           : Commander
  Transmission Interval : 60
  Stack Name            : Big_Waters     Number of members          : 0
  Auto Grab             : No             Members unreachable        : 0

  SN MAC Address   System Name      Device Type Status
  -- ------------- ---------------- ----------- -------------------------
  0  0030c1-b24ac0  3500yl            3500yl      Commander Up
```

The **stack commander** command configures the Commander and names the stack.

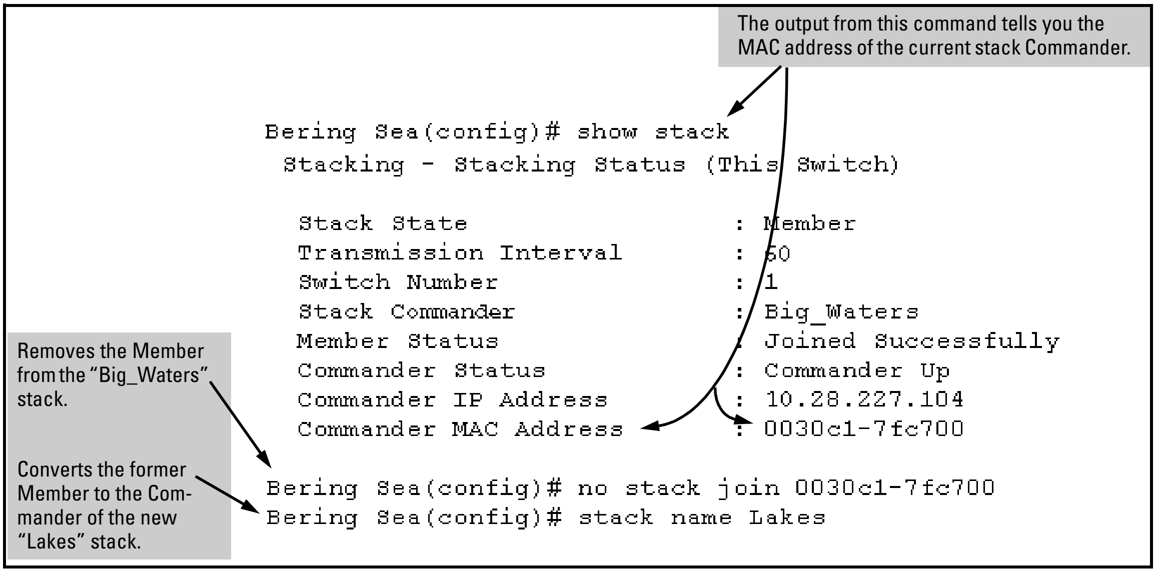
The Commander appears in the stack as Switch Number (SN) 0.

Figure 8-26. Example of the Commander's Show Stack Screen with Only the Commander Discovered

Using a Member's CLI to Convert the Member to the Commander of a New Stack. This procedure requires that you first remove the Member from its current stack, then create the new stack. If you do not know the MAC address for the Commander of the current stack, use **show stack** to list it.

Syntax: no stack
stack commander < *stack name* >

Suppose, for example, that a ProCurve switch named "Bering Sea" is a Member of a stack named "Big_Waters". To use the switch's CLI to convert it from a stack Member to the Commander of a new stack named "Lakes", you would use the following commands:



The output from this command tells you the MAC address of the current stack Commander.

```
Bering Sea(config)# show stack
 Stacking - Stacking Status (This Switch)

  Stack State                   : Member
  Transmission Interval         : 60
  Switch Number                 : 1
  Stack Commander               : Big_Waters
  Member Status                 : Joined Successfully
  Commander Status              : Commander Up
  Commander IP Address          : 10.28.227.104
  Commander MAC Address         : 0030c1-7fc700

Bering Sea(config)# no stack join 0030c1-7fc700
Bering Sea(config)# stack name Lakes
```

Removes the Member from the "Big_Waters" stack.

Converts the former Member to the Commander of the new "Lakes" stack.

Figure 8-27. Example of Using a Member's CLI To Convert the Member to the Commander of a New Stack

Adding to a Stack or Moving Switches Between Stacks

You can add switches to a stack by adding discovered Candidates or by moving switches from other stacks that may exist in the same subnet. (You cannot add a Candidate that the Commander has not discovered.)

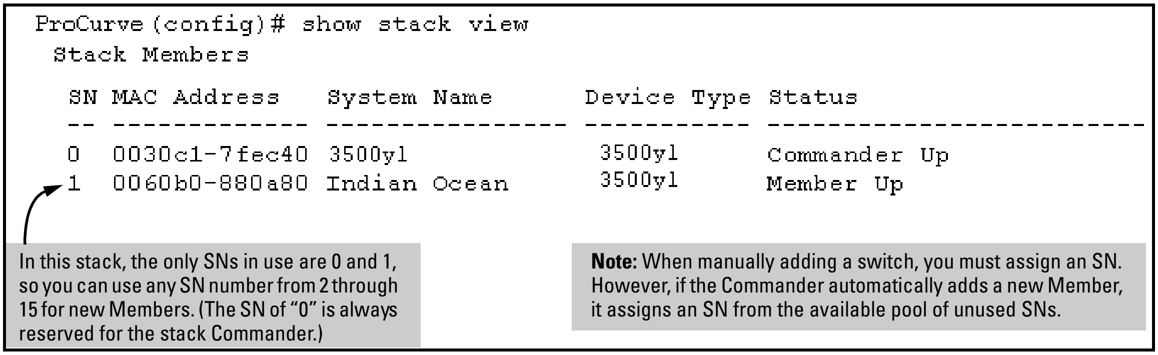
In its default configuration, the Commander's **Auto-Grab** parameter is set to **No** to give you manual control over which switches join the stack and when they join. This prevents the Commander from automatically trying to add every Candidate it finds that has **Auto Join** set to **Yes** (the default for the Candidate).

(If you want any eligible Candidate to automatically join the stack when the Commander discovers it, configure **Auto Grab** in the Commander to **Yes**. When you do so, *any* Candidate discovered with **Auto Join** set to **Yes** (the default) and no Manager password will join the stack, up to the limit of 15 Members.)

Using the Commander's CLI To Manually Add a Candidate to the Stack. To manually add a candidate, you will use:

- A switch number (**SN**) to assign to the new member. Member SNs range from 1 to 15. To see which SNs are already assigned to Members, use **show stack view**. You can use any SN not included in the listing. (SNs are viewable only on a Commander switch.)
- The MAC address of the discovered Candidate you are adding to the stack. To see this data, use the **show stack candidates** listing .

For example:



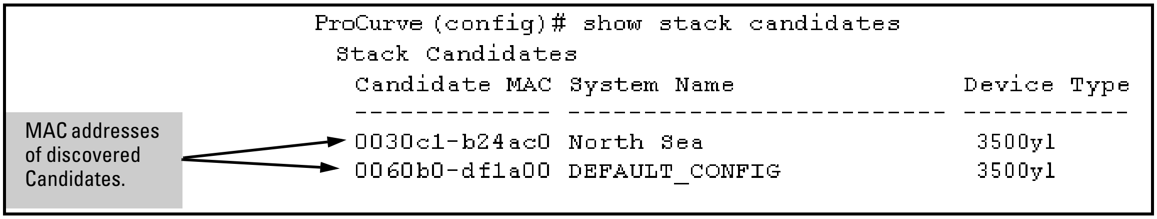
```
ProCurve(config)# show stack view
 Stack Members

  SN MAC Address   System Name      Device Type Status
  -- ------------- ---------------- ----------- -------------------------
  0  0030c1-7fec40 3500yl            3500yl     Commander Up
  1  0060b0-880a80 Indian Ocean      3500yl     Member Up
```

In this stack, the only SNs in use are 0 and 1, so you can use any SN number from 2 through 15 for new Members. (The SN of "0" is always reserved for the stack Commander.)

Note: When manually adding a switch, you must assign an SN. However, if the Commander automatically adds a new Member, it assigns an SN from the available pool of unused SNs.

Figure 8-28. Example of How To Determine Available Switch Numbers (SNs)

To display all discovered Candidates with their MAC addresses, execute **show stack candidates** from the Commander's CLI. For example, to list the discovered candidates for the above Commander:



```
ProCurve(config)# show stack candidates
 Stack Candidates
  Candidate MAC System Name               Device Type
  ------------- ------------------------- -----------
  0030c1-b24ac0 North Sea                  3500yl
  0060b0-df1a00 DEFAULT_CONFIG             3500yl
```

MAC addresses of discovered Candidates.

Figure 8-29. Example of How To Determine MAC Addresses of Discovered Candidates

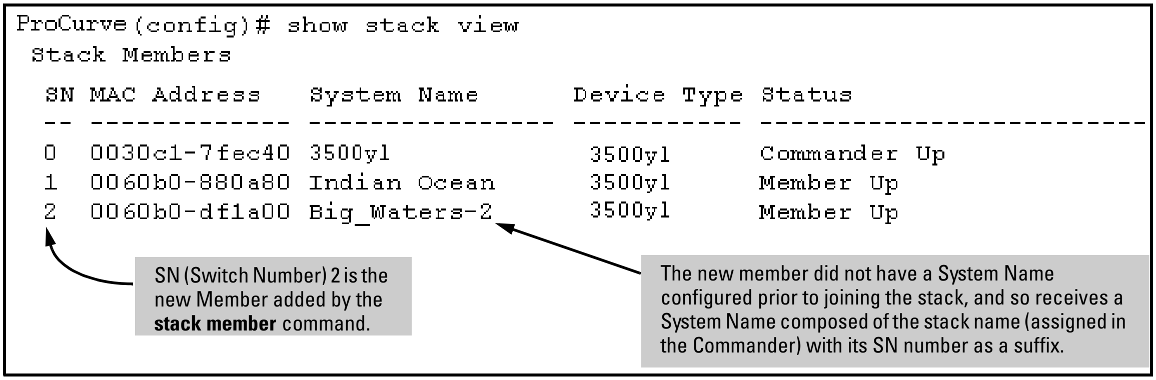
Knowing the available switch numbers (**SN**s) and Candidate MAC addresses, you can proceed to manually assign a Candidate to be a Member of the stack:

Syntax: stack member < *switch-number* > mac-address < *mac-addr* >
[password < *password-str* >]

For example, if the switch in the above listing did not have a Manager password and you wanted to make it a stack Member with an **SN** of **2**, you would execute the following command:

```
ProCurve(config)# stack member 2 mac-address 0060b0-
dfla00
```

The **show stack view** command then lists the Member added by the above command:



```
ProCurve(config)# show stack view
 Stack Members

  SN MAC Address    System Name      Device Type Status
  -- ------------- ---------------- ----------- -------------------------
  0  0030c1-7fec40 3500yl            3500yl      Commander Up
  1  0060b0-880a80 Indian Ocean      3500yl      Member Up
  2  0060b0-df1a00 Big_Waters-2      3500yl      Member Up
```

SN (Switch Number) 2 is the new Member added by the **stack member** command.

The new member did not have a System Name configured prior to joining the stack, and so receives a System Name composed of the stack name (assigned in the Commander) with its SN number as a suffix.

Figure 8-30. Example Showing the Stack After Adding a New Member

Using Auto Join on a Candidate. In the default configuration, a Candidate's Auto Join parameter is set to "Yes", meaning that it will automatically join a stack if the stack's Commander detects the Candidate and the Commander's Auto Grab parameter is set to "Yes". You can disable Auto Join on a Candidate if you want to prevent automatic joining in this case. There is also the instance where a Candidate's Auto Join is disabled, for example, when a Commander leaves a stack and its members automatically return to Candidate status, or if you manually remove a Member from a stack. In this case, you may want to reset Auto Join to "Yes".

Status: **[no] stack auto-join**

```
ProCurve(config)# no stack auto-join
```
Disables Auto Join on a Candidate.

```
ProCurve(config)# stack auto-join
```
Enables Auto Join on a Candidate.

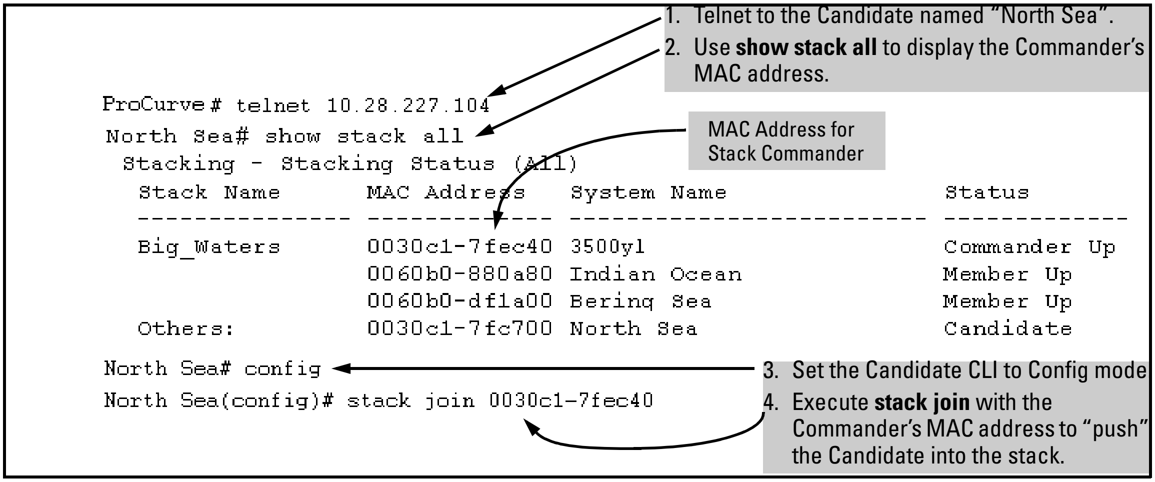
Using a Candidate CLI To Manually "Push" the Candidate Into a Stack . Use this method if any of the following apply:

- The Candidate's **Auto Join** is set to **Yes** (and you do not want to enable **Auto Grab** on the Commander) or the Candidate's **Auto Join** is set to **No**.
- Either you know the MAC address of the Commander for the stack into which you want to insert the Candidate, or the Candidate has a valid IP address and is operating in your network.

Syntax: stack join < *mac-addr* >

where: **<** ***mac-addr*** **>** is the MAC address of the Commander in the destination stack.

Use Telnet (if the Candidate has an IP address valid for your network) or a direct serial port connection to access the CLI for the Candidate switch. For example, suppose that a Candidate named "North Sea" with **Auto Join** off and a valid IP address of 10.28.227.104 is running on a network. You could Telnet to the Candidate, use **show stack all** to determine the Commander's MAC address, and then "push" the Candidate into the desired stack.



```
ProCurve# telnet 10.28.227.104
North Sea# show stack all
 Stacking - Stacking Status (All)
  Stack Name      MAC Address   System Name              Status
  --------------- ------------- ------------------------ -------------
  Big_Waters      0030c1-7fec40 3500yl                   Commander Up
                  0060b0-880a80 Indian Ocean             Member Up
                  0060b0-df1a00 Bering Sea               Member Up
  Others:         0030c1-7fc700 North Sea                Candidate

North Sea# config
North Sea(config)# stack join 0030c1-7fec40
```

1. Telnet to the Candidate named "North Sea".
2. Use **show stack all** to display the Commander's MAC address.

MAC Address for Stack Commander

3. Set the Candidate CLI to Config mode
4. Execute **stack join** with the Commander's MAC address to "push" the Candidate into the stack.

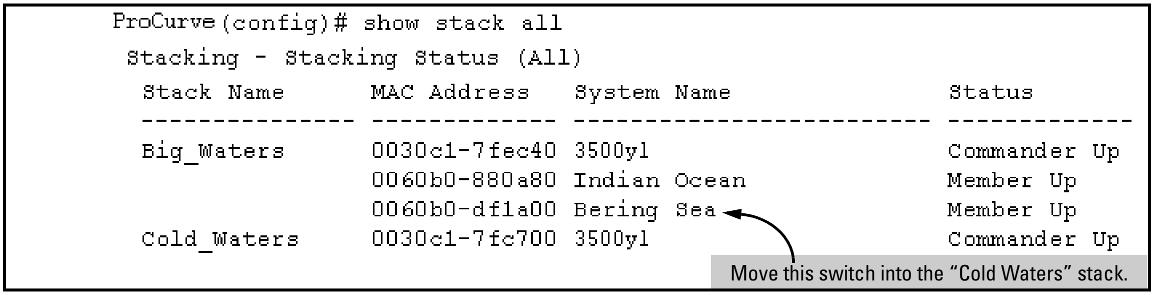
Figure 8-31. Example of "Pushing" a Candidate Into a Stack

To verify that the Candidate successfully joined the stack, execute **show stack all** again to view the stacking status.

Using the Destination Commander CLI To "Pull" a Member from Another Stack. This method uses the Commander in the destination stack to "pull" the Member from the source stack.

Syntax: stack member < *switch-number* >
mac-address < *mac-addr* >
[password < *password-str* >]

In the destination Commander, use **show stack all** to find the MAC address of the Member you want to pull into the destination stack. For example, suppose you created a new Commander with a stack name of "Cold_Waters" and you wanted to move a switch named "Bering Sea" into the new stack:



```
ProCurve(config)# show stack all
 Stacking - Stacking Status (All)
  Stack Name      MAC Address   System Name               Status
  --------------- ------------- ------------------------- -------------
  Big_Waters      0030c1-7fec40 3500yl                    Commander Up
                  0060b0-880a80 Indian Ocean              Member Up
                  0060b0-df1a00 Bering Sea                Member Up
  Cold_Waters     0030c1-7fc700 3500yl                    Commander Up
```

Move this switch into the "Cold Waters" stack.

Figure 8-32. Example of Stack Listing with Two Stacks in the Subnet

You would then execute the following command to pull the desired switch into the new stack:

```
ProCurve(config)# stack member 1 mac-address 0060b0-
df1a00
```

Where **1** is an unused switch number (**SN**).

Since a password is not set on the Candidate, a password is not needed in this example.

You could then use **show stack all** again to verify that the move took place.

Using a Member CLI To "Push" the Member into Another Stack. You can use the Member's CLI to "push" a stack Member into a destination stack if you know the MAC address of the destination Commander.

Syntax: stack join <*mac-addr*>

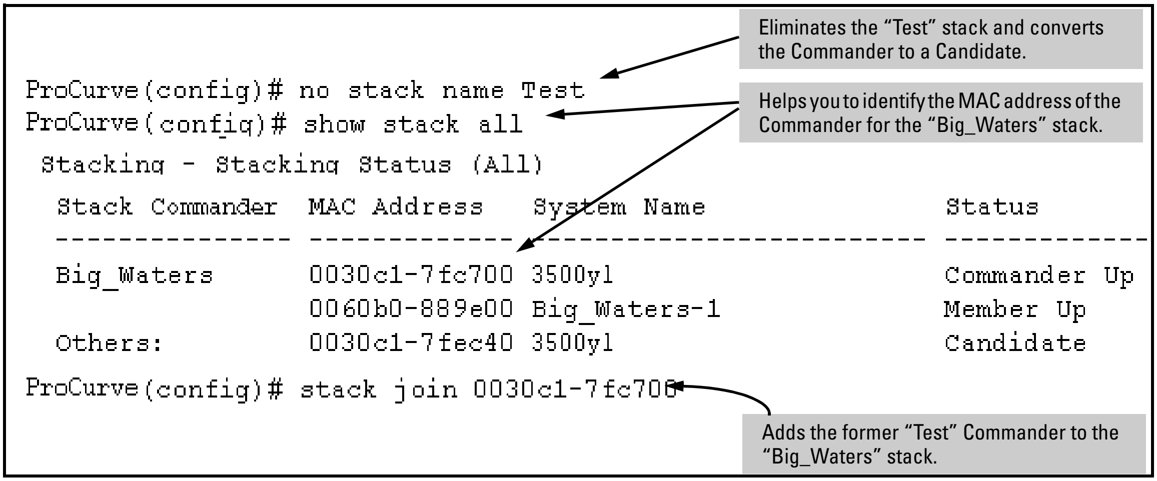
where: ***< mac-addr >*** is the MAC address of the Commander for the destination stack.

Converting a Commander to a Member of Another Stack. Removing the Commander from a stack eliminates the stack and returns its Members to the Candidate pool with **Auto Join** disabled.

Syntax: no stack name < *stack name*>
stack join < *mac-address* >

If you don't know the MAC address of the destination Commander, you can use **show stack all** to identify it.

For example, suppose you have a switch operating as the Commander for a temporary stack named "Test". When it is time to eliminate the temporary "Test" stack and convert the switch into a member of an existing stack named "Big_Waters", you would execute the following commands in the switch's CLI:



```
ProCurve(config)# no stack name Test
ProCurve(config)# show stack all
 Stacking - Stacking Status (All)

  Stack Commander  MAC Address   System Name               Status
  ---------------  ------------- ------------------------- ------------
  Big_Waters       0030c1-7fc700 3500yl                    Commander Up
                   0060b0-889e00 Big_Waters-1              Member Up
  Others:          0030c1-7fec40 3500yl                    Candidate

ProCurve(config)# stack join 0030c1-7fc700
```

Eliminates the "Test" stack and converts the Commander to a Candidate.

Helps you to identify the MAC address of the Commander for the "Big_Waters" stack.

Adds the former "Test" Commander to the "Big_Waters" stack.

Figure 8-33. Example of Command Sequence for Converting a Commander to a Member

Using the CLI To Remove a Member from a Stack

You can remove a Member from a stack using the CLI of either the Commander or the Member.

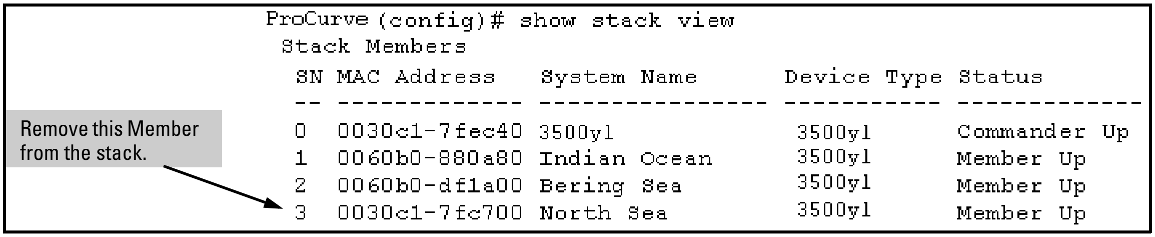
Note

When you remove a Member from a stack, the Member's **Auto Join** parameter is set to **No**.

Using the Commander CLI To Remove a Stack Member. This option requires the switch number (SN) and the MAC address of the switch to remove. (Because the Commander propagates its Manager password to all stack members, knowing the Manager password is necessary only for gaining access to the Commander.)

Syntax: [no] stack member <*switch-num*> mac-address <*mac-addr*>

Use **show stack view** to list the stack Members. For example, suppose that you wanted to use the Commander to remove the "North Sea" Member from the following stack:



```
ProCurve(config)# show stack view
 Stack Members

  SN MAC Address    System Name      Device Type Status
  -- ------------- ---------------- ----------- -------------
  0  0030c1-7fec40 3500yl            3500yl      Commander Up
  1  0060b0-880a80 Indian Ocean      3500yl      Member Up
  2  0060b0-df1a00 Bering Sea        3500yl      Member Up
  3  0030c1-7fc700 North Sea         3500yl      Member Up
```

Remove this Member from the stack.

Figure 8-34. Example of a Commander and Three Switches in a Stack

You would then execute this command to remove the "North Sea" switch from the stack:

```
ProCurve(config)# no stack member 3 mac-address 0030c1-
7fc700
```

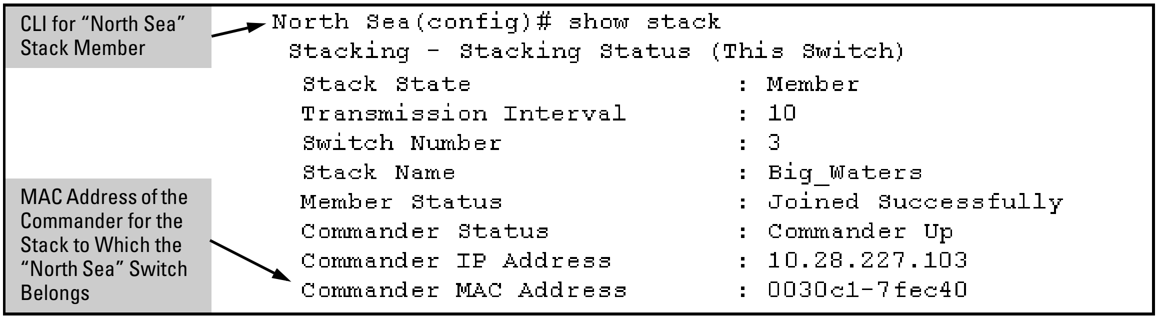
where:

- **3** is the "North Sea" Member's switch number (**SN**)
- **0030c1-7fc700** is the "North Sea" Member's MAC address

Using the Member's CLI To Remove the Member from a Stack.

Syntax: no stack join <*mac-addr*>

To use this method, you need the Commander's MAC address, which is available using the show stack command in the Member's CLI. For example:



```
North Sea(config)# show stack
 Stacking - Stacking Status (This Switch)

  Stack State                   : Member
  Transmission Interval         : 10
  Switch Number                 : 3
  Stack Name                    : Big_Waters
  Member Status                 : Joined Successfully
  Commander Status              : Commander Up
  Commander IP Address          : 10.28.227.103
  Commander MAC Address         : 0030c1-7fec40
```

CLI for "North Sea" Stack Member

MAC Address of the Commander for the Stack to Which the "North Sea" Switch Belongs

Figure 8-35. Example of How To Identify the Commander's MAC Address from a Member Switch

You would then execute this command in the "North Sea" switch's CLI to remove the switch from the stack:

```
North Sea(config)# no stack join 0030c1-7fec40
```

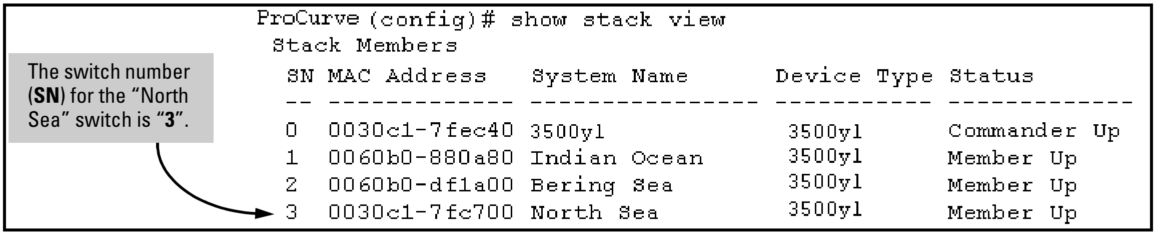
Using the CLI To Access Member Switches for Configuration Changes and Traffic Monitoring

After a Candidate becomes a Member, you can use the telnet command from the Commander to access the Member's CLI or console interface for the same configuration and monitoring that you would do through a Telnet or direct-connect access from a terminal.

Syntax: telnet <*switch-number*>

where: unsigned integer is the switch number (**SN**) assigned by the Commander to each member (range: **1 - 15**).

To find the switch number for the Member you want to access, execute the **show stack view** command in the Commander's CLI. For example, suppose that you wanted to configure a port trunk on the switch named "North Sea" in the stack named "Big_Waters". Do do so you would go to the CLI for the "Big_Waters" Commander and execute show stack view to find the switch number for the "North Sea" switch:



```
ProCurve (config)# show stack view
 Stack Members
  SN MAC Address   System Name      Device Type Status
  -- ------------- ---------------- ----------- -------------
  0  0030c1-7fec40 3500yl            3500yl      Commander Up
  1  0060b0-880a80 Indian Ocean      3500yl      Member Up
  2  0060b0-df1a00 Bering Sea        3500yl      Member Up
  3  0030c1-7fc700 North Sea         3500yl      Member Up
```

The switch number (**SN**) for the "North Sea" switch is "**3**".

Figure 8-36. Example of a Stack Showing Switch Number (SN) Assignments

To access the "North Sea" console, you would then execute the following **telnet** command:

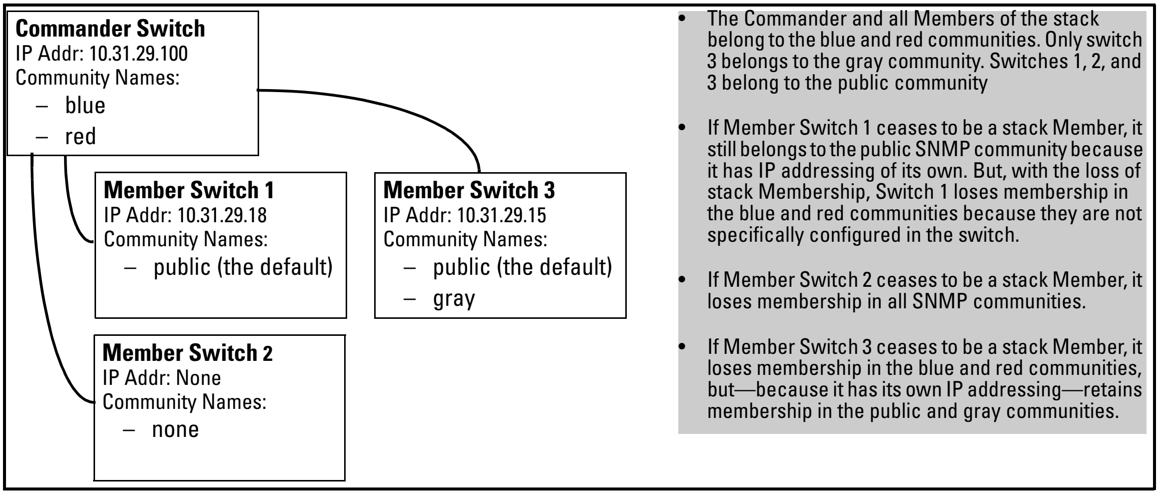
```
ProCurve(config)# telnet 3
```

You would then see the CLI prompt for the "North Sea" switch, allowing you to configure or monitor the switch as if you were directly connected to the console.

SNMP Community Operation in a Stack

Community Membership

In the default stacking configuration, when a Candidate joins a stack, it automatically becomes a Member of any SNMP community to which the Commander belongs, even though any community names configured in the Commander are not propagated to the Member's SNMP Communities listing. However, if a Member has its own (optional) IP addressing, it can belong to SNMP communities to which other switches in the stack, including the Commander, do not belong. For example:



- The Commander and all Members of the stack belong to the blue and red communities. Only switch 3 belongs to the gray community. Switches 1, 2, and 3 belong to the public community
- If Member Switch 1 ceases to be a stack Member, it still belongs to the public SNMP community because it has IP addressing of its own. But, with the loss of stack Membership, Switch 1 loses membership in the blue and red communities because they are not specifically configured in the switch.
- If Member Switch 2 ceases to be a stack Member, it loses membership in all SNMP communities.
- If Member Switch 3 ceases to be a stack Member, it loses membership in the blue and red communities, but—because it has its own IP addressing—retains membership in the public and gray communities.

Figure 8-37. Example of SNMP Community Operation with Stacking

SNMP Management Station Access to Members Via the Commander.

To use a management station for SNMP Get or Set access through the Commander's IP address to a Member, you must append ***@sw<switch number>*** to the community name. For example, in figure 8-37, you would use the following command in your management station to access Switch 1's MIB using the blue community:

```
snmpget <MIB variable> 10.31.29.100 blue@sw1
```

Note that because the gray community is only on switch 3, you could not use the Commander IP address for gray community access from the management station. Instead, you would access switch 3 directly using the switch's own IP address. For example:

```
snmpget <MIB variable> 10.31.29.15 gray
```

Note that in the above example (figure 8-37) you cannot use the public community through the Commander to access any of the Member switches. For example, you can use the public community to access the MIB in switches 1 and 3 by using their unique IP addresses. However, you must use the red or blue community to access the MIB for switch 2.

```
snmpget < MIB variable > 10.31.29.100 blue@sw2
```

Using the CLI To Disable or Re-Enable Stacking

In the default configuration, stacking is enabled on the switch. You can use the CLI to disable stacking on the switch at any time. Disabling stacking has the following effects:

- **Disabling a Commander:** Eliminates the stack, returns the stack Members to Candidates with **Auto Join** disabled, and changes the Commander to a stand-alone (nonstacking) switch. You must re-enable stacking on the switch before it can become a Candidate, Member, or Commander.
- **Disabling a Member:** Removes the Member from the stack and changes it to a stand-alone (nonstacking) switch. You must re-enable stacking on the switch before it can become a Candidate, Member, or Commander.
- **Disabling a Candidate:** Changes the Candidate to a stand-alone (nonstacking) switch.

Syntax: no stack (*Disables stacking on the switch.*)
stack (*Enables stacking on the switch.*)

Transmission Interval

All switches in the stack must be set to the same transmission interval to help ensure proper stacking operation. HP recommends that you leave this parameter set to the default 60 seconds.

Syntax: stack transmission-interval < *seconds* >

Stacking Operation with Multiple VLANs Configured

Stacking uses the primary VLAN in a switch. In the factory-default configuration, the DEFAULT_VLAN is the primary VLAN. However, you can designate any VLAN configured in the switch as the primary VLAN. (See “The Primary VLAN” on page 2-45.)

When using stacking in a multiple-VLAN environment, the following criteria applies:

- Stacking uses only the primary VLAN on each switch in a stack.
- The primary VLAN can be tagged or untagged as needed in the stacking path from switch to switch.
- The same VLAN ID (VID) must be assigned to the primary VLAN in each stacked switch.

Status Messages

Stacking screens and listings display these status messages:

Message	Condition	Action or Remedy
Candidate Auto-join	Indicates a switch configured with Stack State set to **Candidate, Auto Join** set to **Yes** (the default), and no Manager password.	None required
Candidate	Candidate cannot automatically join the stack because one or both of the following conditions apply: • Candidate has **Auto Join** set to **No**. • Candidate has a Manager password.	Manually add the candidate to the stack.
Commander Down	Member has lost connectivity to its Commander.	Check connectivity between the Commander and the Member.
Commander Up	The Member has stacking connectivity with the Commander.	None required.
Mismatch	This may be a temporary condition while a Candidate is trying to join a stack. If the Candidate does not join, then stack configuration is inconsistent.	Initially, wait for an update. If condition persists, reconfigure the Commander or the Member.
Member Down	A Member has become detached from the stack. A possible cause is an interruption to the link between the Member and the Commander.	Check the connectivity between the Commander and the Member.
Member Up	The Commander has stacking connectivity to the Member.	None required.
Rejected	The Candidate has failed to be added to the stack.	The candidate may have a password. In this case, manually add the candidate. Otherwise, the stack may already be full. A stack can hold up to 15 Members (plus the Commander).

—This page intentionally unused—

Index

ProCurve Networking
HP Innovation

Technical information in this document
is subject to change without notice.

July 2006

Manual Part Number
5991-3827