SROS Command Line Interface
Reference Guide
Software Version J.05.02
Disclaimer
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.
# Table of Contents

Basic Mode Command Set ............................................................... 12
Enable Mode Command Set ............................................................. 20
Global Configuration Mode Command Set .......................................... 285
Line (Console) Interface Config Command Set ................................... 490
Line (Telnet) Interface Config Command Set .................................... 503
Line (SSH) Interface Config Command Set ....................................... 512
ADSL Interface Config Command Set ................................................ 519
BRI Interface Configuration Command set ....................................... 524
DSX-1 Interface Configuration Command Set .................................... 540
E1 Interface Configuration Command Set ......................................... 550
Ethernet Interface Configuration Command Set ................................. 565
G.703 Interface Configuration Command set .................................... 626
Serial Interface Configuration Command Set ..................................... 633
Modem Interface Configuration Command Set ................................... 642
T1 Interface Configuration Command Set ......................................... 647
ATM Interface Config Command Set ................................................ 662
ATM Sub-Interface Config Command Set ......................................... 665
Demand Interface Configuration Command Set ................................. 734
Frame Relay Interface Config Command Set .................................... 793
Frame Relay Sub-Interface Config Command Set ............................... 814
HDLC Command Set ........................................................................ 881
Loopback Interface Configuration Command Set ............................... 942
PPP Interface Configuration Command Set ....................................... 976
Tunnel Configuration Command Set ................................................ 1053
ISDN Group Config Command Set .................................................. 1112
CA Profile Configuration Command Set ......................................... 1120
Certificate Configuration Command Set ......................................... 1131
Crypto Map IKE Command Set ....................................................... 1135
Crypto Map Manual Command Set .................................................. 1146
IKE Client Command Set ............................................................... 1157
IKE Policy Attributes Command Set .............................................. 1161
IKE Policy Command Set ............................................................... 1167
AS Path List Command Set ............................................................. 1178
Route Map Command Set ............................................................... 1181
BGP Configuration Command Set .................................................... 1205
BGP Neighbor Configuration Command Set ...................................... 1218
Community List Command Set ....................................................... 1232
Router (OSPF) Configuration Command Set .................................... 1235
Router (PIM Sparse) Configuration Command Set ............................. 1249
Router (RIP) Configuration Command Set ....................................... 1253
Quality of Service (QoS) Map Commands .................................................. 1265  
DHCP Pool Command Set ................................................................. 1279  
Radius Group Command Set ........................................................... 1298  
TACACS+ Group Configuration Command Set ................................. 1300  
Common Commands ........................................................................ 1302  
Index ............................................................................................... 1316
REFERENCE GUIDE INTRODUCTION

This manual provides information about the commands that are available with all of the ProCurve Secure routers.

If you are new to the SROS Command Line Interface (CLI), take a few moments to review the information provided in the section which follows (CLI Introduction).

If you are already familiar with the CLI and you need information on a specific command or group of commands, proceed to Command Descriptions on page 10 of this guide.

CLI INTRODUCTION

This portion of the Command Reference Guide is designed to introduce you to the basic concepts and strategies associated with using the SROS Command Line Interface (CLI).

Accessing the CLI from your PC

All products using the SROS are initially accessed by connecting a VT100 terminal (or terminal emulator) to the CONSOLE port located on the front of the unit using a standard DB-9 (male) to DB-9 (female) serial cable. Configure the VT100 terminal or terminal emulation software to the following settings:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit
- No flow control

Note: For more details on connecting to your unit, refer to the Quick Configuration Guides and Quick Start Guides located on the Secure Router OS Documentation CD provided with your unit.

Understanding Command Security Levels

The SROS has two command security levels — Basic and Enable. Both levels support a specific set of commands. For example, all interface configuration commands are accessible only through the Enable security level. The following table contains a brief description of each level.

<table>
<thead>
<tr>
<th>Level</th>
<th>Access by...</th>
<th>Prompt</th>
<th>With this level you can...</th>
</tr>
</thead>
</table>
| Basic   | beginning an SROS session. | ProCurve>    | • display system information
|         |              |              | • perform traceroute and ping functions
|         |              |              | • open a Telnet session |
Understanding Configuration Modes

The SROS has four configuration modes to organize the configuration commands – Global, Line, Router, and Interface. Each configuration mode supports a set of commands specific to the configurable parameters for the mode. For example, all Frame Relay configuration commands are accessible only through the Interface Configuration mode (for the virtual Frame Relay interface). The following table contains a brief description of each level.

<table>
<thead>
<tr>
<th>Level</th>
<th>Access by...</th>
<th>Prompt</th>
<th>With this level you can...</th>
</tr>
</thead>
</table>
| Enable  | entering `enable` while in the Basic command security level as follows: `>enable` | ProCurve# | • manage the startup and running configurations  
  • use the debug commands  
  • enter any of the configuration modes |

**Note**
To prevent unauthorized users from accessing the configuration functions of your product, immediately install an Enable-level password. Refer to the Quick Configuration Guides and Quick Start Guides located on the Secure Router OS Documentation CD provided with your unit for more information on configuring a password.

**Understanding Configuration Modes**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Access by...</th>
<th>Sample Prompt</th>
<th>With this mode you can...</th>
</tr>
</thead>
</table>
| Global  | entering `config` while at the Enable command security level prompt. For example: `>enable` `#config term` | ProCurve(config)# | • set the system’s Enable-level password(s)  
  • configure the system global IP parameters  
  • configure the SNMP parameters  
  • enter any of the other configuration modes |
| Line    | specifying a line (console or Telnet) while at the Global Configuration mode prompt. For example: `>enable` `#config term` `line console 0` | ProCurve(config-con0)# | • configure the console terminal settings (datarate, login password, etc.)  
  • create Telnet logins and specify their parameters (login password, etc.) |
## Using CLI Shortcuts

The SROS CLI provides several shortcuts which help you configure your product more easily. See the following table for descriptions.

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up arrow key</td>
<td>To re-display a previously entered command, use the up arrow key. Continuing to press the up arrow key cycles through all commands entered starting with the most recent command.</td>
</tr>
<tr>
<td>Tab key</td>
<td>Pressing the &lt;Tab&gt; key after entering a partial (but unique) command will complete the command, display it on the command prompt line, and wait for further input.</td>
</tr>
</tbody>
</table>
| ? | The CLI contains help to guide you through the configuration process. Using the question mark, do any of the following:  
  - Display a list of all subcommands in the current mode. For example:  
    ProCurve(config-t1 1/1)#coding ?  
    ami - Alternate Mark Inversion  
    b8zs - Bipolar Eight Zero Substitution  
  - Display a list of available commands beginning with certain letter(s). For example:  
    ProCurve(config)#ip d?  
    default-gateway dhcp-server domain-lookup domain-name domain-proxy  
  - Obtain syntax help for a specific command by entering the command, a space, and then a question mark (?). The CLI displays the range of values and a brief description of the next parameter expected for that particular command. For example:  
    ProCurve(config-eth 0/1)#mtu ?  
    <64-1500> - MTU (bytes) |
Performing Common CLI Functions

The following table contains descriptions of common CLI commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>do</td>
<td>The <code>do</code> command provides a way to execute commands in other command sets without taking the time to exit the current and enter the desired one. The following example shows the <code>do</code> command used to view the Frame Relay interface configuration while currently in the T1 interface command set:</td>
</tr>
</tbody>
</table>
|                          | `ProCurve(config)#interface t1 1/1`  
|                          | `ProCurve(config-t1 1/1)#do show interfaces fr 7`                         |
| no                       | To undo an issued command or to disable a feature, enter `no` before the command. For example: |
|                          | `no shutdown t1 1/1`                                                       |
| copy running-config startup-config | When you are ready to save the changes made to the configuration, enter this command. This copies your changes to the unit’s nonvolatile random access memory (NVRAM). Once the save is complete, the changes are retained even if the unit is shut down or suffers a power outage. |
The following table lists and defines some of the more common error messages given in the CLI.

<table>
<thead>
<tr>
<th>Message</th>
<th>Helpful Hints</th>
</tr>
</thead>
<tbody>
<tr>
<td>%Ambiguous command</td>
<td>The command may not be valid in the current command mode, or you may not have entered enough correct characters for the command to be recognized. Try using the “?” command to determine your error. See Using CLI Shortcuts on page 7 for more information.</td>
</tr>
<tr>
<td>%Unrecognized Command</td>
<td></td>
</tr>
<tr>
<td>%Invalid or incomplete command</td>
<td>The command may not be valid in the current command mode, or you may not have entered all of the pertinent information required to make the command valid. Try using the “?” command to determine your error. See Using CLI Shortcuts on page 7 for more information.</td>
</tr>
<tr>
<td>%Invalid input detected at “^” marker</td>
<td>The error in command entry is located where the caret (^) mark appears. Enter a question mark at the prompt. The system will display a list of applicable commands or will give syntax information for the entry.</td>
</tr>
</tbody>
</table>
COMMAND DESCRIPTIONS

This portion of the guide provides a detailed listing of all available commands for the SROS CLI (organized by command set). Each command listing contains pertinent information including the default value, a description of all sub-command parameters, functional notes for using the command, and a brief technology review. To search for a particular command alphabetically, use the Index. To search for information on a group of commands within a particular command set, use the linked references given below:

Basic Mode Command Set on page 12
Enable Mode Command Set on page 20
Global Configuration Mode Command Set on page 285
Line (Console) Interface Config Command Set on page 490
Line (Telnet) Interface Config Command Set on page 503
Line (SSH) Interface Config Command Set on page 512
ADSL Interface Config Command Set on page 519
BRI Interface Configuration Command set on page 524
DSX-1 Interface Configuration Command Set on page 540
E1 Interface Configuration Command Set on page 550
Ethernet Interface Configuration Command Set on page 565
G.703 Interface Configuration Command set on page 626
Serial Interface Configuration Command Set on page 633
Modem Interface Configuration Command Set on page 642
T1 Interface Configuration Command Set on page 647
ATM Interface Config Command Set on page 662
ATM Sub-Interface Config Command Set on page 665
Frame Relay Interface Config Command Set on page 793
Frame Relay Sub-Interface Config Command Set on page 814
HDLC Command Set on page 881
Loopback Interface Configuration Command Set on page 942
PPP Interface Configuration Command Set on page 976
Tunnel Configuration Command Set on page 1053
ISDN Group Config Command Set on page 1112
CA Profile Configuration Command Set on page 1120
Certificate Configuration Command Set on page 1131
Crypto Map IKE Command Set on page 1135
Crypto Map Manual Command Set on page 1146
IKE Client Command Set on page 1157
IKE Policy Attributes Command Set on page 1161
IKE Policy Command Set on page 1167
AS Path List Command Set on page 1178
Route Map Command Set on page 1181
BGP Configuration Command Set on page 1205
BGP Neighbor Configuration Command Set on page 1218
Community List Command Set on page 1232
Router (RIP) Configuration Command Set on page 1253
Router (OSPF) Configuration Command Set on page 1235
Router (PIM Sparse) Configuration Command Set on page 1249
Quality of Service (QoS) Map Commands on page 1265
DHCP Pool Command Set on page 1279
Radius Group Command Set on page 1298
TACACS+ Group Configuration Command Set on page 1300
Common Commands on page 1302
BASIC MODE COMMAND SET

To activate the Basic mode, simply log in to the unit. After connecting the unit to a VT100 terminal (or terminal emulator) and activating a terminal session, the following prompt displays:

ProCurve>

The following command is common to multiple command sets and is covered in a centralized section of this guide. For more information, refer to the section listed below:

exit on page 1310
ping <address> on page 1311

All other commands for this command set are described in this section in alphabetical order.

enable on page 13
logout on page 14
show clock on page 15
show snmp on page 16
show version on page 17
telnet <address> on page 18
traceroute <address> on page 19
**enable**

Use the `enable` command (at the Basic Command mode prompt) to enter the Enable Command mode. Use the `disable` command to exit the Enable Command mode.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Functional Notes**

The Enable Command mode provides access to operating and configuration parameters and should be password protected to prevent unauthorized use. Use the `enable password` command (found in the Global Configuration mode) to specify an Enable Command mode password. If the password is set, access to the Enable Commands (and all other “privileged” commands) is only granted when the correct password is entered. Refer to `crypto ca authenticate <name>` on page 312 for more information.

**Usage Examples**

The following example enters the Enable Command mode and defines an Enable Command mode password:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#enable password password
```

At the next login, the following sequence must occur:

```
ProCurve>enable
Password: ******
ProCurve#
```
logout

Use the logout command to terminate the current session and return to the login screen.

Syntax Description
No subcommands.

Default Values
No defaults necessary for this command.

Usage Examples
The following example shows the logout command being executed in the Basic mode:

ProCurve> logout

Session now available

Press RETURN to get started.
**show clock**

Use the `show clock` command to display the system time and date entered using the `clock set` command. Refer to `clock set <time> <day> <month> <year>` on page 60 for more information.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example displays the current time and data from the system clock:

ProCurve> `show clock`

23:35:07 Tue Aug 20 2002
show snmp

Use the show snmp command to display the system Simple Network Management Protocol (SNMP) parameters and current status of SNMP communications.

Syntax Description
No subcommands.

Default Values
No default value necessary for this command.

Usage Examples
The following is an example output using the show snmp command for a system with SNMP disabled and the default chassis and contact parameters:

ProCurve>show snmp

Chassis: Chassis ID
Contact Name:  
Contact Phone:
Contact Email:
Contact Pager:
Management URL:
Management URL Label:
0 Rx SNMP packets
  0 Bad community names
  0 Bad community uses
  0 Bad versions
  0 Silent drops
  0 Proxy drops
  0 ASN parse errors
show version

Use the show version command to display the current SROS version information.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following is a sample show version output:

ProCurve>show version

ProCurve Secure Router 7203dl
SROS Version: J03.01
  Checksum: 4F8DCF96, built on: Tue Dec 21 08:32:18 2004
Boot ROM version J03.01
  Checksum: B133, built on: Tue Dec 21 08:32:25 2004
Copyright (c) 2004-2005, Hewlett-Packard, Co.
Platform: ProCurve Secure Router 7203dl
Serial number US449TS058
Flash: 33554432 bytes  DRAM: 268435455 bytes

System uptime is 0 days, 0 hours, 22 minutes, 42 seconds

Current system image file: "CFLASH:/SROS.BIZ"
Current configuration-file: CFLASH:/startup-config"
Configured system image path:
Primary: "CFLASH:/SROS.BIZ"
Backup: "NONVOL:/SROS.BIZ"
Configured configuration-file path:
Primary: "CFLASH:/startup-config"
Backup: "NONVOL:/startup-config"

ProCurve>
telnet <address>

Use the `telnet` command to open a Telnet session (through the SROS) to another system on the network.

**Syntax Description**

<address> Specifies the IP address of the remote system.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example opens a Telnet session with a remote system (10.200.4.15):

ProCurve>telnet 10.200.4.15

User Access Login

Password:
traceroute <address>

Use the traceroute command to display the IP routes a packet takes to reach the specified destination.

Syntax Description

<address> Specifies the IP address of the remote system to trace the routes to.

Default Values

No default value necessary for this command.

Usage Examples

The following example performs a traceroute on the IP address 192.168.0.1:

ProCurve> traceroute 192.168.0.1

Type CTRL+C to abort.
Tracing route to 192.168.0.1 over a maximum of 30 hops

1  22ms 20ms 20ms  192.168.0.65
2  23ms 20ms 20ms  192.168.0.1

ProCurve>
To activate the Enable mode, enter the **enable** command at the Basic mode prompt. (If an enable password has been configured, a password prompt will display.) For example:

ProCurve>**enable**
Password: XXXXXXX
ProCurve#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the section listed below:

- **exit** on page 1310
- **ping <address>** on page 1311
- **show running-config** on page 1313

All other commands for this command set are described in this section in alphabetical order.

- **autosynch** on page 22
- **clear commands** begin on page 24
- **clock [auto-correct-dst \| no-auto-correct-dst]** on page 59
- **clock set <time> <day> <month> <year>** on page 60
- **clock timezone <text>** on page 61
- **configure** on page 63
- **copy commands** begin on page 64
- **debug commands** begin on page 81
- **dir \[*\]** on page 143
- **dir [cflash \| flash] \[*\]** on page 144
- **disable on page 145**
- **enable on page 146**
- **erase \[SROS.BIZ \| startup-config\]** on page 147
- **erase [cflash \| flash] <filename>** on page 148
- **erase file-system cflash** on page 149
- **events** on page 150
- **exception report generate** on page 151
- **logout** on page 153
- **reload \[cancel \| in <delay>\]** on page 154
- **show commands** begin on page 155
- **sip check-sync** on page 278
- **telnet <address>** on page 279
terminal length <lines> on page 280
traceroute <address> source <address> on page 281
undebug all on page 282
wall <message> on page 283
write [erase | memory | network | terminal] on page 284
autosynch

Use the autosynch command to force a synchronization of the SROS.BIZ and startup-config files located in system flash and compact flash memory.

Syntax Description
No subcommands.

Default Values
No default value necessary for this command.

Functional Notes
The AutoSynch™ feature configures the system to synchronize the startup-config and SROS.BIZ files located in the system flash memory and the compact flash card. When enabled, the system compares the two files in the two locations and replaces the files located in the system flash memory with the ones from the compact flash card (regardless of which set of files is more current). This allows the customer to maintain the version of the operating system, and the configuration for that operating system, at the desired level. To accomplish this, a synchronization check is performed on the system any time there is a change in startup-config or SROS.BIZ on the compact flash card.

The AutoSynch™ feature allows for quick installation and updates of routers by inserting a compact flash card containing the desired software (must be renamed from the desired operating system software, such as J03.01.biz to SROS.BIZ) and startup configuration file (must be named startup-config) into a router with AutoSynch™ enabled. The ProCurve Secure Routers automatically boot from the compact flash (and secondarily from internal flash). After booting, with AutoSynch™ enabled, the router will synchronize the files in system flash memory with the desired files from compact flash.

Caution Deleting the SROS.BIZ and startup-config files from the compact flash card (using the erase command) deletes the files from the system flash memory as well.

Status commands associated with the AutoSynch™ feature include show version flash SROS.BIZ and show autosynch-status.

The show version flash SROS.BIZ command opens the specified .biz file and returns the current SROS version information.

ProCurve>enable
ProCurve#show version flash SROS.BIZ

Version: J03.01.00
The **show autosynch-status** command displays the current AutoSynch™ configuration and the statistics for the **SROS.BIZ** and **startup-config** files (if AutoSynch™ is enabled).

```
ProCurve> enable
ProCurve# show autosynch-status

AutoSynch: Mode - Enabled

AutoSynch: SROS.BIZ synched
AutoSynch: startup-config synched
```

**Usage Examples**

The following example forces a synchronization of **startup-config** and **SROS.BIZ** (located in system flash and compact flash memory):

```
ProCurve> enable
ProCurve# autosynch

AutoSynch: SROS.BIZ synched
AutoSynch: startup-config synched

ProCurve#
```
clear access-list <listname>

Use the clear access-list command to clear all counters associated with all access lists (or a specified access list).

Syntax Description

| <listname> | Optional. Specifies the name (label) of an access list |

Default Values

No default value necessary for this command.

Usage Examples

The following example clears all counters for the access list labeled MatchAll:

ProCurve>enable
ProCurve#clear access-list MatchAll
clear arp-cache

Use the clear arp-cache command to remove all dynamic entries from the Address Resolution Protocol (ARP) cache table.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following example removes all dynamic entries from the ARP cache:

ProCurve>enable
ProCurve#clear arp-cache
clear arp-entry <address>

Use the `clear arp-entry` command to remove a single entry from the Address Resolution Protocol (ARP) cache.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Specifies the IP address of the entry to remove.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example removes the entry for 10.200.4.56 from the ARP cache:

```
ProCurve>enable
ProCurve#clear arp-entry 10.200.4.56
```
clear bridge <group#>

Use the clear bridge command to clear all counters associated with bridging (or for a specified bridge-group).

Syntax Description

| <group#> | Optional. Specifies a single bridge group (1-255). |

Default Values

No default value necessary for this command.

Usage Examples

The following example clears all counters for bridge group 17:

ProCurve>enable
ProCurve#clear bridge 17
clear buffers max-used

Use the clear buffers max-used command to clear the maximum-used statistics for buffers displayed in the show memory heap command.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following example clears the maximum-used buffer statics:

ProCurve>enable
ProCurve#clear buffers max-used
clear counters [<interface> <interface id>]

Use the clear counters command to clear all interface counters (or the counters for a specified interface).

**Syntax Description**

- `<interface>`: Optional. Specifies a single interface. Enter clear counters ? or show interface ? for a complete list of interfaces.
- `<interface id>`: Optional. Specifies the ID of the specific interface to clear (e.g., 1 for port channel 1).

**Default Values**

No default values necessary for this command.

**Usage Examples**

The following example clears all counters associated with the Ethernet 0/1 interface:

ProCurve>enable
ProCurve#clear counters ethernet 0/1
clear crypto ike sa [policy <policy priority> | remote-id <remote-id>]

Use the clear crypto ike sa command to clear existing IKE security associations (SAs), including active ones. Use the policy and remote-id options to remove specific SAs.

Syntax Description

- **policy <policy priority>**
  - Removes all IKE SAs associated with the specified policy priority. The policy priority is assigned using crypto ike on page 319.

- **remote-id <remote-id>**
  - Removes all IKE SAs associated with the specified IKE remote ID. A delete payload is sent to the peers prior to deletion of the SA. This command is preferred to the clear crypto ike sa [policy <policy priority> | remote-id <remote-id>] command when multiple unique SAs have been created on the same IKE policy but the user wants to delete only the SA to a unique peer.

Default Values

No default value necessary for this command.

Usage Examples

The following example clears the entire database of IKE SAs (including the active associations):

ProCurve>enable
ProCurve#clear crypto ike sa

The following example clears IKE SAs associated with policy 101:

ProCurve>enable
ProCurve#clear crypto ike sa policy 101

The following example clears an IKE SA associated with remote-id procurve:

ProCurve>enable
ProCurve#clear crypto ike sa remote-id procurve
clear crypto ipsec sa

Use the clear crypto ipsec sa command to clear existing IPSec security associations (SAs), including active ones. Variations of this command include the following:

- clear crypto ipsec sa
- clear crypto ipsec sa entry <ip address> ah <SPI>
- clear crypto ipsec sa entry <ip address> esp <SPI>
- clear crypto ipsec sa map <map name>
- clear crypto ipsec sa peer <ip address>

**Syntax Description**

- **entry <ip address>**  
  Clears only the SAs related to a certain destination IP address.
- **ah <SPI>**  
  Clears only a portion of the SAs by specifying the AH (authentication header) protocol and a security parameter index (SPI). You can determine the correct SPI value using the *show crypto ipsec* command.
- **esp <SPI>**  
  Clears only a portion of the SAs by specifying the ESP (encapsulating security payload) protocol and a security parameter index (SPI). You can determine the correct SPI value using the *show crypto ipsec* command.
- **map <map name>**  
  Clears only the SAs associated with the crypto map name given.
- **peer <ip address>**  
  Clears only the SAs associated with the far-end peer IP address given.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example clears all IPSec SAs:

```
ProCurve> enable
ProCurve# clear crypto ipsec sa
```

The following example clears the IPSec SA used for ESP traffic with the SPI of 300 to IP address 172.27.45.57:

```
ProCurve> enable
ProCurve# clear crypto ipsec sa entry 172.27.45.57 esp 300
```
clear dump-core

The clear dump-core command clears diagnostic information appended to the output of the show version command. This information results from an unexpected unit reboot.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following example clears the entire database of IKE SAs (including the active associations):

ProCurve> enable
ProCurve# clear dump-core
clear event-history

Use the `clear event-history` command to clear all messages logged to the local event-history.

**Warning**

Messages cleared from the local event-history (using the `clear event-history` command) are no longer accessible.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example clears all local event-history messages:

```
ProCurve> enable
ProCurve# clear event-history
```
clear host [ * | <hostname> ]

Use the clear host command to clear a hostname (and associated address) from the DNS host-to-address table.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Clears all hosts from the host table.</td>
</tr>
<tr>
<td>&lt;hostname&gt;</td>
<td>Clears a specific host entry from the host-to-address table.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example clears all hostnames:

ProCurve>enable
ProCurve#clear host *

clear ip bgp [* | <as-number> | <ip address>] [in | out | soft]

Use the clear ip bgp command to clear BGP neighbors as specified.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Clears all BGP neighbors.</td>
</tr>
<tr>
<td>&lt;as-number&gt;</td>
<td>Clears all BGP neighbors with the specified AS number (Range: 1 to 65,535).</td>
</tr>
<tr>
<td>&lt;ip address&gt;</td>
<td>Clears the BGP neighbor with the specified IP address.</td>
</tr>
<tr>
<td>in</td>
<td>Causes a “soft” reset inbound with a neighbor, reprocessing routes advertised by that neighbor.</td>
</tr>
<tr>
<td>out</td>
<td>Causes a “soft” reset outbound with a neighbor, re-sending advertised routes to that neighbor.</td>
</tr>
<tr>
<td>soft</td>
<td>Causes a “soft” reset both inbound and outbound.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Functional Notes

The clear ip bgp command must be issued to re-initialize the BGP process between the peers matching the given arguments. Most neighbor changes, including changes to prefix-list filters, do not take effect until the clear command is issued. A hard reset clears the TCP connection with the specified peers, which results in clearing the table. This method of clearing is disruptive and causes peer routers to record a route flap for each route.

The out version of this command provides a soft reset out to occur by causing all routes to be re-sent to the specified peer(s). TCP connections are not torn down, so this method is less disruptive. Output filters/policies are re-applied before sending the update.

The in version of this command provides a soft reset in to occur by allowing the router to receive an updated table from a peer without tearing down the TCP connection. This method is less disruptive and does not count as a route flap. Currently, all of the peer’s routes are stored permanently, even if they are filtered by a prefix list. The command causes the peer’s routes to be reprocessed with any new parameters.

Usage Examples

The following example clears the information for all peers with an AS number of 101:

ProCurve>enable
ProCurve#clear ip bgp 101
clear ip cache

Use the `clear ip cache` command to delete cache table entries.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example removes all entries from the cache table:

```
ProCurve>enable
ProCurve#clear ip cache
```
clear ip dhcp-server binding [* | <ip address>]

Use the clear ip dhcp-server binding command to clear Dynamic Host Configuration Protocol (DHCP) server binding entries from the database.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Clears all automatic binding entries.</td>
</tr>
<tr>
<td>&lt;ip address&gt;</td>
<td>Clears a specific binding entry. Enter the source IP address (format is A.B.C.D).</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Functional Notes

A DHCP server binding represents an association between a MAC address and an IP address that was offered by the unit to a DHCP client (i.e., most often a PC). Clearing a binding allows the unit to offer that IP address again, should a request be made for one.

Usage Examples

The following example clears a DHCP server binding for the IP address 192.168.47.4:

```
ProCurve>enable
ProCurve#clear ip dhcp-server binding 192.168.47.4
```
clear ip igmp group [<group-address> | <interface>]

Use the clear ip igmp group command to clear entries from the Internet Group Management Protocol (IGMP) tables. If no address or interface is specified, all non-static IGMP groups are cleared with this command.

Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group-address&gt;</td>
<td>Optional. Specifies the multicast IP address of the multicast group.</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>Optional. Designates the clearing of parameters for a specific interface (in the format type slot/port). For example: eth 0/1.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example shows output for the show igmp groups command before and after a clear ip igmp group command is issued. This example clears the IGMP entry that was registered dynamically by a host. Interfaces that are statically joined are not cleared:

ProCurve>enable
ProCurve#show ip igmp groups

<table>
<thead>
<tr>
<th>Group Address</th>
<th>Interface</th>
<th>Uptime</th>
<th>Expires</th>
<th>Last Reporter</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.1.50</td>
<td>Loopback100</td>
<td>01:22:59</td>
<td>00:02:46</td>
<td>172.23.23.1</td>
</tr>
<tr>
<td>172.21.1.1</td>
<td>Ethernet0/1</td>
<td>00:00:14</td>
<td>00:02:45</td>
<td>1.1.1.2</td>
</tr>
<tr>
<td>172.31.1.1</td>
<td>Loopback100</td>
<td>01:22:59</td>
<td>00:02:46</td>
<td>172.23.23.1</td>
</tr>
</tbody>
</table>

ProCurve#clear ip igmp group
ProCurve#show ip igmp groups

This version of the command clears all dynamic groups that have the specified output interface (Ethernet 0/1):

ProCurve#clear ip igmp group ethernet 0/1

This version of the command clears the specified group on all interfaces where it is dynamically registered:

ProCurve#clear ip igmp group 172.21.1.1
clear ip ospf [process | redistribution]

Use the clear ip ospf command to reset open shortest path first (OSPF) information.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>process</td>
<td>Restarts the OSPF process.</td>
</tr>
<tr>
<td>redistribution</td>
<td>Refreshes routes redistributed over OSPF.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example resets the OSPF process:

ProCurve>enable
ProCurve#clear ip ospf process
clear ip policy-sessions

Use the `clear ip policy-sessions` command to clear policy class sessions. You may clear all the sessions or a specific session. Refer to the `show ip policy-sessions` for a current session listing. The following lists the complete syntax for the `clear ip policy-sessions` commands:

```
clear ip policy-sessions
```

**Syntax Description**

- `<classname>`: Alphanumeric descriptor for identifying the configured access policy (access policy descriptors are not case-sensitive).
- `ahp`: Specifies authentication header protocol (AHP).
- `esp`: Specifies encapsulating security payload protocol (ESP).
- `gre`: Specifies general routing encapsulation protocol (GRE).
- `icmp`: Specifies Internet control message protocol (ICMP) protocol.
- `tcp`: Specifies transmission control protocol (TCP).
- `udp`: Specifies universal datagram protocol (UDP).
- `<protocol>`: Specifies protocol (valid range: 0 to 255).
- `<source ip>`: Specifies the source IP address (format is A.B.C.D).
- `<source port>`: Specifies the source port (in hex format AHP, ESP, and GRE; decimal for all other protocols).
- `<dest ip>`: Specifies the destination IP address (format is A.B.C.D).
- `<dest port>`: Specifies the destination port (in hex format for AHP, ESP, and GRE; decimal for all other protocols).
- `[destination | source]`: For NAT sessions, this specifies whether to select a NAT source or NAT destination session.
- `<nat ip>`: For NAT sessions, this specifies the NAT IP address (format is A.B.C.D).
- `<nat port>`: For NAT sessions, this specifies the NAT port (in hex format for AHP, ESP, and GRE; decimal for all other protocols).

**Default Values**

No default value necessary for this command.

**Functional Notes**

The second half of this command, beginning with the source IP address may be copied and pasted from a row in the `show ip policy-sessions` table for easier use.
Usage Examples

The following example clears the Telnet association (TCP port 23) for policy class pclass1 with source IP address 172.22.71.50 and destination 172.22.71.130:

ProCurve>enable
ProCurve#clear ip policy-sessions pclass1 tcp 172.22.71.50 23 172.22.71.130 23
clear ip policy-stats <classname> entry <policy class #>

Use the clear ip policy-stats command to clear statistical counters for policy classes.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;classname&gt;</td>
<td>Optional. Specifies the policy class to clear. If no policy class is specified, statistics are cleared for all policies.</td>
</tr>
<tr>
<td>entry &lt;policy class #&gt;</td>
<td>Optional. Use this keyword to clear statistics of a specific policy class entry.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example clears statistical counters for all policy classes:

ProCurve>enable
ProCurve#clear ip policy-stats

The following example clears statistical counters for the policy class MatchALL:

ProCurve>enable
ProCurve#clear ip policy-stats MatchALL
clear ip prefix-list <listname>

Use the clear ip prefix-list command to clear the IP prefix list hit count shown in the show ip prefix-list detail command output. See show ip prefix-list [detail | summary] <listname> on page 230.

Syntax Description

<listname> Specifies hit count statistics of the IP prefix list to clear.

Default Values

No default value necessary for this command.

Usage Examples

The following example clears the hit count statistics for prefix list test:

ProCurve>enable
ProCurve#clear ip prefix-list test
clear ip route [** | <ip address> <subnet mask>]

Use the `clear ip route` command to remove all learned routes from the IP route table. Static and connected routes are not cleared by this command.

**Syntax Description**

| ** | Deletes all destination routes. |
| <ip address> | Specifies the IP address of the destination routes to be deleted. |
| <subnet mask> | Specifies the subnet mask of the destination routes to be deleted |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example removes all learned routes from the route table:

```
ProCurve>enable
ProCurve#clear ip route **
```
clear lldp counters

Use the clear lldp counters command to reset all LLDP packet counters to 0 on all interfaces.

Syntax Description

No subcommands.

Default Values

There are no default settings for this command.

Usage Examples

The following example resets all LLDP counters:

ProCurve>enable
ProCurve#clear lldp counters
clear lldp counters interface <interface>

Use the clear lldp counters interface command to reset all LLDP packet counters to 0 for a specified interface.

Syntax Description

| <interface> | Clears the information for the specified interface. Type clear lldp counters interface ? for a complete list of applicable interfaces. |

Default Values

No default values are necessary for this command.

Usage Examples

The following example resets the counters on a PPP interface:

ProCurve>enable
ProCurve#clear lldp counters interface ppp 1
**clear lldp neighbors**

Use the `clear lldp neighbors` command to remove all neighbors from this unit’s database. As new local loop demarcation point (LLDP) packets are received, the database will contain information about neighbors included in those frames.

**Syntax Description**

No subcommands.

**Default Values**

There are no default settings for this command.

**Functional Notes**

This command generates output indicating the names of any neighbors deleted from the database and the name of the interface on which the neighbor was learned.

**Usage Examples**

The following example clears LLDP neighbor Switch_1 from the Ethernet interface 0/1:

ProCurve>`enable
ProCurve>enable
ProCurve>enable
ProCurve>clear lldp neighbors
LLDP: Deleted neighbor “Switch_1” on interface eth 0/1
ProCurve#
clear pppoe <ppp interface #>

Use the clear pppoe command to terminate the current PPPoE client session and cause the SROS to attempt to re-establish the session.

Syntax Description

<ppp interface #> PPP interface number.

Default Values

No default value necessary for this command.

Usage Examples

The following example ends the current PPPoE client session for ppp 1:

ProCurve>enable
ProCurve#clear pppoe 1
clear processes cpu max

Use the clear processes cpu max command to clear the maximum CPU usage statistic which is displayed in the show process cpu command output.

Syntax Description
No subcommands.

Default Values
No default value necessary for this command.

Usage Examples
The following example resets the CPU maximum usage statistics:

ProCurve>enable
ProCurve#clear process cpu max
clear processes queue

Use the **clear processes queue** command to clear the contents of the system processing queues.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example clears the contents of the system processing queues:

ProCurve>**enable**
ProCurve#**clear process queue**
clear qos map

Use the clear qos map command to clear the statistics for all defined QoS maps or for maps meeting user-configured specifications.

Variations of this command include the following:
- `clear qos map <map name>`
- `clear qos map <map name> <sequence number>`
- `clear qos map interface <interface id>`

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;map name&gt;</code></td>
<td>Specifies the name of a defined QoS map.</td>
</tr>
<tr>
<td><code>&lt;sequence number&gt;</code></td>
<td>Specifies one of the map’s defined sequence numbers.</td>
</tr>
<tr>
<td><code>&lt;interface&gt;</code></td>
<td>Specifies an interface for which to clear QoS map statistics (for just that interface).</td>
</tr>
<tr>
<td></td>
<td>Type clear cos map ? for a complete list of applicable interfaces.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example clears statistics for all defined QoS maps:
ProCurve#clear qos map

The following example clears statistics for all entries in the priority QoS map:
ProCurve#clear qos map priority

The following example clears statistics in entry 10 of the priority QoS map:
ProCurve#clear qos map priority 10

The following example clears QoS statistics for a specified interface:
ProCurve#clear qos map interface frame-relay 1

Note

The clear counters command clears ALL interface statistics (including QoS map interface statistics).
clear route-map counters <map>

Use the clear route-map counters command to reset route map hit counters.

**Syntax Description**

| <map> | Specifies specific route map to be cleared. |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example clears all route map counters:

ProCurve>enable
ProCurve#clear route-map counters
clear sip location [** | <username>]

Use the clear sip location command to clear session initiation protocol (SIP) location database statistics.

Syntax Description

**
Clears all dynamic location entries.

<username>
Specifies specific username to clear.

Default Values
No default value necessary for this command.

Usage Examples

The following example deletes all dynamic location entries:

ProCurve> enable
ProCurve# clear sip location **
clear sip user-registration

Use the clear sip user-registration command to clear local session initiation protocol (SIP) server registration information.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example clears all SIP server registration information:

```
ProCurve>enable
ProCurve#clear sip user-registration
```
clear spanning-tree counters [interface <interface id>]

The clear spanning-tree counters command clears the following counts: BPDU transmit, BPDU receive, and number of transitions to forwarding state.

Syntax Description

| interface <interface id> | Optional. Specifies a single interface. Enter clear spanning-tree counters ? for a complete list of applicable interfaces. |

Default Values

No default value necessary for this command.

Usage Examples

The following example clears the spanning tree counters for Ethernet 0/1:

ProCurve>enable
ProCurve#clear spanning-tree counters interface eth 0/1
clear spanning-tree detected-protocols [interface <interface id>]

Use the clear spanning-tree detected-protocols command to restart the protocol migration process.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>Optional. Choose the Ethernet interface.</td>
</tr>
<tr>
<td>&lt;interface id&gt;</td>
<td>Optional. Enter a valid interface ID (e.g., 0/1 for Ethernet 0/1).</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Functional Notes

The ProCurve Secure Router has the ability to operate using the rapid spanning-tree protocol or the legacy 802.1D version of spanning-tree. When a BPDU (bridge protocol data unit) of the legacy version is detected on an interface, the ProCurve Secure Router automatically regresses to using the 802.1D spanning-tree protocol for that interface. Issue the clear spanning-tree detected-protocols command to return to rapid spanning-tree operation.

Usage Examples

The following example re-initiates the protocol migration process on eth 0/2:

ProCurve>enable
ProCurve#clear spanning-tree detected-protocols interface ethernet 0/2

The following example re-initiates the protocol migration process on all interfaces:

ProCurve>enable
ProCurve#clear spanning-tree detected-protocols
clear tacacs+ statistics

Use the clear tacacs+ statistics command to delete all terminal access controller access control system (TACACS+) protocol statistics.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following example clears all TACACS+ protocol statistics:

ProCurve>enable
ProCurve#clear tacacs+ statistics
clear user [console <user number> | ssh <user number> | telnet <user number>]

Use the clear user command to detach a user from a given line.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>Detaches a specific console user. Valid range is 0 to 1.</td>
</tr>
<tr>
<td>ssh</td>
<td>Detaches a specific secure shell (SSH) user. Valid range is 0 to 4.</td>
</tr>
<tr>
<td>telnet</td>
<td>Detaches a specific Telnet user. Valid range is 0 to 5.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example detaches the console 1 user:

ProCurve>enable
ProCurve#clear user console 1
clock [auto-correct-dst | no-auto-correct-dst]

The clock auto-correct-dst command allows the unit to automatically correct for Daylight Saving Time (DST). Use the clock no-auto-correct-dst command to disable this feature.

Syntax Description

<table>
<thead>
<tr>
<th>auto-correct-DST</th>
<th>Configures the unit to automatically correct for DST.</th>
</tr>
</thead>
<tbody>
<tr>
<td>no-auto-correct-DST</td>
<td>Disables DST correction.</td>
</tr>
</tbody>
</table>

Default Values

By default DST correction takes place automatically.

Functional Notes

Depending on the clock timezone chosen (see clock timezone <text> on page 61 for more information) one-hour DST correction may be enabled automatically. You may override this default using this command.

Usage Examples

The following example allows for automatic DST correction:

ProCurve>enable
ProCurve#clock auto-correct-dst

The following example overrides the one-hour offset for DST:

ProCurve>enable
ProCurve#clock no-auto-correct-dst
clock set <time> <day> <month> <year>

Use the clock set command to configure the system software clock. For the command to be valid, all fields must be entered. See the Usage Example below for an example.

Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;time&gt;</td>
<td>Sets the time (in 24-hr format) of the system software clock in the format HH:MM:SS (hours:minutes:seconds).</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>Sets the current day of the month (Range: 1 to 31).</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>Sets the current month (Range: January to December). You need only enter enough characters to make the entry unique. This entry is not case-sensitive.</td>
</tr>
<tr>
<td>&lt;year&gt;</td>
<td>Sets the current year (Range: 2000 to 2100).</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example sets the system software clock for 3:42 pm, August 22 2004:

ProCurve>enable
ProCurve#clock set 11:22:00 07 Au 2005
clock timezone <text>

The clock timezone command sets the unit’s internal clock to the timezone of your choice. This setting is based on the difference in time (in hours) between Greenwich Mean Time (GMT) or Central Standard Time (CST) and the timezone for which you are setting up the unit. Use the no form of this command to disable this feature.

Syntax Description

| <text> | Specifies the difference in time (in hours) between Greenwich Mean Time (GMT) or Central Standard Time (CST) and the timezone for which you are setting up the unit. |

Default Values

No default value is necessary for this command.

Note

Depending on the clock timezone chosen, one-hour Daylight Savings Time (DST) correction may be enabled automatically. See clock [auto-correct-dst | no-auto-correct-dst] on page 59 for more information.

Functional Notes

The following list shows sample cities and their timezone codes.

- clock timezone +1-Amsterdam
- clock timezone +1-Belgrade
- clock timezone +1-Brussels
- clock timezone +1-Sarajevo
- clock timezone +1-West-Africa
- clock timezone +10-Brisbane
- clock timezone +10-Canberra
- clock timezone +10-Guam
- clock timezone +10-Hobart
- clock timezone +10-Vladivostok
- clock timezone +11
- clock timezone +12-Auckland
- clock timezone +12-Fiji
- clock timezone +13
- clock timezone +2-Athens
- clock timezone +2-Bucharest
- clock timezone +2-Cairo
- clock timezone +2-Harare
### Functional Notes (cont’d)

<table>
<thead>
<tr>
<th>Function Call</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>clock timezone +2-Helsinki</td>
<td>clock timezone -3-Greenland</td>
</tr>
<tr>
<td>clock timezone +2-Jerusalem</td>
<td>clock timezone -3:30</td>
</tr>
<tr>
<td>clock timezone +3-Baghdad</td>
<td>clock timezone -4-Atlantic-Time</td>
</tr>
<tr>
<td>clock timezone +3-Kuwait</td>
<td>clock timezone -4-Caracas</td>
</tr>
<tr>
<td>clock timezone +3-Moscow</td>
<td>clock timezone -4-Santiago</td>
</tr>
<tr>
<td>clock timezone +3-Nairobi</td>
<td>clock timezone -5</td>
</tr>
<tr>
<td>clock timezone +3:30</td>
<td>clock timezone -5-Bogota</td>
</tr>
<tr>
<td>clock timezone +4-Abu-Dhabi</td>
<td>clock timezone -5-Eastern-Time</td>
</tr>
<tr>
<td>clock timezone +4-Baku</td>
<td>clock timezone -6-Central-America</td>
</tr>
<tr>
<td>clock timezone +4:30</td>
<td>clock timezone -6-Central-Time</td>
</tr>
<tr>
<td>clock timezone +5-Ekaterinburg</td>
<td>clock timezone -6-Mexico-City</td>
</tr>
<tr>
<td>clock timezone +5-Islamabad</td>
<td>clock timezone -6-Saskatchewan</td>
</tr>
<tr>
<td>clock timezone +5:30</td>
<td>clock timezone -7-Arizona</td>
</tr>
<tr>
<td>clock timezone +5:45</td>
<td>clock timezone -7-Mountain-Time</td>
</tr>
<tr>
<td>clock timezone +6-Almaty</td>
<td>clock timezone -8</td>
</tr>
<tr>
<td>clock timezone +6-Astana</td>
<td>clock timezone -9</td>
</tr>
<tr>
<td>clock timezone +6-Sri-Jay</td>
<td>clock timezone 0-(UTC) Universal Coordinated Time</td>
</tr>
<tr>
<td>clock timezone +6:30</td>
<td>clock timezone GMT-Casablanca</td>
</tr>
<tr>
<td>clock timezone +7-Bangkok</td>
<td>clock timezone GMT-Dublin</td>
</tr>
<tr>
<td>clock timezone +7-Kranoysark</td>
<td></td>
</tr>
</tbody>
</table>

### Usage Examples

The following example sets the timezone for Santiago, Chile.

ProCurve>**enable**
ProCurve#**clock timezone -4-Santiago**
configure

Use the `configure` command to enter the Global Configuration mode or to configure the system from memory. See *Global Configuration Mode Command Set* on page 285 for more information.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>terminal</td>
<td>Enters the Global Configuration mode.</td>
</tr>
<tr>
<td>memory</td>
<td>Configures the active system with the commands located in the default configuration file stored in flash memory.</td>
</tr>
<tr>
<td>network</td>
<td>Configures the system from a TFTP network host.</td>
</tr>
<tr>
<td>overwrite-network</td>
<td>Overwrites flash memory from a TFTP network host.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example enters the Global Configuration mode from the Enable Command mode:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#
```
copy [cflash | flash] <filename> boot

Use the copy boot commands to copy the specified file (located in flash memory or on the compact flash card) to the Boot ROM.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cflash</td>
<td>Specifies the memory location for the file to copy as compact flash memory.</td>
</tr>
<tr>
<td>flash</td>
<td>Specifies the memory location for the file to copy as flash memory.</td>
</tr>
<tr>
<td>&lt;filename&gt;</td>
<td>Specifies the name of the source file to copy.</td>
</tr>
<tr>
<td>boot</td>
<td>Specifies the destination memory location for the file copy as the Boot ROM.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Functional Notes**

Updates to the Boot ROM are required periodically to enhance and expand the router’s operation. The Boot Code can now be updated from within the Command Line Interface (CLI) (beginning with software release J03.01) using the copy boot command.

**Usage Examples**

The following example copies the file J03_01-boot.biz (located on the compact flash card) to the Boot ROM:

ProCurve>enable
ProCurve#copy cflash J03_01-boot.biz boot

Upgrading boot code is a critical process that cannot be interrupted. If something were to happen and the process was not able to be completed, it would render your unit inoperable. It is for this reason that during a bootcode upgrade, all other system tasks will be halted. This means packets will not be routed, and all console sessions will not respond during the upgrade process. Once the process finishes, the system will function as it did before. This process will take approximately 20 seconds. DO NOT REMOVE COMPACT FLASH DURING THIS OPERATION!!

Do you want to proceed? [yes/no]
y

WARNING!! A bootcode upgrade has been initiated. Your session will become nonresponsive for the duration of the upgrade (approx. 20 seconds). A message will be sent when the upgrade is completed.

Bootcode upgrade process done. Your session should function normally.

Reading 318189 bytes of code, stand by...

Image is compressed, inflating.................................

Verifying image

Erasing boot sector

Programming boot sector

Success!!!
copy cflash <filename> [cflash <filename> | flash <filename> | startup-config]

Use the `copy cflash` command to copy file located on the compact flash card to a specified destination.

**Syntax Description**

- `<filename>` Specifies the name of the file (located on the compact flash card) to copy. The asterisk (*) can be used as a wildcard to specify a pattern for erasing multiple files. When a wildcard is specified, only files matching the listed pattern are copied. When using a wildcard, leave the destination filename blank because the source files are copied identically (including the filename) to the destination. You cannot use a wildcard when the destination is `startup-config`.
- `cflash <filename>` Specifies the destination memory location for the file copy as compact flash memory and specifies the filename.
- `flash <filename>` Specifies the destination memory location for the file copy as flash memory and specifies the filename.
- `startup-config` Replaces the primary startup-configuration file with a copy of the specified file.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the file `myfile.biz` (located on the compact flash card) to flash memory (naming the new file `newfile.biz`):

```
ProCurve>enable
ProCurve#copy cflash myfile.biz flash newfile.biz
```

The following example creates a copy of the file `myfile.biz` (located on the compact flash card), names the new file `newfile.biz`, and places the new file on the installed compact flash card:

```
ProCurve>enable
ProCurve#copy cflash myfile.biz cflash newfile.biz
```

The following example replaces the startup configuration file with the file `newconfig.txt`.

```
ProCurve>enable
ProCurve#copy cflash newconfig.txt startup-config
```
copy cflash tftp

Use the `copy cflash tftp` command to copy any file on the compact flash card to a specified Trivial File Transfer Protocol (TFTP) server.

**Syntax Description**

After entering `copy cflash tftp`, the SROS prompts for the following information:

- **Address of remote host**: Specifies the IP address of the TFTP server.
- **Source filename**: Specifies the name of the file (located on the compact flash card) to copy to the TFTP server.
- **Destination filename**: Specifies the filename to use when storing the copied file on the TFTP server. (The file will be placed in the default directory established by the TFTP server.)

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the file `myfile.biz` (located on the compact flash card) to the specified TFTP server:

```
ProCurve>enable
ProCurve#copy cflash tftp
Address of remote host?10.200.2.4
Source filename `myfile.biz`
Destination filename `myfile.biz`
Initiating TFTP transfer...
Received 45647 bytes.
Transfer Complete!
```
copy cflash xmodem

Use the copy cflash xmodem command to copy any file on the compact flash card (using the XMODEM protocol) to the terminal connected to the console port. XMODEM capability is provided in terminal emulation software such as HyperTerminal™.

Syntax Description

After entering copy cflash xmodem, the SROS prompts for the following information:

Source filename: Specifies the name of the file to copy from compact flash to the connected terminal.

Default Values

No default value necessary for this command.

Usage Examples

The following example copies the file myfile.biz (located on the compact flash card) to the connected terminal using XMODEM protocol:

ProCurve>enable
ProCurve#copy cflash xmodem

Source filename myfile.biz
Begin the Xmodem transfer now...
Press CTRL+X twice to cancel
CCCCCC

The SROS is now ready to transmit the file on the CONSOLE port (using the XMODEM protocol). The next step in the process may differ depending on the type of terminal emulation software you are using. For HyperTerminal, you will now select Transfer > Receive File and select the destination. Once the transfer is complete, information similar to the following is displayed:

Received 231424 bytes.
Transfer complete
copy console flash <filename>

Use the copy console command to copy the console’s input to a text file. To end copying to the text file, type <Ctrl+D>. The file will be saved in the SROS root directory.

Syntax Description

<filename> Specifies the destination file (located in flash memory) for console input.

Default Values

No default is necessary for this command.

Functional Notes

The copy console command works much like a line editor. Prior to pressing <Enter>, changes can be made to the text on the line. Changes can be made using <Delete> and <Backspace> keys. The text can be traversed using the arrow keys, <Ctrl+A> (to go to the beginning of a line), and <Ctrl+E> (to go to the end of a line). To end copying to the text file, type <Ctrl+D>. The file will be saved in the SROS root directory. Use the dir command to see a list of files in the root directory.

Usage Examples

The following example copies the console input into the file config.txt (located in the SROS root directory):

ProCurve>enable
ProCurve#copy console flash config.txt
**copy flash <filename> [cflash <filename> | flash <filename>] startup-config**

Use the `copy flash` command to copy files located in flash memory to a specified destination.

**Syntax Description**

- `<filename>`: Specifies the name of the file (located on the compact flash card) to copy. The asterisk (*) can be used as a wildcard to specify a pattern for erasing multiple files. When a wildcard is specified, only files matching the listed pattern are copied. When using a wildcard, leave the destination filename blank because the source files are copied identically (including the filename) to the destination. You cannot use a wildcard when the destination is `startup-config`.
- `cflash <filename>`: Specifies the destination memory location for the file copy as compact flash memory and the filename.
- `flash <filename>`: Specifies the destination memory location for the file copy as flash memory and the filename.
- `startup-config`: Replaces the primary startup-configuration file with a copy of the specified file.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example creates a copy of the file `myfile.biz` (located in flash memory), names the new file `newfile.biz`, and places the new file in flash memory:

```
ProCurve>enable
ProCurve#copy flash myfile.biz flash newfile.biz
```

The following example copies the file `myfile.biz` (located in flash memory) to compact flash memory (naming the new file `newfile.biz`):

```
ProCurve>enable
ProCurve#copy flash myfile.biz flash newfile.biz
```

The following example copies the file `new_startup_config` (located in flash memory) to the primary startup-configuration:

```
ProCurve>enable
ProCurve#copy flash new_startup_config startup-config
```
copy flash tftp

Use the copy flash tftp command to copy a file located in flash memory to a specified Trivial File Transfer Protocol (TFTP) server.

Syntax Description

- **Address of remote host:** Specifies the IP address of the TFTP server.
- **Source filename:** Specifies the name of the file (located in flash memory) to copy to the TFTP server.
- **Destination filename:** Specifies the filename to use when storing the copied file on the TFTP server. (The file will be placed in the default directory established by the TFTP server.)

Default Values

No default value necessary for this command.

Usage Examples

The following example copies the software file `J03_01.biz` located in flash memory to a TFTP server:

```
ProCurve>enable
ProCurve#copy flash tftp

Address of remote host? 10.200.2.4
Source filename J03_01.biz
Destination filename J03_01.biz
Initiating TFTP transfer...
Sent 769060 bytes.
Transfer Complete!
```
copy flash xmodem

Use the `copy flash xmodem` command to copy any file located in flash memory (using the XMODEM protocol) to the terminal connected to the console port. XMODEM capability is provided in terminal emulation software such as HyperTerminal™.

**Syntax Description**

After entering `copy flash xmodem`, the SROS prompts for the following information:

| Source filename: | Specifies the name of the file (located in flash memory) to copy using XMODEM. |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the software file `J03_01.biz` (located in flash memory) to the connected terminal using XMODEM protocol:

```
ProCurve>enable
ProCurve#copy flash xmodem

Source filename J03_01.biz
Begin the Xmodem transfer now...
Press CTRL+X twice to cancel
CCCCCCC

The SROS is now ready to transmit the file on the CONSOLE port (using the XMODEM protocol). The next step in the process may differ depending on the type of terminal emulation software you are using. For HyperTerminal, you will now select Transfer > Receive File and select the destination. Once the transfer is complete, information similar to the following is displayed:

Sent 231424 bytes.
Transfer complete
```
copy flash <filename> interface <type> <slot/port>

Use the copy interface command to copy a software file to a specified interface. This command is only valid for modules that contain module-specific software that is independent of the system software.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;filename&gt;</td>
<td>Specifies the filename of source software file. The source file MUST be located in flash memory. All module software files should have a .biz extension.</td>
</tr>
<tr>
<td>&lt;type&gt;</td>
<td>Specifies the type of module to update. Enter the copy flash XXXX interface ? to display a list of all available module types.</td>
</tr>
<tr>
<td>&lt;slot/port&gt;</td>
<td>Specifies the slot and port number of the interface.</td>
</tr>
</tbody>
</table>

Default Values

No default is necessary for this command.

Usage Examples

The following example updates the ADSL interface with the software file J8459A_02_01_01.biz:

ProCurve>enable
ProCurve#copy J8459A_02_01_01.biz interface adsl 0/1
copy running-config [cflash <filename> | flash <filename>]
  startup-config]

Use the `copy running-config` command to create a copy of the current running-configuration and replace the current startup-configuration or save it to a specified memory location (compact flash or system flash).

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cflash &lt;filename&gt;</code></td>
<td>Specifies the destination memory location for the copied file as compact flash and the destination filename.</td>
</tr>
<tr>
<td><code>flash &lt;filename&gt;</code></td>
<td>Specifies the destination memory location for the copied file as system flash memory and the destination filename.</td>
</tr>
<tr>
<td><code>startup-config</code></td>
<td>Replaces the primary startup-configuration file with the current running-configuration.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the current running-configuration file to the startup configuration file located in flash memory:

```
ProCurve>enable
ProCurve#copy running-config startup-config
Building configuration...
NONVOL:/startup-config
Done. Success!
```

The following example copies the current running-configuration file to compact flash memory and names the file `config_01.txt`:

```
ProCurve>enable
ProCurve#copy running-config cflash config_01.txt
Percent Compete 100%
ProCurve#
```
**copy running-config tftp**

Use the `copy running-config tftp` command to copy the current running-configuration file to a specified Trivial File Transfer Protocol (TFTP) server.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address of remote host:</td>
<td>Specifies the IP address of the TFTP server.</td>
</tr>
<tr>
<td>Destination filename:</td>
<td>Specifies the filename to use when storing the copied file on the TFTP server.</td>
</tr>
</tbody>
</table>

*(The file will be placed in the default directory established by the TFTP server.)*

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the current running-configuration file to a TFTP server and names the file `config_01.txt`:

```
ProCurve> enable
ProCurve# copy running-config tftp

Address of remote host? 10.200.2.4
Destination filename config_01.txt
Initiating TFTP transfer...
Sent 3099 bytes.
Transfer Complete!
```
copy running-config xmodem

Use the `copy running-config xmodem` command to copy the current running-configuration file (using the XMODEM protocol) to the terminal connected to the console port. XMODEM capability is provided in terminal emulation software such as HyperTerminal™.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the current running-configuration to the connected terminal using XMODEM protocol:

```
ProCurve>enable
ProCurve#copy running-config xmodem
```

Begin the Xmodem transfer now...
Press CTRL+X twice to cancel
CCCCCCC

The SROS is now ready to transmit the file on the **CONSOLE** port (using the XMODEM protocol). The next step in the process may differ depending on the type of terminal emulation software you are using. For HyperTerminal, you will now select **Transfer > Receive File** and select the destination. Once the transfer is complete, information similar to the following is displayed:

Sent 3704 bytes.
Transfer complete
**copy startup-config [cflash <filename> | flash <filename> | running-config]**

Use the `copy startup-config` command to create a copy of the current startup-configuration and replace the current running-configuration or save it to a specified memory location (compact flash or system flash).

**Syntax Description**

- **cflash <filename>**
  - Specifies the destination memory location for the copied file as compact flash and specifies the filename for the copied file.
- **flash <filename>**
  - Specifies the destination memory location for the copied file as system flash memory and specifies the filename for the copied file.
- **running-config**
  - Replaces the current running configuration file with the primary startup-configuration.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the primary startup-configuration file to the current running-configuration:

```
ProCurve>enable
ProCurve#copy startup-config running-config
Opening and applying file...
```

**Note**

Any changes made to the current running configuration of the router that have not been saved to the startup configuration file (using the `write` command) will be lost when the `copy startup-config running-config` command is entered.

The following example copies the primary startup-configuration file (located in flash memory) to compact flash and names the file `config_01.txt`:

```
ProCurve>enable
ProCurve#copy startup-config cflash config_01.txt
Percent Complete 100%
ProCurve#
```
**copy startup-config tftp**

Use the **copy startup-config tftp** command to copy the current startup-configuration file to a specified Trivial File Transfer Protocol (TFTP) server.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>copy startup-config tftp</strong></td>
<td>After entering <strong>copy startup-config tftp</strong>, the SROS prompts for the following information:</td>
</tr>
<tr>
<td><strong>Address of remote host:</strong></td>
<td>Specifies the IP address of the TFTP server.</td>
</tr>
<tr>
<td><strong>Destination filename:</strong></td>
<td>Specifies the filename to use when storing the copied file on the TFTP server. (The file will be placed in the default directory established by the TFTP server.)</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies the current startup-configuration file to a TFTP server and names the file **startup_01.txt**:

ProCurve>enable
ProCurve#copy startup-config tftp

Address of remote host? 10.200.2.4
Destination filename **startup_01.txt**
Initiating TFTP transfer...
Sent 3099 bytes.
Transfer Complete!
copy startup-config xmodem

Use the `copy startup-config xmodem` command to copy the current startup-configuration file (using the XMODEM protocol) to the terminal connected to the console port. XMODEM capability is provided in terminal emulation software such as HyperTerminal™.

**Syntax Description**
No subcommands.

**Default Values**
No default value necessary for this command.

**Usage Examples**
The following example copies the current startup-configuration to the connected terminal using XMODEM protocol:

```
ProCurve>enable
ProCurve#copy startup-config xmodem

Begin the Xmodem transfer now...
Press CTRL+X twice to cancel
CCCCCCC

The SROS is now ready to transmit the file on the CONSOLE port (using the XMODEM protocol). The next step in the process may differ depending on the type of terminal emulation software you are using. For HyperTerminal, you will now select Transfer > Receive File and select the destination. Once the transfer is complete, information similar to the following is displayed:

Sent 3704 bytes.
Transfer complete
```
copy tftp <destination>

Use the `copy tftp` command to copy a file located on a network Trivial File Transfer Protocol (TFTP) server to a specified destination.

**Syntax Description**

- `<destination>`: Specifies the destination of the file copied from the TFTP server. Valid destinations include: `cflash` (installed compact flash card), `flash` (flash memory), `startup-config` (the primary configuration file), or `running-config` (the current running configuration file).

After entering `copy tftp` and specifying a destination, the SROS prompts for the following information:

  - **Address of remote host:** Specifies the IP address of the TFTP server.
  - **Source filename:** Specifies the Name of the file to copy from the TFTP server.
  - **Destination filename:** Specifies the filename to use when storing the copied file to flash memory. (Valid only for the `copy tftp cflash` and `copy tftp flash` commands.)

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example replaces the current running-configuration file with `new_config.txt` from the TFTP server (10.200.2.4):

```
ProCurve#copy tftp running-config
Address of remote host? 10.200.2.4
Source filename new_config.txt
Initiating TFTP transfer...
Received 4562 bytes.
Transfer Complete!
ProCurve#
```

The following example copies the file `J03_01.biz` from the TFTP server (10.200.2.4) and saves it compact flash memory (naming the copy `SROS.BIZ`):

```
ProCurve#copy tftp cflash
Address of remote host? 10.200.2.4
Source filename J03_01.biz
Destination filename SROS.BIZ
Initiating TFTP transfer...
Received 45647 bytes.
Transfer Complete!
ProCurve#
```
**copy xmodem** <destination>

Use the `copy xmodem` command to copy a file (using the XMODEM protocol) to a specified destination. XMODEM capability is provided in terminal emulation software such as HyperTerminal™.

**Syntax Description**

<destination> Specifies the destination of the copied file.

Valid destinations include: `cflash` (installed compact flash card), `flash` (flash memory), `startup-config` (the configuration file stored in flash memory), or `running-config` (the current running configuration file).

After entering `copy xmodem` and specifying a destination, the SROS prompts for the following information:

*Destination filename:* Specifies the filename to use when storing the copied file to flash memory. (Valid only for the `copy cflash` and `copy flash` commands.)

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example copies a software file (`J03_01.biz`) to flash memory and labels it `SROS.BIZ`:

```
ProCurve#copy xmodem flash
Destination filename SROS.BIZ
Begin the Xmodem transfer now...
Press CTRL+X twice to cancel
CCCCCC
```

The SROS is now ready to accept the file on the CONSOLE port (using the XMODEM protocol). The next step in the process may differ depending on the type of terminal emulation software you are using. For HyperTerminal, you will now select **Transfer > Send File** and browse to the file you wish to copy (`J03_01.biz`). Once the transfer is complete, information similar to the following is displayed:

```
Received 531424 bytes.
Transfer complete.
```
debug aaa

Use the **debug aaa** command to activate debug messages associated with authentication from the AAA subsystem. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The **debug aaa** events include connection notices, login attempts, and session tracking.

**Usage Examples**

The following is sample output for this command:

```
ProCurve>enable
ProCurve#debug aaa
AAA: New Session on portal 'TELNET 0 (172.22.12.60:4867)'.
AAA: No list mapped to 'TELNET 0'. Using 'default'.
AAA: Attempting authentication (username/password).
AAA: RADIUS authentication failed.
AAA: Authentication failed.
AAA: Closing Session on portal 'TELNET 0 (172.22.12.60:4867)'.
```
debug access-list <listname>

Use the `debug access-list` command to activate debug messages (for a specified list) associated with access list operation. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

Syntax Description

`<listname>`  
Specifies a configured access list

Default Values

By default, all debug messages in the SROS are disabled.

Functional Notes

The `debug access-list` command provides debug messages to aid in troubleshooting access list issues.

Usage Examples

The following example activates debug messages for the access list labeled MatchAll:

```
ProCurve>enable
ProCurve#debug access-list MatchAll
```
**debug arp**

Use the `debug arp` command to activate debug messages associated with IP Address Resolution Protocol (ARP) transactions. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
*Turning on a large amount of debug information can adversely affect the performance of your unit.*

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages associated with ARP transactions:

```plaintext
ProCurve>enable
ProCurve#debug arp
```
debug atm events

Use the `debug atm events` command to display events on all ATM ports and all virtual circuits. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**
No subcommands.

**Default Values**
By default, all debug messages in the SROS are disabled.

**Usage Examples**
The following example activates ATM event messages:

```
ProCurve>enable
ProCurve#debug atm events
```
debug atm oam

Use the **debug atm oam** command to display Operation, Administration, and Maintenance (OAM) packets for an ATM virtual circuit descriptor (VCD). Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable debug messages. Variations of this command include the following:

- **debug atm oam <vcd>**
- **debug atm oam <vcd> loopback [end-to-end | segment]**
- **debug atm oam <vcd> loopback [end-to-end | segment] <LLID>**

**Note**: Turning on a large amount of debug information can adversely affect the performance of your unit.

## Syntax Description

- **<vcd>** Shows OAM packets for a specific VCD.
- **loopback** Configures an OAM loopback.
- **end-to-end** Configures an end-to-end OAM loopback.
- **segment** Configures a segment loopback.
- **<LLID>** Specifies 16-byte OAM loopback location ID (LLID).

## Default Values

By default, all debug messages in the SROS are disabled.

## Usage Examples

The following example activates ATM OAM debug messages for VCD 1:

```
ProCurve>enable
ProCurve#debug atm oam 1
```
**debug atm packet**

Use the **debug atm packet** command to activate debug messages associated with packets on ATM ports and virtual circuits. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages. Variations of this command include the following:

```
debug atm packet
debug atm packet interface atm <port id>
default atm packet interface atm <port id> vcd <vcd number>
default atm packet vc <VPI/VCI>
```

**Note**  
 Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface atm &lt;port id&gt;</td>
<td>Shows packets on a specific ATM port and on all virtual circuits.</td>
</tr>
<tr>
<td>vc &lt;VPI/VCI&gt;</td>
<td>Shows packets on a specific virtual circuit identified by the virtual path identifier and virtual channel identifier (VPI/VCI).</td>
</tr>
<tr>
<td>vcd &lt;vcd number&gt;</td>
<td>Shows packets on specific virtual circuit descriptors (VCD).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug ATM packet debug messages on ATM port 1:

```
ProCurve>enable
ProCurve#debug atm packet interface atm 1
```
debug backup

Use the **debug backup** command to activate debug messages associated with backup operation. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

**Note**

*Turning on a large amount of debug information can adversely affect the performance of your unit.*

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The **debug backup** command activates debug messages to aid in the troubleshooting of backup links.

**Usage Examples**

The following example activates debug messages for backup operation:

ProCurve>**enable**
ProCurve#**debug backup**
debug bridge

Use the `debug bridge` command to display messages associated with bridge events. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages.

**Note** Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates bridge debug messages:

ProCurve#`debug bridge`
debug chat-interfaces <chat interface>

Use the debug chat-interfaces command to activate debug messages associated with chat AT command driven interfaces. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note** Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<chat interface> Specifies the chat interface to debug in slot/port format.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages for the chat interface 0/1:

ProCurve>enable
ProCurve#debug chat-interfaces 0/1
debug crypto [ike | ike negotiation | ike client authentication | ike client configuration | ipsec | pki]

Use the `debug crypto` command to activate debug messages associated with IKE and IPSec functions. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

### Syntax Description

- **ike**: Displays all IKE debug messages.
- **ike negotiation**: Displays only IKE key management debug messages (e.g., handshaking).
- **ike client authentication**: Displays IKE client authentication messages as they occur.
- **ike client configuration**: Displays mode-config exchanges as they take place over the IKE SA. It is enabled independently from the `ike negotiation` debug described previously.
- **ipsec**: Displays all IPSec debug messages.
- **pki**: Displays all PKI (public key infrastructure) debug messages.

### Default Values

By default, all debug messages in the SROS are disabled.

### Usage Examples

The following example activates the IPSec debug messages:

```plaintext
ProCurve>enable
ProCurve#debug crypto ipsec
```
**debug data-call**

Use the **debug data-call** command to activate debug messages associated with data call errors and events. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

**Note**

*Turning on a large amount of debug information can adversely affect the performance of your unit.*

---

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages associated with data call errors and events:

```
ProCurve>enable
ProCurve#debug data-call
```
debug demand-routing

Use the `debug demand-routing` command to activate debug messages associated with demand routing errors and events. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages.

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates demand routing error and event messages:

```
ProCurve>enable
ProCurve#debug demand-routing
```
debug dialup-interfaces

Use the `debug dialup-interfaces` command to generate debug messages used to aid in troubleshooting problems with all dialup interfaces such as the modem or the BRI cards. Use the `no` version of this command to disable it.

| Note | Turning on a large amount of debug information can adversely affect the performance of your unit. |

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

When enabled, these messages provide status information on incoming calls, dialing and answering progress, etc. These messages also give information on why certain calls are dropped or rejected. It is beneficial to use this command when troubleshooting backup (in addition to the `debug backup` command).

**Usage Examples**

The following example activates the debug messages for dialup interfaces:

```
ProCurve>enable
ProCurve#debug dialup-interfaces
```
debug dynamic-dns [verbose]

Use the `debug dynamic-dns` command to display debug messages associated with dynamic DNS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note** Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

- **verbose**
  
  Turns on verbose messaging.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates dynamic DNS debug messages:

```
ProCurve>enable
ProCurve#debug dynamic-dns verbose
```
debug firewall

Use the debug firewall command to activate debug messages associated with the SROS firewall operation. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

### Syntax Description

No subcommands.

### Default Values

By default, all debug messages in the SROS are disabled.

### Functional Notes

The debug firewall command activates debug messages to provide real-time information about the SROS stateful inspection firewall operation.

### Usage Examples

The following example activates the debug messages for the SROS stateful inspection firewall:

```
ProCurve>enable
ProCurve#debug firewall
```
**debug firewall alg sip [verbose]**

Use the `debug firewall alg sip` command to activate debug messages associated with Session Initiation Protocol (SIP) information with SROS firewall operation. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

### Syntax Description

- **verbose**
  
  Enables detailed debug messages.

### Default Values

By default, all debug messages in the SROS are disabled.

### Usage Examples

The following example activates debug messages associated with SIP information with SROS firewall operation:

```plaintext
ProCurve>enable
ProCurve#debug firewall alg sip
```

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.
debug frame-relay [events | llc2 | lmi]

Use the debug frame-relay command to activate debug messages associated with the Frame Relay operation. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>events</td>
<td>Activates debug messages for generic Frame Relay events (such as Frame Relay interface state).</td>
</tr>
<tr>
<td>llc2</td>
<td>Activates debug messages for the logical link control layer.</td>
</tr>
<tr>
<td>lmi</td>
<td>Activates debug messages for the local management interface (such as DLCI status signaling state, etc.).</td>
</tr>
</tbody>
</table>

### Default Values

By default, all debug messages in the SROS are disabled.

### Functional Notes

The debug frame-relay command activates debug messages to aid in the troubleshooting of Frame Relay links.

### Usage Examples

The following example activates all possible debug messages associated with Frame Relay operation:

ProCurve>enable
ProCurve#debug frame-relay events
ProCurve#debug frame-relay llc2
ProCurve#debug frame-relay lmi

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.
debug frame-relay multilink <interface>

Use the debug frame-relay multilink command to activate debug messages associated with Frame Relay multilink operation. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

### Syntax Description

| <interface> | Optional. Activates debug messages for the specified interface. Type debug frame-relay multilink ? for a complete list of applicable interfaces. |

### Default Values

By default, all debug messages in the SROS are disabled.

### Usage Examples

The following example activates debug messages associated with multilink operation for all Frame Relay interfaces:

ProCurve>enable  
ProCurve#debug frame-relay multilink
debug interface <interface>

Use the debug interface command to activate debug messages associated with the specified interface. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

< interface >

Activates debug messages for the specified interface. Type debug interface ? for a complete list of applicable interfaces.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The debug interface command activates debug messages to aid in the troubleshooting of physical interfaces.

**Usage Examples**

The following example activates all possible debug messages associated with the Ethernet port:

ProCurve>enable
ProCurve#debug interface ethernet
debug interface adsl events

Use the `debug interface adsl events` command to activate debug messages associated with ADSL events. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages for ADSL events:

```
ProCurve>enable
ProCurve#debug interface adsl events
```
**debug ip bgp [events | in | out | keepalives | updates | updates quiet]**

Use the `debug ip bgp` command to activate debug messages associated with IP BGP. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>events</td>
<td>Displays significant BGP events such as a neighbor state change.</td>
</tr>
<tr>
<td>in/out</td>
<td>Displays the same information as <code>debug ip bgp</code>, but limits messages to the</td>
</tr>
<tr>
<td></td>
<td>specified direction (in or out).</td>
</tr>
<tr>
<td>keepalives</td>
<td>Displays BGP keepalive packets.</td>
</tr>
<tr>
<td>updates</td>
<td>Displays BGP updates for all neighbors.</td>
</tr>
<tr>
<td>updates quiet</td>
<td>Displays summary information about BGP neighbor updates. (Note: <code>updates quiet</code> displays a one-line summary of what <code>update</code> displays in 104 lines.)</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

If no arguments are given, the `debug ip bgp` command displays general BGP events such as sent/received message summaries, route processing actions, and results. Keepalive packets are not debugged with this command.

**Usage Examples**

The following example enables debug messages on general outbound BGP messages and events:

```
ProCurve#debug ip bgp out
07:42:39: BGP OUT 10.15.240.1[2]: Transmitting msg, type=UPDATE (2), len=142
```
debug ip dhcp-client

Use the debug ip dhcp-client command to activate debug messages associated with DHCP client operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**  Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The debug ip dhcp-client command activates debug messages to provide information on DHCP client activity in the SROS. The SROS DHCP client capability allows interfaces to dynamically obtain an IP address from a network DHCP server.

**Usage Examples**

The following example activates debug messages associated with DHCP client activity:

ProCurve>enable
ProCurve#debug ip dhcp-client
debug ip dhcp-server

Use the `debug ip dhcp-server` command to activate debug messages associated with Dynamic Host Configuration Protocol (DHCP) server operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**

`Turning on a large amount of debug information can adversely affect the performance of your unit.`

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The `debug ip dhcp-server` command activates debug messages to provide information on DHCP server activity in the SROS. The SROS DHCP server capability allows the SROS to dynamically assign IP addresses to hosts on the network.

**Usage Examples**

The following example activates debug messages associated with DHCP server activity:

```
ProCurve>enable
ProCurve#debug ip dhcp-server
```
debug ip dns-client

Use the `debug ip dns-client` command to activate debug messages associated with DNS (domain naming system) client operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
*Turning on a large amount of debug information can adversely affect the performance of your unit.*

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The `debug ip dns-client` command activates debug messages to provide information on DNS client activity in the SROS. The IP DNS capability allows for DNS-based host translation (name-to-address).

**Usage Examples**

The following example activates debug messages associated with DNS client activity:

```
ProCurve>enable
ProCurve#debug ip dns-client
```
**debug ip dns-proxy**

Use the `debug ip dns-proxy` command to activate debug messages associated with DNS (domain naming system) proxy operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The `debug ip dns-proxy` command activates debug messages to provide information on DNS proxy activity in the SROS. The IP DNS capability allows for DNS-based host translation (name-to-address).

**Usage Examples**

The following example activates debug messages associated with DNS proxy activity:

```plaintext
ProCurve>enable
ProCurve#debug ip dns-proxy
```
debug ip http [verbose]

Use the debug ip http command to activate debug messages associated with HyperText Transfer Protocol (HTTP) operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**  
*Turning on a large amount of debug information can adversely affect the performance of your unit.*

**Syntax Description**

| verbose        | Activates detailed debug messages for HTTP operation. |

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages associated with HTTP activity:

ProCurve>enable
ProCurve#debug ip http
**debug ip icmp [send | recv]**

Use the `debug ip icmp` command to show all ICMP messages as they come into the router or are originated by the router. If an optional keyword (`send` or `recv`) is not used, all results are displayed. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

| send | Optional keyword which allows you to only display ICMP messages sent by the router. |
| recv | Optional keyword which allows you to only display ICMP messages received by the router. |

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates the `debug ip icmp` send and receive messages for the SROS:

```
ProCurve>enable
ProCurve#debug ip icmp
```

ICMP SEND: From (0.0.0.0) to (172.22.14.229) Type=8 Code=0 Length=72 Details:echo request
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=0 Code=0 Length=72 Details:echo reply
ICMP SEND: From (0.0.0.0) to (172.22.14.229) Type=8 Code=0 Length=72 Details:echo request
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=0 Code=0 Length=72 Details:echo reply
ICMP RECV: From (172.22.255.200) to (10.100.23.19) Type=11 Code=0 Length=36 Details:TTL equals 0 during transit
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=3 Code=3 Length=36 Details:port unreachable
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=3 Code=3 Length=36 Details:port unreachable
debug ip igmp <group-address>

Use the debug ip igmp command to enable debug messages for IGMP transactions (including helper activity). Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note** Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<group-address> Optional. IP address of a multicast group.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example enables IGMP debug messages for the specified multicast group:

ProCurve>enable
ProCurve#debug ip igmp 10.1.1.1
debug ip mrouting

Use the debug ip mrouting command to activate debug messages associated with multicast table routing events. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following sample activates ip mrouting debug messages:

```
ProCurve>enable
ProCurve#debug ip mrouting
```
**debug ip ospf**

Use the `debug ip ospf` command to activate debug messages associated with OSPF routing operations. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adj</td>
<td>Displays OSPF adjacency events.</td>
</tr>
<tr>
<td>database-timer</td>
<td>Displays OSPF database timer.</td>
</tr>
<tr>
<td>events</td>
<td>Displays OSPF events.</td>
</tr>
<tr>
<td>flood</td>
<td>Displays OSPF flooding.</td>
</tr>
<tr>
<td>hello</td>
<td>Displays OSPF hello events.</td>
</tr>
<tr>
<td>lsa-generation</td>
<td>Displays OSPF link state advertisement generation.</td>
</tr>
<tr>
<td>packet</td>
<td>Displays OSPF packets.</td>
</tr>
<tr>
<td>retransmission</td>
<td>Displays OSPF retransmission events.</td>
</tr>
<tr>
<td>spf</td>
<td>Displays OSPF shortest-path-first calculations.</td>
</tr>
<tr>
<td>tree</td>
<td>Displays OSPF database tree.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following is an example of `debug ip ospf` command results:

```
ProCurve>enable
ProCurve#debug ip ospf flood

OSPF: Update LSA: id=00000000 rtid=192.168.2.13 area=0.0.0.0 type=1
OSPF: Update LSA: id=00000000 rtid=192.168.2.13 area=0.0.0.0 type=1
OSPF: Queue delayed ACK lasid=00000000 lsartid=192.168.2.13 nbr=192.168.2.13
OSPF: Rx ACK lasid=00000000 lsartid=192.168.2.13 nbr=192.168.2.13
OSPF: Received LSA ACK LSA_ID=192.168.2.13 LSA_RT_ID=192.168.2.13
OSPF: Rx ACK lasid=00000000 lsartid=192.168.2.13 nbr=192.168.2.13
OSPF: Received LSA ACK LSA_ID=0.0.0.0 LSA_RT_ID=0.0.0.0
OSPF: Sending delayed ACK
```

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.
**debug ip packet [detail | dump | <accesslist>]**

Use the `debug ip packet` command to display debug messages associated with protocol-independent multicast (PIM) sparse assert transactions. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages. Variations of this command include the following:

- `debug ip packet`
- `debug ip packet detail`
- `debug ip packet dump`
- `debug ip packet <accesslist>`
- `debug ip packet <accesslist> detail`
- `debug ip packet <accesslist> dump`

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

- **detail**
  - Optional. Displays IP packet detailed information.

- **dump**
  - Optional. Displays IP packet detailed information on the console or Telnet terminal session.

  Note: The console stream can be captured to a log file and used as an input file for display with ETHEREAL by using `text2pcap.exe`, which is a part of the ETHEREAL distribution.

  Execute as follows: `text2pcap -l 101 <input_file> <output_file>`

  Next, open the output file with ETHEREAL for display and decode. The typical lower layer information in ETHEREAL may not be present. This converted capture file is treated as a raw IP capture and also has no timestamp data. Remember to take advantage of access control lists (ACLs) to narrow down the amount of data being processed with this facility. This is a CPU intensive operation and also disables any fast flow/fast cache routing.

- `<accesslist>`
  - Optional. Enter the access-list name.

**Default Values**

By default, all debug messages in the SROS are disabled.
Usage Examples

The following is sample output for the debug ip packet command:

ProCurve>enable
ProCurve#debug ip packet

IP: s= 192.168.8.101 (eth 0/1) d=192.168.7.2 (eth 0/2) g= 192.168.7.2, forward
IP: s= 192.168.7.2 (eth 0/2) d=192.168.8.101 (eth 0/1) g= 192.168.8.101, forward
IP: s= 192.168.8.101 (eth 0/1) d=192.168.7.2 (eth 0/2) g= 192.168.7.2, forward
IP: s= 192.168.7.2 (eth 0/2) d=192.168.8.101 (eth 0/1) g= 192.168.8.101, forward

Where:

s=192.168.8.101 (eth 0/1) indicates source address and interface of received packet.
d=192.168.7.2 (eth 0/2) indicates destination address and interface from which the packet is being sent.
g=192.168.7.2 indicates the address of the next hop gateway.
forward indicates the router is forwarding this packet.
debug ip pim-sparse

Use the debug ip pim-sparse command to display all protocol-independent multicast (PIM) sparse mode information. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**
No subcommands.

**Default Values**
By default, all debug messages in the SROS are disabled.

**Usage Examples**
The following example activates all PIM sparse mode messages:

```plaintext
ProCurve>enable
ProCurve#debug ip pim-sparse
```
debug ip pim-sparse assert [event | state] <address>

Use the `debug ip pim-sparse assert` command to display debug messages associated with protocol-independent multicast (PIM) sparse assert transactions. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Displays PIM sparse assert events.</td>
</tr>
<tr>
<td>state</td>
<td>Displays PIM sparse assert state changes.</td>
</tr>
<tr>
<td>&lt;address&gt;</td>
<td>Specifies group address to filter.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all PIM sparse assert event messages:

```
ProCurve>enable
ProCurve#debug ip pim-sparse assert event
```
debug ip pim-sparse hello

Use the debug ip pim-sparse hello command to display protocol-independent multicast (PIM) sparse mode hello transactions. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

---

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates PIM sparse mode hello messages:

ProCurve>enable  
ProCurve#debug ip pim-sparse hello
debug ip pim-sparse joinprune [event | state] <address>

Use the **debug ip pim-sparse joinprune** command to display protocol-independent multicast (PIM) sparse mode join and prune transactions. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable debug messages.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Displays PIM sparse join and prune events.</td>
</tr>
<tr>
<td>state</td>
<td>Displays PIM sparse join and prune state changes.</td>
</tr>
<tr>
<td>&lt;address&gt;</td>
<td>Specifies group address to filter.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates PIM sparse mode messages for all join and prune events and state changes:

```
ProCurve>enable
ProCurve#debug ip pim-sparse joinprune
```

**Note**: Turning on a large amount of debug information can adversely affect the performance of your unit.
debug ip pim-sparse packets [in | out] <interface> <interface id>

Use the `debug ip pim-sparse packets` command to display protocol-independent multicast (PIM) sparse mode packet information. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>Displays messages for inbound PIM sparse packets</td>
</tr>
<tr>
<td>out</td>
<td>Displays messages for outbound PIM sparse packets.</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>Specifies specific interface. Type `debug ip pim-sparse packets [in</td>
</tr>
<tr>
<td>&lt;interface id&gt;</td>
<td>Specifies a valid interface ID.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all PIM sparse packet messages (both inbound and outbound):

ProCurve>enable
ProCurve#debug ip pim-sparse packets
debug ip pim-sparse register [event | state] <address>

Use the `debug ip pim-sparse register` command to display protocol-independent multicast (PIM) sparse source registration messages. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Displays PIM sparse register events.</td>
</tr>
<tr>
<td>state</td>
<td>Displays PIM sparse register state changes.</td>
</tr>
<tr>
<td>&lt;address&gt;</td>
<td>Specifies group address to filter.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all PIM sparse registration state changes:

```
ProCurve> enable
ProCurve# debug ip pim-sparse register state
```
debug ip policy

Use the **debug ip policy** command to display policy-based routing events. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable debug messages.

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.

### Syntax Description

No subcommands.

### Default Values

By default, all debug messages in the SROS are disabled.

### Usage Examples

The following example activates policy-based routing event messages:

ProCurve> enable
ProCurve# debug ip policy
debug ip rip [events]

Use the debug ip rip command to activate debug messages associated with Routing Information Protocol (RIP) operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**

> Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>events</td>
<td>Optional. Use this optional keyword to display only RIP protocol events.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The debug ip rip command activates debug messages to provide information on RIP activity in the SROS. RIP allows hosts and routers on a network to exchange information about routes.

**Usage Examples**

The following example activates debug messages associated with RIP activity:

ProCurve>enable
ProCurve#debug ip rip
debug ip routing

Use the `debug ip routing` command to activate debug messages associated with routing table events. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.

Syntax Description

No subcommands.

Default Values

By default, all debug messages in the SROS are disabled.

Usage Examples

The following sample activates `ip routing` debug messages:

```
ProCurve>enable
ProCurve#debug ip routing
```
debug ip tcp [events]

Use the **debug ip tcp events** command to activate debug messages associated with significant TCP events such as state changes, retransmissions, session aborts, etc., in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

**Note**
These debug events are logged for packets that are sent or received from the router. Forwarded TCP packets are not included.
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

| events | Optional. Displays only TCP protocol events. |

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

In the **debug ip tcp events** information, TCB stands for TCP task control block. The numbers which sometimes appear next to TCB (e.g., **TCB5** in the following example) represent the TCP session number. This allows you to differentiate debug messages for multiple TCP sessions.

**Usage Examples**

The following is sample output for this command:

ProCurve>enable
ProCurve#debug ip tcp events

2003.02.17 07:40:56 IP:TCP EVENTS TCB5: state change: FREE->SYNRCVD
2003.02.17 07:41:06 IP:TCP EVENTS TCB5: Connection aborted -- error = RESET
2003.02.17 07:41:06 IP:TCP EVENTS TCB5: De-allocating tcb
**debug ip tcp md5**

Use the `debug ip tcp md5` command to activate debug messages that detail the results of each incoming TCP packet’s MD5 authentication with an internal route in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable debug messages.

**Note**  
*Turning on a large amount of debug information can adversely affect the performance of your unit.*

**Syntax Description**

*No subcommands.*

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

Debug messages will only be generated for TCP ports that have MD5 authentication enabled.

**Usage Examples**

The following example activates the display of these debug messages:

```
ProCurve#debug ip tcp md5
```
**debug ip udp**

Use the `debug ip udp` command to activate debug messages associated with UDP send and receive events in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
These debug events are logged for packets that are sent or received from the router. Forwarded UDP packets are not included.

**Caution**  
The overhead associated with this command takes up a large portion of your router’s resources and at times can halt other router processes. It is best to only use the command during times when the network resources are in low demand (non-peak hours, weekends, etc.).

**Syntax Description**
No subcommands.

**Default Values**
By default, all debug messages in the SROS are disabled.

**Functional Notes**
In the `debug ip udp` information, the message `no listener` means that there is no service listening on this UDP port (i.e., the data is discarded).

**Usage Examples**
The following is sample output for this command:

```
ProCurve>enable
ProCurve#debug ip udp

```
debug isdn [cc-ie | cc-messages | endpoint | interface | l2-formatted | l2-messages] bri <interface id>

Use the debug isdn bri command to activate debug messages associated with integrated services digital network (ISDN) events in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc-ie</td>
<td>Displays call control information elements.</td>
</tr>
<tr>
<td>cc-messages</td>
<td>Displays call control messages.</td>
</tr>
<tr>
<td>endpoint</td>
<td>Displays endpoint events.</td>
</tr>
<tr>
<td>interface</td>
<td>Displays ISDN interface events.</td>
</tr>
<tr>
<td>l2-formatted</td>
<td>Displays layer 2 formatted messages.</td>
</tr>
<tr>
<td>l2-messages</td>
<td>Displays layer 2 messages.</td>
</tr>
<tr>
<td>&lt;interface id&gt;</td>
<td>Specifies the ISDN interface. Range is 1 to 255.</td>
</tr>
</tbody>
</table>

### Default Values

By default, all debug messages in the SROS are disabled.

### Usage Examples

The following example activates all layer 2 formatted messages:

ProCurve>enable  
ProCurve#debug isdn l2-formatted
debug isdn events

Use the `debug isdn events` command to activate debug messages associated with ISDN events in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example activates debug messages associated with ISDN activity:

```
ProCurve>enable
ProCurve#debug isdn events
```
debug isdn resource-manager

Use the `debug isdn resource-manager` command to activate integrated services digital network (ISDN) resource manager errors and messages. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  Turning on a large amount of debug information can adversely affect the performance of your unit.

### Syntax Description

No subcommands.

### Default Values

By default, all debug messages in the SROS are disabled.

### Usage Examples

The following example activates debug messages associated with the ISDN resource manager:

```
ProCurve> enable
ProCurve# debug isdn resource-manager
```
debug isdn verbose

Use the **debug isdn verbose** command to activate all debug messages associated with integrated services digital network (ISDN) events in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

| Note | Turning on a large amount of debug information can adversely affect the performance of your unit. |

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all debug messages associated with ISDN activity:

```
ProCurve>enable
ProCurve#debug isdn verbose
```
debug lldp [rx | tx] verbose

Use the **debug lldp** command to display debug output for all LLDP receive and transmit packets. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of the command to disable debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

- **rx**: Shows information about received packets.
- **tx**: Shows information about transmitted packets.
- **verbose**: Shows detailed debugging information.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all possible debug messages associated with LLDP operation:

```
ProCurve#debug lldp rx
ProCurve#debug lldp tx
ProCurve#debug lldp verbose
```
debug port-auth [general | packet [both | rx | tx] | supp-sm]

Use the `debug port-auth` command to generate debug messages used to aid in troubleshooting problems during the port authentication process. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the `no` form of the command to disable the messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>Optional. Displays configuration changes to the port authentication system.</td>
</tr>
<tr>
<td>packet</td>
<td>Optional. Displays information for packet exchange in transmit-only, receive-only or both directions.</td>
</tr>
<tr>
<td>both</td>
<td>Optional. Displays packet exchange information in both receive and transmit directions.</td>
</tr>
<tr>
<td>rx</td>
<td>Optional. Displays packet exchange information in the receive-only direction.</td>
</tr>
<tr>
<td>tx</td>
<td>Optional. Displays packet exchange information in the transmit-only direction.</td>
</tr>
<tr>
<td>supp-sm</td>
<td>Optional. Displays information pertaining to the supplicant state machine.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates port authentication debug information on received packets:

ProCurve>enable
ProCurve#debug port-auth packet rx
**debug ppp [authentication | errors | negotiation | verbose]**

Use the `debug ppp` command to activate debug messages associated with point-to-point protocol (PPP) operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

### Syntax Description

- **authentication**: Activates debug messages pertaining to PPP authentication (CHAP, PAP, EAP, etc.).
- **errors**: Activates debug messages that indicate a PPP error was detected (mismatch in negotiation authentication, etc.).
- **negotiation**: Activates debug messages associated with PPP negotiation.
- **verbose**: Activates detailed debug messages for PPP operation.

### Default Values

By default, all debug messages in the SROS are disabled.

### Functional Notes

The `debug ppp` command activates debug messages to provide information on PPP activity in the system. PPP debug messages can be used to aid in troubleshooting PPP links.

### Usage Examples

The following example activates debug messages associated with PPP authentication activity:

```
ProCurve>enable
ProCurve#debug ppp authentication
```
**debug pppoe client**

Use the `debug pppoe client` command to activate debug messages associated with point-to-point protocol over Ethernet (PPPoE) operation in the SROS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages associated with PPPoE activity:

```
ProCurve>enable
ProCurve#debug pppoe client
```

*Note*  
Turning on a large amount of debug information can adversely affect the performance of your unit.
**debug radius**

Use the `debug radius` command to enable debug messages from the RADIUS subsystem. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

---

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

---

**Syntax Description**

No subcommands.

---

**Default Values**

By default, all debug messages in the SROS are disabled.

---

**Functional Notes**

The `debug radius` messages show the communication process with the remote RADIUS servers.

---

**Usage Examples**

The following is an example output for the `debug radius` command:

```
ProCurve>enable
ProCurve#debug radius
RADIUS AUTHENTICATION: Sending packet to 172.22.48.1 (1645).
RADIUS AUTHENTICATION: Received response from 172.22.48.1.
```
debug sip [location | manager | proxy <subsource>]

Use the 
`debug sip` command to activate debug messages associated with Session Initiation Protocol (SIP) events. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Activates SIP location database event debug messages.</td>
</tr>
<tr>
<td>manager</td>
<td>Activates SIP stack manager event debug messages.</td>
</tr>
<tr>
<td>proxy &lt;subsource&gt;</td>
<td>Activates SIP proxy event debug messages. Input for specifying a subsoure is optional.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all debug messages associated with SIP CLDU events:

```
ProCurve>enable
ProCurve#debug sip
```
**debug sip** [user-registration <extension> | trunk-registration <Txx> <identity>]

Use the **debug sip** command to activate debug messages associated with Session Initiation Protocol (SIP) events. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

**Note**
Refer to debug sip [location | manager | proxy <subsource>] on page 134 for more information.

Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

- **user-registration <extension>**
  Activates SIP user-registration event debug messages. Specifying a particular trunk is optional.

- **trunk-registration <Txx> <identity>**
  Activates SIP trunk-registration event debug messages. Specifying a particular trunk is optional. For example: Txx (T01) where xx is the trunk’s two-digit identifier and <identity> is the specific name associated with the trunk.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all debug messages associated with SIP stack manager events for all registered users:

ProCurve>**enable**
ProCurve#**debug sip user-registration**
debug sip stack [debug | errors | exceptions | info | messages | verbose | warnings]

Use the `debug sip stack` command to activate debug messages associated with Session Initiation Protocol (SIP) stack events. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug</td>
<td>Activates SIP stack debug event debug messages.</td>
</tr>
<tr>
<td>errors</td>
<td>Activates SIP stack error event debug messages.</td>
</tr>
<tr>
<td>exceptions</td>
<td>Activates SIP stack exception event debug messages.</td>
</tr>
<tr>
<td>info</td>
<td>Activates SIP stack info event debug messages.</td>
</tr>
<tr>
<td>messages</td>
<td>Activates all SIP debug messages.</td>
</tr>
<tr>
<td>verbose</td>
<td>Activates all SIP stack event debug messages.</td>
</tr>
<tr>
<td>warnings</td>
<td>Activates SIP stack warning event debug messages.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates all debug messages associated with SIP stack events:

```
ProCurve>enable
ProCurve#debug sip stack
```
**debug sntp**

Use the `debug sntp` command to enable debug messages associated with the Simple Network Time Protocol (SNTP). All SNTP Packet Exchanges and time decisions are displayed with these debugging events enabled. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**

*Turning on a large amount of debug information can adversely affect the performance of your unit.*

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Functional Notes**

The `debug sntp` command activates debug messages to aid in troubleshooting SNTP protocol issues.

**Usage Examples**

The following is an example output for the `debug sntp` command:

```
ProCurve>enable
ProCurve#debug sntp
ProCurve#config term
ProCurve(config)#sntp server timeserver.localdomain
2002.12.11 15:06:37 SNTP.CLIENT sent Version 1 SNTP time request to 172.27.45.57
2002.12.11 15:06:37 SNTP.CLIENT received SNTP reply packet from 172.27.45.57
2002.12.11 15:06:37 SNTP.CLIENT setting time to 12-11-2002 15:06:02
2002.12.11 15:06:37 SNTP.CLIENT waiting for 86400 seconds for the next poll interval
```
debug spanning-tree [config | events | general | topology]

Use the debug spanning-tree command to enable the display of spanning-tree debug messages. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the no form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Enables the display of spanning-tree debug messages when configuration changes occur.</td>
</tr>
<tr>
<td>events</td>
<td>Enables the display of debug messages when spanning-tree protocol events occur.</td>
</tr>
<tr>
<td>general</td>
<td>Enables the display of general spanning-tree debug messages.</td>
</tr>
<tr>
<td>topology</td>
<td>Enables the display of debug messages when spanning-tree protocol topology events occur.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example enables the display of general spanning-tree debug messages:

ProCurve>enable
ProCurve#debug spanning-tree general
debug spanning-tree bpdu [receive | transmit | all]

Use the **debug spanning-tree bpdu** command to display BPDU (bridge protocol data unit) debug messages. When enabled, a debug message is displayed for each BPDU packet that is transmitted or received by the unit. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>receive</td>
<td>Displays debug messages for BPDU packets received by the unit.</td>
</tr>
<tr>
<td>transmit</td>
<td>Displays debug messages for BPDU packets transmitted by the unit.</td>
</tr>
<tr>
<td>all</td>
<td>Displays debug messages for BPDU packets that are transmitted and received by the unit.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example displays debug messages for BPDU packets that are transmitted and received by the unit:

ProCurve>enable
ProCurve#debug spanning-tree bpdu all

**Note**

Turning on a large amount of debug information can adversely affect the performance of your unit.
debug system

Use the **debug system** command to enable debug messages associated with system events (i.e., login, logouts, etc.). Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

**Note**  
*Turning on a large amount of debug information can adversely affect the performance of your unit.*

**Syntax Description**

No subcommands.

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages associated with system information:

```
ProCurve>enable
ProCurve#debug system
```
debug tacacs+ packets

Use the `debug tacacs+ packets` command to activate debug messages associated with terminal access controller access control system (TACACS+) protocol. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note** *Turning on a large amount of debug information can adversely affect the performance of your unit.*

### Syntax Description

- **events**: Activates TACACS+ event debug messages.
- **packets**: Activates TACACS+ packet debug messages.

### Default Values

By default, all debug messages in the SROS are disabled.

### Usage Examples

The following example activates debug messages associated with the TACACS+ protocol:

```
ProCurve>enable
ProCurve#debug tacacs+ packets
```
debug tftp [client packets | server events | server packets]

Use the `debug tftp packets` command to activate debug messages associated with Trivial File Transfer Protocol (TFTP) packets. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the `no` form of this command to disable the debug messages.

**Note**  
Turning on a large amount of debug information can adversely affect the performance of your unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client packets</td>
<td>Activates TFTP client packet debug messages.</td>
</tr>
<tr>
<td>server events</td>
<td>Activates TFTP server event debug messages.</td>
</tr>
<tr>
<td>server packets</td>
<td>Activates TFTP server packet debug messages.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all debug messages in the SROS are disabled.

**Usage Examples**

The following example activates debug messages associated TFTP server packets:

```plaintext
ProCurve>enable
ProCurve#debug tftp server packets
```
dir [ * ]

Use the `dir` command to display a directory list of all files on the system in flash memory or all files on the system in flash memory matching the specified pattern.

**Syntax Description**

* Optional. When a wildcard is specified, only files located in flash memory matching the listed pattern are displayed.

When no wildcard is specified, the entire contents of flash memory is displayed.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output from the `dir` command specifying a list of all .biz files:

```
ProCurve> enable
ProCurve# dir *.biz
4206603 J03_01.biz
4206603 SROS.biz
4039977J02_02A.biz
24208408 bytes used, 4915176 available, 29123584 total
```
**dir [cflash | flash] [ * ]**

Use the `dir` command to display a directory list of all files on the system in flash memory or on the installed compact flash card or all files on the system in flash memory or on the installed compact flash card matching the specified pattern.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cflash</td>
<td>Specifies files located on the installed compact flash card.</td>
</tr>
<tr>
<td>flash</td>
<td>Specifies files located on the system in flash memory.</td>
</tr>
<tr>
<td>*</td>
<td>Optional. When a wildcard is specified, only files located in the specified location matching the listed pattern are displayed. When no wildcard is specified, the entire contents of flash memory is displayed.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output from the `dir` command specifying a list of all .biz files found on the installed compact flash card:

```
ProCurve>enable
ProCurve#dir cflash *.biz
4206603 HP7203A-08-00-23b-HP1-E.biz
284007 HP7203B-boot-08-01-01-HP.biz
4234845 HP7203A-08-01-01-HP-E.biz
284238 HP7203B-boot-08-01-02-HPatp.biz
4038590 HP7203A-08-01-02-HPatp-E.biz
285416 J01_01_02-boot.biz
4039977 J01_01_02.biz
4043024 J01_01_03.biz
2649600 ericcode.biz
24208408 bytes used, 4915176 available, 29123584 total
```
disable

Use the disable command to exit the Enable Command mode and enter the Basic Command mode.

Syntax Description
No subcommands.

Default Values
No default value necessary for this command.

Usage Examples
The following example exits the Enable Command mode and enters the Basic Command mode:

ProCurve#disable
ProCurve>
enable

Use the enable command to enter a password for the Enable mode.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Functional Notes**

The Enable Command mode provides access to operating and configuration parameters and should be password protected to prevent unauthorized use. Use the enable password command (found in the Global Configuration mode) to specify an Enable Command mode password. If the password is set, access to the Enable Commands (and all other “privileged” commands) is only granted when the correct password is entered. Refer to enable password [md5] <password> on page 330 for more information.

**Usage Examples**

The following example enters the Enable Command mode and defines an Enable Command mode password:

ProCurve>enable
Password: *****
ProCurve#
erase [SROS.BIZ | startup-config]

Use the erase command to erase the specified file.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SROS.BIZ</strong></td>
<td>Erases the SROS.BIZ file stored in flash memory. The SROS.BIZ file is the current software. If the autosynch feature is enabled (using the global configuration autosynch-mode command), the SROS.BIZ file is removed from both flash and compact flash.</td>
</tr>
<tr>
<td><strong>startup-config</strong></td>
<td>Erases the startup configuration file stored in flash memory. If the autosynch feature is enabled (using the autosynch command), the startup-config file is removed from both flash and compact flash.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example erases the startup configuration file stored in flash memory:

ProCurve>enable
ProCurve#erase startup-config

If a new startup-configuration file is not specified before power-cycling the unit, the SROS will initialize using a default configuration.
erase [cflash | flash] <filename>

Use the erase cflash or erase flash command to erase the specified file from the system flash memory or an installed compact flash card.

**Syntax Description**

- **cflash**: Specifies the location of the file to erase as the installed compact flash card.
- **flash**: Specifies the location of the file to erase as the system flash memory.
- **<filename>**: Specifies the name of the file to erase. The asterisk (*) can be used as a wildcard to specify a pattern for erasing multiple files. When a wildcard is specified, only files matching the listed pattern are erased.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example erases the file **myfile.biz** stored in flash memory:

```
ProCurve>enable
ProCurve#erase flash myfile.biz
```
erase file-system cflash

Use the erase file-system cflash command to erase all files on the installed compact flash card.

**Note** Erasing the file system is equivalent to formatting the compact flash card.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example erases all files located on the installed compact flash card:

ProCurve>enable
ProCurve#erase file-system cflash
events

Use the `events` command to enable event reporting to the current CLI session. Use the `no` form of this command to disable all event reporting to the current CLI session.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is enabled.

**Usage Examples**

The following example enables event reporting:

```
ProCurve>enable
ProCurve#events
```
**exception report generate**

Use the `exception report generate` command to immediately generate an exception report that contains a snapshot of all current system processes.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example immediately generates an exception report:

```
ProCurve>enable
ProCurve#exception report generate

Exception report generated.

ProCurve#show file flash exception-report-20050726071500

Using 47428 bytes
```

```
===================================================================== 
VERSION
===================================================================== 

ProCurve Secure Router 7102dl
SROS Version: J03.01.00
  Checksum: 5D5AE64E, built on: Mon Jun 20 13:31:52 2005
Boot ROM version J03.01.00
  Checksum: B1BC, built on: Mon Jul 18 13:11:02 2005
Copyright (c) 2005-2005, Hewlett-Packard, Co.
Platform: ProCurve Secure Router 7102dl
Serial number US449TR053
Flash: 33554432 bytes  DRAM: 134217727 bytes

System uptime is 0 days, 0 hours, 14 minutes, 40 seconds

Current system image:  "CFLASH:/SROS.BIZ"
Current configuration-file:  "CFLASH:/startup-config"
```
Configured system image path:
Primary: "CFLASH:/SROS.BIZ"
Backup: "NONVOL:/SROS.BIZ"

Configured configuration-file path:
Primary: "CFLASH:/startup-config"
Backup: "NONVOL:/startup-config"

CORE DUMP

BUFFER USERS
Number of users: 9

<table>
<thead>
<tr>
<th>Rank</th>
<th>User</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>fixedsize</td>
<td>128</td>
</tr>
<tr>
<td>2</td>
<td>0x00873a50</td>
<td>128</td>
</tr>
<tr>
<td>3</td>
<td>0x00162530</td>
<td>84</td>
</tr>
<tr>
<td>4</td>
<td>0x00863e5c</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>0x0051c1e8</td>
<td>43</td>
</tr>
<tr>
<td>6</td>
<td>0x0086cfa8</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>0x00226cf0</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>0x00144990</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0x0051f408</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0x00000000</td>
<td>0</td>
</tr>
</tbody>
</table>

EVENT HISTORY

<table>
<thead>
<tr>
<th>CurrentTime</th>
<th>ActiveQueue</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>68169518</td>
<td>FrontPanel</td>
<td>0x002294b4</td>
</tr>
<tr>
<td>68169510</td>
<td>PacketRouting</td>
<td>0x005b9f68</td>
</tr>
<tr>
<td>68169500</td>
<td>PacketRouting</td>
<td>0x005b9f68</td>
</tr>
<tr>
<td>68169498</td>
<td>IP Events</td>
<td>0x00148164</td>
</tr>
</tbody>
</table>
logout

Use the **logout** command to terminate the current session and return to the login screen.

**Syntax Description**

No subcommands.

**Default Values**

No defaults necessary for this command.

**Usage Examples**

The following example shows the logout command being executed in Enable mode:

```plaintext
ProCurve>enable
ProCurve#logout

Session now available
Press RETURN to get started.
```
reload [cancel | in <delay>]

Use the reload command to perform a manual reload of the SROS.

**Caution** Performing a reload disrupts data traffic.

### Syntax Description

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancel</td>
<td>Optional. Use the cancel keyword to deactivate a pending reload command.</td>
</tr>
<tr>
<td>in</td>
<td>Optional. Use the in keyword to specify a delay period the SROS will wait before reloading.</td>
</tr>
<tr>
<td>&lt;delay&gt;</td>
<td>Specifies the delay period in minutes (mmm) or hours and minutes (hh:mm).</td>
</tr>
</tbody>
</table>

### Default Values

No default value necessary for this command.

### Usage Examples

The following example reloads the SROS software in 3 hours and 27 minutes:

ProCurve>enable
ProCurve#reload in 03:27

The following example reloads the SROS software in 15 minutes:

ProCurve>enable
ProCurve#reload in 15

The following example terminates a pending reload command:

ProCurve>enable
ProCurve#reload cancel
show access-lists <listname>

Use the show access-lists command to display all configured access lists in the system (or a specific list).

Syntax Description

<table>
  <tr><td><listname></td><td>Optional. Specify a particular access list to display.</td></tr>
</table>

Default Values

No default value necessary for this command.

Functional Notes

The show access-lists command displays all configured access lists in the system. All entries in the access list are displayed, and a counter indicating the number of packets matching the entry is listed.

Usage Examples

The following is a sample output from the show access-lists command:

```
ProCurve>enable
ProCurve#show access-lists

Standard access list MatchAll
  permit host 10.3.50.6 (0 matches)
  permit 10.200.5.0 wildcard bits 0.0.0.255 (0 matches)
extended access list UnTrusted
  deny icmp 10.5.60.0 wildcard bits 0.0.0.255 any source-quench (0 matches)
deny tcp any (0 matches)
```
show arp [realtime]

Use the **show arp** command to display the Address Resolution Protocol (ARP) table.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>realtime</td>
<td>Displays full-screen output in real-time. See the <strong>Functional Notes</strong> below for more information.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the **realtime** argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the **#terminal length** command).

**Usage Examples**

The following is a sample output of the **show arp** command:

```
ProCurve>enable
ProCurve#show arp

ADDRESS   TTL (min)   MAC ADDRESS       LAST UPDATED (min)   INTERFACE
192.168.30.36  13   00:E0:7D:88:1A:B9   4260             eth 0/1
192.168.30.253  17   02:60:8C:DD:0A:CE   4264             eth 0/1
```
**show atm [pvc | traffic] interface atm <interface>**

Use the `show atm` command to display information specific to the ATM interface.

Variations of this command include the following:

- `show atm pvc`
- `show atm [pvc | traffic] interfaces atm <interface>`

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvc</code></td>
<td>Shows ATM PVC information.</td>
</tr>
<tr>
<td><code>traffic</code></td>
<td>Shows ATM traffic information.</td>
</tr>
<tr>
<td><code>&lt;sub-interface number&gt;</code></td>
<td>For ATM PVC information, enter the sub-interface (x.x) number.</td>
</tr>
<tr>
<td><code>&lt;atm port interface&gt;</code></td>
<td>For ATM port traffic information, enter the port ATM number 1-1023.</td>
</tr>
<tr>
<td><code>&lt;atm vcl interface&gt;</code></td>
<td>For ATM VCL traffic information, enter the ATM VCL number 1-1023.1-65536.</td>
</tr>
</tbody>
</table>

### Default Values

No default is necessary for this command.

### Usage Examples

The following is sample output from this command:

ProCurve>enable
ProCurve#show atm pvc interface atm 1.1

<table>
<thead>
<tr>
<th>Name</th>
<th>VPI</th>
<th>VCI</th>
<th>Type</th>
<th>SC</th>
<th>Peak</th>
<th>Avg/Min</th>
<th>Burst</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>atm 1.1</td>
<td>0</td>
<td>200</td>
<td>SNAP</td>
<td>N/A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Active</td>
</tr>
</tbody>
</table>
show autosynch-status

Use the show autosynch-status command to display the current autosynch configuration and the statistics for the SROS.BIZ and startup-config files (if autosynch is enabled).

Syntax Description

No subcommands.

Default Value

No default is necessary for this command.

Usage Examples

The following is a sample output from the show autosynch-status command (with AutoSynch™ disabled):

ProCurve>enable
ProCurve#show autosynch-status

AutoSynch: Mode - Disabled

AutoSynch: SROS.BIZ not synched
AutoSynch: startup-config not synched

The following is a sample output from the show autosynch-status command (with AutoSynch™ enabled):

ProCurve>enable
ProCurve#show autosynch-status

AutoSynch: Mode - Enabled

AutoSynch: SROS.BIZ synched
AutoSynch: startup-config synched
show backup interfaces

Use the show backup interfaces command to display all configured backup interfaces and the associated parameters for each.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example enters the Enable command mode and uses the show command to display backup interface information:

ProCurve>enable
ProCurve#show backup interfaces
Backup interfaces...
fr 1.16 backup interface:
   Backup state:    idle
   Backup protocol: PPP
   Call mode:        originate
   Auto-backup:      enabled
   Auto-restore:     enabled
   Priority:         50
   Backup delay:     10 seconds
   Restore delay:    10 seconds
   Connect timeout:  60 seconds

   Redial retries:  unlimited
   Redial delay:   10 seconds
   Backup enabled all day on the following days:
       Sunday Monday Tuesday Wednesday Thursday Friday Saturday

   Backup phone number list:
   Number      Call Type min/max DS0s Backup I/F
   5551212      analog  1/1      ppp 2
**show bridge** <interface> <slot/port> <bridge group #>

Use the **show bridge** command to display a list of all configured bridge groups (including individual members of each group). Enter an interface or a bridge number to display the corresponding list.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt; &lt;slot/port&gt;</td>
<td>Optional. Displays all bridge groups associated with the specific interface. Type the <strong>show bridge ?</strong> command to display a list of applicable interfaces.</td>
</tr>
<tr>
<td>&lt;bridgegroup#&gt;</td>
<td>Optional. Displays information for a specific bridge group.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output from the **show bridge** command:

```
ProCurve>enable
ProCurve#show bridge

Total of 300 station blocks 295 free

<table>
<thead>
<tr>
<th>Address</th>
<th>Action</th>
<th>Interface</th>
<th>Age</th>
<th>Rx Count</th>
<th>Tx Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:04:51:57:4D:5A</td>
<td>forward</td>
<td>eth 0/1</td>
<td>0</td>
<td>713392</td>
<td>7042770</td>
</tr>
<tr>
<td>00:04:5A:57:4F:2A</td>
<td>forward</td>
<td>eth 0/1</td>
<td>0</td>
<td>402365</td>
<td>311642</td>
</tr>
<tr>
<td>00:10:A4:B3:A2:72</td>
<td>forward</td>
<td>eth 0/1</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>00:12:79:00:8F:98</td>
<td>forward</td>
<td>eth 0/1</td>
<td>0</td>
<td>412367</td>
<td>231</td>
</tr>
<tr>
<td>00:E0:81:10:FF:CE</td>
<td>forward</td>
<td>fr 1.17</td>
<td>0</td>
<td>1502106</td>
<td>1486963</td>
</tr>
</tbody>
</table>
```
show buffers [realtime]

Use the show buffers command to display the statistics for the buffer pools on the network server.

Syntax Description

| realtime | Displays full-screen output in real-time. See the Functional Notes below for more information. |

Default Values

No default value necessary for this command.

Functional Notes

Use the realtime argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the #terminal length command).

Usage Examples

The following is a sample output from the show buffers command:

ProCurve>enable
ProCurve#show buffers
Buffer handles: 119 of 2000 used.

<table>
<thead>
<tr>
<th>Pool</th>
<th>Size</th>
<th>Total</th>
<th>Used</th>
<th>Available</th>
<th>Max. Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1800</td>
<td>1894</td>
<td>119</td>
<td>1775</td>
<td>122</td>
</tr>
<tr>
<td>1</td>
<td>2048</td>
<td>64</td>
<td>0</td>
<td>64</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4096</td>
<td>32</td>
<td>0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>8192</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>16384</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>32768</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>65536</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
show buffers users [realtime]

Use the `show buffers users` command to display a list of the top users of packet buffers. Typically, this command will only be used as a debug tool.

**Syntax Description**

| realtime | Displays full-screen output in real-time. See the Functional Notes below for more information. |

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `#terminal length` command).

**Usage Examples**

The following is a sample from the `show buffers users` command:

```
ProCurve>enable
ProCurve#show buffers users
Number of users: 7

<table>
<thead>
<tr>
<th>Rank</th>
<th>User</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x0052f4f8</td>
<td>59</td>
</tr>
<tr>
<td>2</td>
<td>0x0051a4fc</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>0x00528564</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>0x0053c1c8</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>fixedsize</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0x001d8298</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>0x0010d970</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0x00000000</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0x00000000</td>
<td>0</td>
</tr>
</tbody>
</table>
```
**show cflash <filename>**

Use the `show cflash` command to display a list of all files currently stored in compact flash memory or details about a specified file.

**Syntax Description**

| <filename> | Optional. Displays details for the specified file located in compact flash memory. Enter a wildcard (such as ".biz") to display the details for all files matching the entered pattern. |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample `show cflash` output:

```
ProCurve>enable
ProCurve#show cflash
4043024 J03_01.BIZ
285188 J03_01-boot.biz
3154 startup-config
4043024 SROS.BIZ
8374390 bytes used, 119545738 available, 127920128 total
```
show clock [detail]

Use the show clock command to display the system time and date entered using the clock set command. See clock set <time> <day> <month> <year> on page 60 for more information.

Syntax Description

detail Optional. Use this optional keyword to display more detailed clock information, including the time source.

Default Values

No default value necessary for this command.

Usage Examples

The following example displays the current time and data from the system clock:

ProCurve>show clock

23:35:07 Tue Aug 20 2002
**show configuration**

Use the `show configuration` command to display a text printout of the startup configuration file stored in flash memory.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output of the `show configuration` command:

```
ProCurve>enable
ProCurve#show configuration
!
!
no enable password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
!
ip policy-timeout tcp all-ports 600
ip policy-timeout udp all-ports 60
ip policy-timeout icmp 60
!
!
interface eth 0/1
speed auto
no ip address
shutdown
!
ip access-list standard Outbound
  permit host 10.3.50.6
  permit 10.200.5.0 0.0.0.255
```
!  
ip access-list extended Untrusted
    deny icmp 10.5.60.0 0.0.0.255 any source-quench
    deny tcp any any
!  
no ip snmp agent
!
!
line con 0
    no login
!
line telnet 0
    login
line telnet 1
    login
line telnet 2
    login
line telnet 3
    login
line telnet 4
    login
!

show connections

Use the **show connections** command to display information (including TDM group assignments) for all active connections.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output from the **show connections** command:

```
ProCurve>enable
ProCurve#show connections
Displaying all connections....

<table>
<thead>
<tr>
<th>Conn ID</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>atm 1</td>
<td>adsl 1/1</td>
</tr>
<tr>
<td>2</td>
<td>ppp 1</td>
<td>t1 2/1, tdm-group 1</td>
</tr>
<tr>
<td>3</td>
<td>ppp 1</td>
<td>t1 2/2, tdm-group 1</td>
</tr>
<tr>
<td>4</td>
<td>ppp 3</td>
<td>e1 3/1, tdm-group 1</td>
</tr>
<tr>
<td>5</td>
<td>ppp 3</td>
<td>e1 3/2, tdm-group 1</td>
</tr>
<tr>
<td>6</td>
<td>ppp 3</td>
<td>e1 3/3, tdm-group 1</td>
</tr>
</tbody>
</table>
```
show crypto ca [certificates | crls | profiles]

Use the **show crypto ca** command to display information regarding certificates and profiles.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Displays information on all certificates.</td>
</tr>
<tr>
<td>crls</td>
<td>Displays a summary of all certificate revocation lists (CRLs) for each CA.</td>
</tr>
<tr>
<td>profiles</td>
<td>Displays information on all configured CA profiles.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample from the **show crypto ca certificates** command:

```
ProCurve>enable
ProCurve#show crypto ca certificates
CA Certificate
  Status: Available
  Certificate Serial Number: 012d
  Subject Name: /C=FI/O=SSH Communications Security/OU=Web test/CN=Test CA 1
  Issuer: /C=FI/O=SSH Communications Security/OU=Web test/CN=Test CA 1
  CRL Dist. Pt: /C=FI/O=SSH Communications Security/OU=Web test/CN=Test CA 1
  Start date is Jan 9 16:25:15 2003 GMT
  End date is Dec 31 23:59:59 2003 GMT
  Key Usage:
    Non-Repudiation
    Key Encipherment
    Data Encipherment
    CRL Signature
    Encipherment Only
```
show crypto ike

Use the **show crypto ike** command to display information regarding the IKE configuration. Variations of this command include the following:

**show crypto ike client configuration pool**
**show crypto ike client configuration pool** `<poolname>`
**show crypto ike policy**
**show crypto ike policy** `<policy priority>`
**show crypto ike remote-id** `<remote-id>`
**show crypto ike sa**

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>client configuration pool</strong> <code>&lt;poolname&gt;</code></td>
<td>Displays the list of all configured IKE client configuration pools. Displays detailed information regarding the specified IKE client configuration pool.</td>
</tr>
<tr>
<td><strong>policy</strong> <code>&lt;policy priority&gt;</code></td>
<td>Displays information on all IKE policies. Indicates if client configuration is enabled for the IKE policies and displays the pool names.</td>
</tr>
<tr>
<td><strong>remote-id</strong> <code>&lt;remote-id&gt;</code></td>
<td>Displays information on all IKE information regarding the remote-id. The remote-id value is specified using the <strong>crypto ike remote-id</strong> command (refer to <strong>crypto ike remote-id</strong> on page 323).</td>
</tr>
<tr>
<td><strong>sa</strong></td>
<td>Displays the configuration of active IKE security associations.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.
Usage Examples

The following is a sample from the `show crypto ike policy` command:

ProCurve>**enable**
ProCurve#**show crypto ike policy**
Crypto IKE Policy 100
   Main mode
   Using System Local ID Address
   Peers:
       63.105.15.129
   initiate main
   respond anymode
   Attributes:
       10
       Encryption: 3DES
       Hash: SHA
       Authentication: Pre-share
       Group: 1
       Lifetime: 900 seconds
**show crypto ipsec**

Use the `show crypto ipsec` command to display information regarding the IPSec configuration. Variations of this command include the following:

- `show crypto ipsec sa`
- `show crypto ipsec sa address <ip address>`
- `show crypto ipsec sa map <mapname>`
- `show crypto ipsec transform-set`
- `show crypto ipsec transform-set <transform-set name>`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sa</code></td>
<td>Displays all IPSec security associations.</td>
</tr>
<tr>
<td><code>sa address &lt;ip address&gt;</code></td>
<td>Displays all IPSec security associations associated with the designated peer IP address.</td>
</tr>
<tr>
<td><code>sa map &lt;mapname&gt;</code></td>
<td>Displays all IPSec security associations associated with the designated crypto map name.</td>
</tr>
<tr>
<td><code>transform-set</code></td>
<td>Displays all defined transform sets.</td>
</tr>
<tr>
<td><code>&lt;transform-set name&gt;</code></td>
<td>Displays information for a specific transform set.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.
show crypto map

Use the `show crypto map` command to display information regarding crypto map settings. Variations of this command include the following:

- `show crypto map` 
- `show crypto map <interface>`
- `show crypto map <map name>`
- `show crypto map <map name> <map number>`

**Syntax Description**

- `<interface>`: Displays the crypto map settings for the specified interface. Type `show interfaces ?` for a complete list of valid interfaces.
- `<map name>`: Specifies a specific crypto map name.
- `<map number>`: Specifies a specific crypto map number.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample from the `show crypto map` command:

ProCurve> enable
ProCurve# show crypto map testMap

Crypto Map "testMap" 10 ipsec-ike
Extended IP access list NewList
Peers:
  172.27.45.57
Transform sets:
  esp-des
Security-association lifetimes:
  0 kilobytes
  86400 seconds
No PFS group configured
Interfaces using crypto map testMap:
  eth 0/1
show debugging

Use the show debugging command to display a list of all activated debug message categories.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following is a sample output from the show debugging command:

ProCurve>enable
ProCurve#show debugging

db ug a c c e s s - l i s t MatchAll
deb ug firewall
deb ug ip rip
deb ug f r a m e - r e l a y e v e n t s
deb ug fr a m e - r e l a y Llc2
deb ug frame - relay lmi
show demand

Use the show demand command to display information regarding demand routing parameters and statistics. Variations of this command include the following:

show demand
show demand interface
show demand interface <interface>
show demand resource pool
show demand resource pool <resource pool name>
show demand sessions

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>Displays the information for all demand routing interfaces.</td>
</tr>
<tr>
<td>interface &lt;interface&gt;</td>
<td>Displays information for a specific demand routing interface. Valid range: 1 to 1024. Type show demand interface ? for a list of valid interfaces.</td>
</tr>
<tr>
<td>resource pool</td>
<td>Displays all resource pool information.</td>
</tr>
<tr>
<td>resource pool &lt;resource pool name&gt;</td>
<td>Displays resource pool information for a specific resource pool name.</td>
</tr>
<tr>
<td>sessions</td>
<td>Displays active demand sessions.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following is example output from the show demand interface command:

ProCurve>enable
ProCurve#show demand int 1
Demand 1 is UP (connected)
Configuration:
  Keep-alive is set (10 sec.)
  Admin MTU = 1500
  Mode: Either, 1 dial entries, idleTime = 120, fastIdle = 20
  Resource pool demand
  No authentication configured
  IP address 10.100.0.2 255.255.255.0
Connect Sequence: Successes = 0, Failures = 0
  Seq  DialString  Technology Successes Busys NoAnswers NoAuths InUse
  5     5552222        ISDN       0       0       0       0
Current values:
  Local IP address 10.100.0.2, Peer IP address 10.100.0.1
  Seconds until disconnect: 63
  Queueing method: weighted fair
  Output queue: 0/1/428/64/0 (size/highest/max total/threshold/drops)
    Conversations: 0/1/256 (active/max active/max total)
  Available Bandwidth 48 kilobits/sec
  Bandwidth=64 Kbps
  Link through bri 1/3, Uptime 0:01:10
  IN: Octets 588, Frames 19, Errors 0
  OUT: Octets 498, Frames 18, Errors 0
  Last callerID 2565552222, last called num 5552222

The following is example output from the `show demand interface demand` command:

```
ProCurve>enable
ProCurve#show demand interface demand 1
demand 1
  Idle timer (120 secs), Fast idle timer (20 secs)

  Dialer state is data link layer up
  Dial reason: answered

  Interface bound to resource bri 1/3
  Time until disconnect 105 secs
  Current call connected 00:00:27
  Connected to 2565552222

  Number of active calls = 1
  Interesting Traffic = list junk

  Connect Sequence: Successes = 0, Failures = 0
    Seq  DialString Technology Successes Busys NoAnswers NoAuths InUse
    5    5552222   ISDN      0      0         0       0
```

The following is example output from the `show demand resource pool` command:

```
ProCurve>enable
ProCurve#show demand resource pool
Pool demand
  Resources: bri 1/3, bri 2/3
  Demand Interfaces: demand 1
```
The following is example output from the `show demand sessions` command:

```
ProCurve>enable
ProCurve#show demand sessions
Session 1
  Interface demand 1
  Local IP address = 10.100.0.2
  Remote IP address = 10.100.0.1
  Remote Username =
  Dial reason: ip (s=, d=)
  Link 1
    Dialed number = 5552222
    Resource interface = bri 1/3, Multilink not negotiated
    Connect time: 0:0:13
    Idle Timer: 119
```
show dialin interfaces

Use the `show dialin interfaces` command to display information regarding remote console dialin.

**Syntax Description**

*No subcommands.*

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output from the `show dialin interfaces` command:

```
ProCurve>enable
ProCurve#show dialin interfaces
Dialin interfaces...
modem 1/3 dialin interface:
  Connection Status: Connected
  Caller id info: name-John Smith number-5551212 time-14:23:10 2/17/2003
```
show dynamic-dns

Use the show dynamic-dns command to show information related to the dynamic DNS configuration.

Syntax Description

No subcommands.

Default Values

No default is necessary for this command.

Usage Examples

The following is sample output from this command:

ProCurve#show dynamic-dns
eth 0/1:
  Hostname: host
  Is Updated: no
  Last Registered IP: 10.15.221.33
  Last Update Time: 00:00:00 Thu Jan 01 1970
show event-history

Use the **show event-history** command to display all entries in the current local event-history log.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The event history provides useful information regarding the status of the system and individual port states. Use the event history as a troubleshooting tool when identifying system issues. The following is a sample event history log.

ProCurve>enable
ProCurve#show event-history
Using 526 bytes
2002.07.12 15:34:01 T1.t1 1/1 Yellow
2002.07.12 15:34:01 INTERFACE_STATUS.t1 1/1 changed state to down.
2002.07.12 15:34:02 T1.t1 1/1 No Alarms
2002.07.12 15:34:02 INTERFACE_STATUS.eth 0/1 changed state to up.
2002.07.12 15:34:10 OPERATING_SYSTEM Warm Start
2002.07.12 15:34:12 PPP.NEGOTIATION LCP up
2002.07.12 15:34:12 PPP.NEGOTIATION IPCP up
**show file [cflash | flash] <filename>**

Use the `show file` command to display a specified file (located in either compact flash or flash memory) to the terminal screen.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cflash</code></td>
<td>Specifies a file located in compact flash memory.</td>
</tr>
<tr>
<td><code>flash</code></td>
<td>Specifies a file located in flash memory.</td>
</tr>
<tr>
<td><code>&lt;filename&gt;</code></td>
<td>Specify the filename of the file located in the specified memory location. Wildcard entries (such as *.biz) are not valid for the <code>show file</code> command.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample `show cflash` output:

```
ProCurve>enable
ProCurve#show file cflash startup-config
Using 2354 bytes
!
!
hostname "HP_West"
enable password password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
no logging email
logging email priority-level info
!
username "admin" password "password"
!
!
no ip firewall alg h323
ip firewall alg sip
--MORE--
```
show flash <filename>

Use the show flash command to display a list of all files currently stored in flash memory or details about a specified file.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;filename&gt;</th>
<th>Optional. Displays details for a specified file located in flash memory. Enter a wildcard (such as *.biz) to display the details for all files matching the entered pattern.</th>
</tr>
</thead>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following is a sample show flash output:

ProCurve>enable
ProCurve#show flash

Files:
245669 030100boot.biz
1141553 new.biz
821 startup-config
1638 startup-config.old
1175679 SROS.biz
821 startup-config.bak

2572304 bytes used 4129776 available 6702080 total
show frame-relay

Use the `show frame-relay` command to display configuration and status parameters for configured virtual Frame Relay interfaces. Variations of this command include the following:

- `show frame-relay lmi`
- `show frame-relay pvc`
- `show frame-relay pvc interface frame-relay <interface>`
- `show frame-relay pvc interface frame-relay <interface> realtime`
- `show frame-relay pvc realtime`

**Syntax Description**

- **lmi**: Displays Link Management Interface (LMI) statistics for each virtual Frame Relay interface.
- **pvc**: Displays Permanent Virtual Circuit (PVC) configuration and statistics for all virtual Frame Relay interfaces (or a specified interface).
- **interface**: Displays configuration and statistics for a specified Frame Relay interface.
- **frame-relay**: Optional. Displays Frame Relay PVC statistics for a specific Frame Relay interface.
- **<interface>**: Specifies the virtual Frame Relay interface (for example fr 1).
- **realtime**: Displays full-screen output in realtime. See the `Functional Notes` section below for more information.

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `#terminal length` command).
Usage Examples
The following are sample outputs from various show frame-relay commands:

ProCurve>enable
ProCurve#show frame-relay lmi

LMI statistics for interface FR 1 LMI TYPE = ANSI
Num Status Enq. Sent 79 Num Status Msgs Rcvd 71
Num Update Status Rcvd 12 Num Status Timeouts 5

ProCurve>enable
ProCurve#show frame-relay pvc

Frame Relay Virtual Circuit Statistics for interface FR 1

<table>
<thead>
<tr>
<th></th>
<th>Active</th>
<th>Inactive</th>
<th>Deleted</th>
<th>Static</th>
</tr>
</thead>
<tbody>
<tr>
<td>local</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

DLCI = 16 DLCI USAGE = LOCAL PVC STATUS = ACTIVE INTERFACE = FR 1.16
MTU: 1500
input pkts: 355 output pkts: 529 in bytes: 23013
out bytes: 115399 dropped pkts: 13 in FECN pkts: 0
in BECN pkts: 0 in DE pkts: 0 out DE pkts: 0
pvc create time: 00:00:00:12 last time pvc status changed: 00:00:13:18

DLCI = 20 DLCI USAGE = LOCAL PVC STATUS = ACTIVE INTERFACE = FR 1.20
MTU: 1500
input pkts: 0 output pkts: 44 in bytes: 0
out bytes: 22384 dropped pkts: 11 in FECN pkts: 0
in BECN pkts: 0 in DE pkts: 0 out DE pkts: 0
pvc create time: 00:00:01:25 last time pvc status changed: 00:00:13:18
show frame-relay fragment [interface frame-relay <port.sublink>]

Use the show frame-relay fragment command to display FRF.12 statistics for Frame Relay sublinks enabling FRF.12 fragmentation.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface frame-relay &lt;port.sublink&gt;</td>
<td>Optional. Displays detailed FRF.12 statistics for the specified Frame Relay sublink (if FRF.12 is enabled on that sublink).</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following are sample outputs from various show frame-relay fragment commands:

ProCurve>enable
ProCurve#show frame-relay fragment
interface        dlcI        frag_size      rx_frag   tx_frag   dropped_frag
fr 1.1            17          100               46        48        0
fr 1.2            18          200               42        21        0

ProCurve>enable
ProCurve#show frame-relay fragment frame-relay 1.1
DLCI = 17 FRAGMENT SIZE = 100
rx frag. pkts     46          tx frag. pkts   48
rx frag. bytes    4598        tx frag. bytes  4724
rx non-frag. pkts 18          tx non-frag. pkts 28
rx non-frag. bytes 1228       tx non-frag. bytes 1960
rx assembled pkts 23          tx pre-fragment pkts 34
rx assembled bytes 5478       tx pre-fragment bytes 6324
dropped reassembling pkts 0    dropped fragmenting pkts 0
rx out-of-sequence fragments 0
rx unexpected beginning fragment 0
**show frame-relay multilink** `<interface>` **detailed**

Use the `show frame-relay multilink` command to display information associated with the Frame Relay multilink interface.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;interface&gt;</code></th>
<th>Optional. Specifies the display of information for a specific interface. Enter the <code>show frame-relay multilink ?</code> command for a complete list of interfaces.</th>
</tr>
</thead>
<tbody>
<tr>
<td>detailed</td>
<td>Optional. Use this optional keyword to display more detailed information.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output from this command:

ProCurve>`enable`
ProCurve>`show frame-relay multilink`
Bundle: frame-relay 1 is DOWN; class A bundle
Near-end BID: MFR1; Far-end BID: unknown
**show hosts [verbose]**

Use the `show hosts` command to display information such as the domain name, name lookup service, a list of name server hosts, and the cached list of host names and addresses on the network to which you can connect.

**Syntax Description**

| verbose | Enables detailed messaging. |

**Default Values**

No default value necessary for this command.

**Functional Notes**

The list below describes the fields contained in the host table:

- **Flags**: Indicate whether the entry is permanent (P) or temporary (T) and if the entry is OK or expired (EXP).
- **Age**: Indicates the age of the entry.
- **Type**: Shows the protocol type.
- **Address**: Displays the IP address for the entry.

**Usage Examples**

The following example is sample output from the `show hosts` command:

```
ProCurve>enable
ProCurve#show hosts

Name/address lookup uses domain name service
DNS Proxy is disabled
Default domain is not set
Name servers are 1.1.1.1 2.2.2.2

<table>
<thead>
<tr>
<th>Host</th>
<th>Flags</th>
<th>Age</th>
<th>Type</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example1</td>
<td>(P OK)</td>
<td>-</td>
<td>IP</td>
<td>1.1.1.1</td>
</tr>
<tr>
<td>Example2</td>
<td>(P OK)</td>
<td>-</td>
<td>IP</td>
<td>2.2.2.2</td>
</tr>
</tbody>
</table>
```
show interfaces <interface>

Use the `show interfaces` command to display configuration parameters and current statistics for all interfaces (or a specified interface). Variations of this command include the following:

- `show interfaces <interface> description`
- `show interfaces <interface> performance-statistics`
- `show interfaces <interface> performance-statistics <x-y>`
- `show interfaces <interface> performance-statistics total-24-hour`
- `show interfaces <interface> realtime`
- `show interfaces <interface> status`
- `show interfaces <interface> verbose`
- `show interfaces <interface> version`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;interface&gt;</code></td>
<td>Optional. Specifies the interface to display. Type <code>show interfaces ?</code> for a complete list of valid interfaces.</td>
</tr>
<tr>
<td>description</td>
<td>Optional. Displays information such as name, administrative status, protocol, and description for all the interfaces.</td>
</tr>
<tr>
<td>performance-statistics</td>
<td>Optional. Displays the current 15-minute interval, the current 24-hour totals, and all 96 stored intervals.</td>
</tr>
<tr>
<td>performance-statistics &lt;x-y&gt;</td>
<td>Shows the current 15-minute interval, the current 24-hour totals, and all intervals from x through y. This command is basically the same thing as the <code>performance-statistics</code> command with the added function of allowing you to specify a particular interval (or range of intervals) to display rather than displaying all 96.</td>
</tr>
<tr>
<td>performance-statistics total-24-hour</td>
<td>Optional. Displays the current 24-hour totals and the past seven 24-hour intervals.</td>
</tr>
</tbody>
</table>

**Note**

If you want to display the 24th interval, enter (for example) `show interface t11/1 performance-statistics 24-24`. Entering `show interface t1 1/1 performance-statistics 24` results in displaying the 24-hour statistics. Any number other than 24 (between 1 and 96) results in the correct display of the selected interval (e.g., `show interface t1 1/1 performance-statistics 4` shows the fourth interval).

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>realtime</td>
<td>Displays full-screen output in real-time. Refer to the Functional Notes below for more information.</td>
</tr>
<tr>
<td>status</td>
<td>Optional. Displays information such as name, type, status, VLAN, speed, and duplex for all the Ethernet interfaces only.</td>
</tr>
<tr>
<td>verbose</td>
<td>Displays detailed configuration information on the terminal screen (versus only the non-default values).</td>
</tr>
<tr>
<td>version</td>
<td>Optional. Displays current version information (e.g., model and list number, software version, etc.) for the T1 interface.</td>
</tr>
</tbody>
</table>
Default Values

No default value necessary for this command.

Functional Notes

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `terminal length` command; refer to `terminal length <lines>` on page 280).

Usage Examples

The following are samples from various `show interfaces` commands:

ProCurve>enable
ProCurve#show interfaces t1 1/1

t1 1/1 is UP
  T1 coding is B8ZS framing is ESF
  Clock source is line FDL type is ANSI
  Line build-out is 0dB
  No remote loopbacks No network loopbacks

  DS0 Status: 123456789012345678901234
                NNNNNNNNNNNNNNNNNNNNNNNNN

  Line Status: -- No Alarms --

  Current Performance Statistics:
    0 Errored Seconds 0 Bursty Errored Seconds
    0 Severely Errored Seconds 0 Severely Errored Frame Seconds
    0 Unavailable Seconds 0 Path Code Violations
    0 Line Code Violations 0 Controlled Slip Seconds
    0 Line Errored Seconds 0 Degraded Minutes

ProCurve#show interfaces modem 1/2
modem 1/2 is UP
  Line status: on-hook
  Caller ID will be used to route incoming calls
    0 packets input 0 bytes 0 no buffer
    0 runts 0 giants 0 throttles
    0 input errors 0 CRC 0 frame
    0 abort 0 ignored 0 overruns
    0 packets output 0 bytes 0 underruns
0 input clock glitches 0 output clock glitches
0 carrier lost 0 cts lost

**ProCurve**#show interfaces eth 0/1
Ip address is 10.200.1.50
Netmask is 255.255.0.0
MTU is 1500
Fastcaching is Enabled
RIP Authentication is Disabled
RIP Tx uses global version value
RIP Rx uses global version value

**ProCurve**#show interfaces fr 1
TDM group 10 line protocol is UP
Encapsulation FRAME-RELAY (fr 1)
463 packets input 25488 bytes 0 no buffer
0 runts 0 giants 0 throttles
0 input errors 0 CRC 0 frame
0 abort 0 ignored 0 overruns
864 packets output 239993 bytes 0 underruns
0 input clock glitches 0 output clock glitches
0 carrier lost 0 cts lost

Line Status: -- No Alarms --

Current Performance Statistics:
0 Errored Seconds 0 Bursty Errored Seconds
0 Severely Errored Seconds 0 Severely Errored Frame Seconds
0 Unavailable Seconds 0 Path Code Violations
0 Line Code Violations 0 Controlled Slip Seconds
0 Line Errored Seconds 0 Degraded Minutes

**ProCurve**#show interfaces fr 1.100*
fr 1.100 is Active
Ip address is 63.97.45.57, mask is 255.255.255.248
Interface-dlci is 100
MTU is 1500 bytes, BW is 96000 Kbit (limited)
Average utilization is 53%

*Note: If the user has configured a **Bc** and **Be** value on the virtual circuit, the bandwidth (**BW**) displayed is the sum of those values (**Bc + Be**). If not, the value for **BW** is the speed of the interface. The **Average utilization** displayed is the average utilization of the displayed bandwidth. If the bandwidth number is the **Bc + Be** value, the (**limited**) text appears (as shown above).
show interfaces adsl <slot/port> information [atuc | atur | bit-allocation]

Use the `show interfaces adsl` command to display information related to the ADSL port. Variations of this command include the following:

- `show interfaces adsl <slot/port>`
- `show interfaces adsl <slot/port> information`
- `show interfaces adsl <slot/port> information atuc`
- `show interfaces adsl <slot/port> information atur`
- `show interfaces adsl <slot/port> information bit-allocation`
- `show interfaces adsl <slot/port> performance-statistics`
- `show interfaces adsl <slot/port> performance-statistics <x-y>`
- `show interfaces adsl <slot/port> performance-statistics total-24-hour`
- `show interfaces adsl <slot/port> version`

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;slot/port&gt;</code></th>
<th>Specifies interface slot and port number.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>information</code></td>
<td>Optional. Specifies all available information for the ADSL configuration (atuc, atur, and bit-allocation).</td>
</tr>
<tr>
<td><code>information atuc</code></td>
<td>Optional. Shows ADSL interface remote information.</td>
</tr>
<tr>
<td><code>information atur</code></td>
<td>Optional. Shows ADSL local information.</td>
</tr>
<tr>
<td><code>information bit-allocation</code></td>
<td>Optional. Shows ADSL DMT bit-allocation table.</td>
</tr>
<tr>
<td><code>performance-statistics</code></td>
<td>Optional. Displays the current 15-minute interval, the current 24-hour totals, and all 96 stored intervals.</td>
</tr>
<tr>
<td><code>performance-statistics &lt;x-y&gt;</code></td>
<td>Optional. Shows the current 15-minute interval, the current 24-hour totals, and all intervals from x through y. This command is basically the same thing as the <code>performance-statistics</code> command with the added function of allowing you to specify a particular interval (or range of intervals) to display rather than displaying all 96. <code>performance-statistics total-24-hour</code> Optional. Displays the current 24-hour totals and the past seven 24-hour intervals.</td>
</tr>
</tbody>
</table>

**Note**

Note: If you want to display the 24th interval, enter (for example) `show interface t1 1/1 performance-statistics 24-24`. Entering `show interface t1 1/1 performance-statistics 24` results in displaying the 24-hour statistics. Any number other than 24 (between 1 and 96) results in the correct display of the selected interval (e.g., `show interface t1 1/1 performance-statistics 4` shows the fourth interval).

| `version` | Optional. Displays current version information (e.g., model and list number, software version, etc.) for the T1 interface. |

**Default Values**

No default is necessary for this command.
Usage Examples

The following example shows sample output for this command:

ProCurve# **show interfaces adsl 1/1 information**
adsl 1/1 line information
adsl 1/1 Local Line Information
  Vendor Id: 
  Serial Number: 
  Firmware Version: 
  ADSL Capabilities  G.DMT, G.LITE, ADSL2, ADSL2+

adsl 1/1 Remote Line Information
  Vendor Id: 00000000
  Serial Number: 00000000
  Firmware Version: 0
  ADSL Capabilities  G.DMT, G.LITE, ADSL2, ADSL2+
**show ip access-lists** <listname>

Use the `show ip access-lists` command to display all configured IP access lists in the system.

**Syntax Description**

| `<listname>` | Optional. Specify a particular access list to display. |

**Default Values**

No default value necessary for this command.

**Functional Notes**

The `show ip access-lists` command displays all configured IP access lists in the system. All entries in the access list are displayed, and a counter indicating the number of packets matching the entry is listed.

**Usage Examples**

The following is a sample output from the `show ip access-lists` command:

```
ProCurve>enable
ProCurve#show ip access-lists

Standard IP access list MatchAll
  permit host 10.3.50.6 (0 matches)
  permit 10.200.5.0 wildcard bits 0.0.0.255 (0 matches)
Extended IP access list UnTrusted
  deny   icmp 10.5.60.0 wildcard bits 0.0.0.255 any source-quench (0 matches)
  deny   tcp any any (0 matches)
```
show ip arp [realtime]

Use the `show ip arp` command to display the Address Resolution Protocol (ARP) table.

**Syntax Description**

| **realtime** | Displays full-screen output in real-time. See the `Functional Notes` below for more information. |

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing `Ctrl-C`). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `terminal length` command; refer to `terminal length <lines>` on page 280).

**Usage Examples**

The following is a sample output of the `show ip arp` command:

```
ProCurve>enable
ProCurve#show ip arp

ADDRESS       TTL (min)    MAC ADDRESS     LAST UPDATED (min)
192.168.30.36  13           00:E0:7D:88:1A:B9  4260
192.168.30.253 17           02:60:8C:DD:0A:CE  4264
```
show ip as-path-list [<listname>]

Use the show ip as-path-list command to display any AS path lists that have been configured in the router, along with any permit and deny clauses in each list.

Syntax Description

| <listname> | Optional. Specifies that the command display only the list matching the specified AS path listname. If not specified, all AS path lists are displayed. |

Default Values

By default, this command displays all AS path lists.

Usage Examples

In the following example, all AS path lists defined in the router are displayed.

ProCurve>enable
ProCurve#show ip as-path-list
ip as-path-list AsPathList1:
  permit 100
  permit 200
  permit 300
  deny 6500
ip as-path-list AsPathList2:
  permit 400
  permit 500

In the following example, only the AS path list with the name AsPathList2 is displayed.

ProCurve>enable
ProCurve#show ip as-path-list AsPathList2
ip as-path-list AsPathList2:
  permit 400
  permit 500
**show ip bgp community**

Use the `show ip bgp community` command to display only those routes learned via BGP that match the community numbers specified in the command. If no communities are specified, all BGP routes are shown.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;community-number&gt;</code></td>
<td>Optional. Displays routes that contain this value in their community attribute. This is a numeric value that can be an integer from 1 to 4,294,967,295 or string in the form &quot;aa:nn&quot;, where the value of &quot;aa&quot; is the AS number and the value of &quot;nn&quot; is the community number. Multiple community-number parameters can be present in the command.</td>
</tr>
<tr>
<td><code>internet</code></td>
<td>Optional. Displays routes that contain this value in their community attribute. This represents the well-known reserved community number for the INTERNET community.</td>
</tr>
<tr>
<td><code>local-as</code></td>
<td>Optional. Displays routes that contain this value in their community attribute. This represents the well-known reserved community number for NO_EXPORT_SUBCONFED. Routes containing this attribute should not be advertised to external BGP peers.</td>
</tr>
<tr>
<td><code>no-export</code></td>
<td>Optional. Displays routes containing this value in the community attribute. This represents the well-known reserved community number for NO_EXPORT. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.</td>
</tr>
<tr>
<td><code>no-advertise</code></td>
<td>Optional. Displays routes containing this value in the community attribute. This represents the well-known reserved community number for NO_ADVERTISE. Routes containing this attribute should not be advertised to any BGP peer.</td>
</tr>
<tr>
<td><code>exact</code></td>
<td>Optional. Displays BGP routes with the community numbers specified and only those specified.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command displays all BGP routes.
Usage Examples

In the following example, all BGP routes are displayed whose community numbers match those listed in the show ip bgp community command.

ProCurve>enable
ProCurve#show ip bgp community local-as 10:405
BGP local router ID is 10.22.131.241, local AS is 302.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.22.152.20/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 i</td>
</tr>
<tr>
<td>10.22.152.24/29</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 5 i</td>
</tr>
<tr>
<td>10.22.152.36/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 i</td>
</tr>
<tr>
<td>10.22.152.52/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 i</td>
</tr>
<tr>
<td>11.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 6 i</td>
</tr>
<tr>
<td>12.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 6 i</td>
</tr>
<tr>
<td>13.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 6 i</td>
</tr>
<tr>
<td>14.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 6 i</td>
</tr>
<tr>
<td>20.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 5 i</td>
</tr>
<tr>
<td>21.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 1 3 4 5 i</td>
</tr>
</tbody>
</table>

Total RIB entries = 10

Information displayed includes: the ID of this router and its Autonomous System (AS) number; the destination Network address of the route learned; the Next Hop address to that network; the Metric; the Local Preference value (set using the set local-preference command); and the AS Path to the destination network.

The following is a sample output for the show-ip bgp community command with an exact match specified: BGP routes with the community numbers specified and only those specified are shown

ProCurve>enable
ProCurve#show ip bgp community 1001 2001 3001 exact
BGP local router ID is 192.168.9.1, local AS is 252.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>NextHop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 192.168.11.0/24</td>
<td>10.22.27.251</td>
<td></td>
<td></td>
<td>249 251 i</td>
</tr>
<tr>
<td>* 192.168.12.0/24</td>
<td>10.22.27.251</td>
<td></td>
<td></td>
<td>249 251 i</td>
</tr>
<tr>
<td>* 192.168.32.0/24</td>
<td>10.22.27.249</td>
<td></td>
<td></td>
<td>249 i</td>
</tr>
<tr>
<td>* 192.168.33.0/24</td>
<td>10.22.27.249</td>
<td></td>
<td></td>
<td>249 i</td>
</tr>
</tbody>
</table>

Total RIB entries = 4
**show ip bgp community-list** `<community-list-name>` [exact]

Use the **show ip bgp community-list** command to display BGP routes that are permitted by the specified community list.

**Syntax Description**

- `<community-list-name>`: Specifies the name of the community list whose routes you wish to see.
- **exact**: Optional. Specifying this option restricts the routes displayed to only those whose community lists exactly match those specified in the named community list. If this parameter is omitted, all routes matching any part of the specified community list will be displayed.

**Default Values**

No default value necessary for this command.

**Functional Notes**

Information displayed includes the ID of this router and its Autonomous System (AS) number, the destination Network address of the route learned, the Next Hop address to that network, the Metric, the Local Preference (LocPrf) value (set using the **set local-preference** command), and the AS Path to the destination network.

**Usage Examples**

In the following example, all BGP routes are displayed whose community numbers match those defined in the community list named CList1.

```
ProCurve>enable
ProCurve#show ip bgp community-list CList1
BGP local router ID is 10.22.131.241, local AS is 302.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.22.152.20/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 4 i</td>
</tr>
<tr>
<td>10.22.152.24/29</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 5 i</td>
</tr>
<tr>
<td>10.22.152.36/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 i</td>
</tr>
<tr>
<td>10.22.152.52/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 i</td>
</tr>
<tr>
<td>11.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 6 i</td>
</tr>
<tr>
<td>12.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 6 i</td>
</tr>
<tr>
<td>13.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 6 i</td>
</tr>
<tr>
<td>14.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 6 i</td>
</tr>
<tr>
<td>20.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 5 i</td>
</tr>
<tr>
<td>21.0.0.0/30</td>
<td>10.22.131.10</td>
<td>304</td>
<td></td>
<td>302 300 3 4 5 i</td>
</tr>
</tbody>
</table>

Total RIB entries = 10
```
show ip bgp neighbors

Use the `show ip bgp neighbors` command to display information for the specified neighbor. Variations of this command include the following:

```
show ip bgp neighbors
show ip bgp neighbors <ip address>
show ip bgp neighbors <ip address> advertised-routes
show ip bgp neighbors <ip address> received-routes
show ip bgp neighbors <ip address> routes
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Displays information for the specified neighbor. If no IP address is entered, information for all neighbors is displayed.</td>
</tr>
<tr>
<td>advertised-routes</td>
<td>Displays all routes being advertised to the specified neighbor. Command output is the same as for <code>show ip bgp</code> except filtered to only the BGP routes being advertised to the specified neighbor.</td>
</tr>
<tr>
<td>received-routes</td>
<td>Displays all routes (accepted and rejected) advertised by the specified neighbor. Routes may be rejected by inbound filters such as prefix list filters.</td>
</tr>
<tr>
<td>routes</td>
<td>Displays all accepted received routes advertised by the specified neighbor. Routes displayed have passed inbound filtering. This command output is the same as <code>show ip bgp</code> except the output is filtered to those learned from the specified neighbor.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Functional Notes**

Entries that are not filtered by prefix lists are marked with an asterisk (*) to show they are valid. Entries that are deemed the best path to advertised route are marked with a caret (>).
Usage Examples

The following are output variations of the `show ip bgp neighbors` command:

ProCurve>`enable
ProCurve>`show ip bgp neighbors
BGP neighbor is 10.15.43.17, remote AS 100, external link
Configured hold time is 180, keepalive interval is 60 seconds
Default minimum time between advertisement runs is 30 seconds
Connections established 6; dropped 5
Last reset: Interface went down
  Connection ID: 15
    BGP version 4, remote router ID 8.1.1.1
    BGP state is Established, for 01:55:05
    Negotiated hold time is 180, keepalive interval is 60 seconds
    Message statistics:
      InQ depth is 0, OutQ depth is 0
        Local host: 10.15.43.18, Local port: 179
        Sent    Rcvd
          Opens: 1    1
          Notifications: 0  0
          Updates: 0  8
          Keepalives: 116 116
          Unknown: 0 0
          Total: 117 125
Foreign host: 10.15.43.17, foreign port: 1048
  Flags: passive open

ProCurve>`show ip bgp neighbors 10.15.43.34 advertised-routes
BGP local router ID is 10.0.0.1, local AS is 101.
  Status codes: * valid, > best, i - internal
  Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>NextHop</th>
<th>Metric Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 1.0.0.0/8</td>
<td>10.15.43.17</td>
<td>1 100 i</td>
</tr>
<tr>
<td>*&gt; 2.0.0.0/9</td>
<td>10.15.43.17</td>
<td>1 100 i</td>
</tr>
</tbody>
</table>
ProCurve#**show ip bgp neighbors 10.15.43.17 received-routes**
BGP local router ID is 10.0.0.1, local AS is 101.
Status codes: * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>NextHop</th>
<th>Metric</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 1.0.0.0/8</td>
<td>10.15.43.17</td>
<td>1</td>
<td>100 i</td>
</tr>
<tr>
<td>*&gt; 2.0.0.0/9</td>
<td>10.15.43.17</td>
<td>1</td>
<td>100 i</td>
</tr>
</tbody>
</table>

ProCurve#**show ip bgp neighbors 10.15.43.17 routes**
BGP local router ID is 10.0.0.1, local AS is 101.
Status codes: * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>NextHop</th>
<th>Metric</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 1.0.0.0/8</td>
<td>10.15.43.17</td>
<td>1</td>
<td>100 i</td>
</tr>
<tr>
<td>*&gt; 2.0.0.0/9</td>
<td>10.15.43.17</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>
show ip bgp regexp <expression>

Use the `show ip bgp regexp` command to display a summary of the BGP route table that includes routes who autonomous system (AS) path matches the specified expression.

**Syntax Description**

| `<expression>` | Displays routes whose autonomous system (AS) path matches the regular expression specified. |

**Default Values**

No default value necessary for this command.

**Functional Notes**

Entries that are not filtered by prefix lists are marked with an asterisk (*) to show they are valid. Entries that are deemed the best path to advertised route are marked with a caret (>).

**Usage Examples**

The following sample output of the `show ip bgp regexp _303_` command shows all of the entries in the BGP database that contain "303" in the AS path.

```
ProCurve#show ip bgp regexp _303_
BGP local router ID is 192.168.3.1, local AS is 304.
Status codes: * valid, > best, i - internal, o - local
Origin codes: i - IGP, e - EGP, ? - incomplete

Network   NextHop      Metric  LocPrf  Path
10.22.130.8/29 10.22.132.9 303 304 302 i
* i10.22.130.240/28 0.22.132.1 100 303 300 i
* 10.22.130.240/28 10.22.132.9 303 300 i
10.22.131.0/29 10.22.132.9 303 304 302 i
* 10.22.131.8/29 10.22.132.9 303 304 302 i
* i10.22.131.16/29 10.22.132.1 0 100 303 i
* 10.22.131.16/29 10.22.132.9 0 303 i
* i10.22.131.240/28 10.22.132.1 100 303 300 i
* 10.22.131.240/28 10.22.132.9 303 300 i
* 10.22.132.0/29 10.22.131.1 0 302 303 i
* 10.22.132.0/29 10.22.131.9 0 302 303 i
* i10.22.132.0/29 10.22.132.1 0 100 303 i
*> 10.22.132.0/29 10.22.132.9 0 303 i
* 10.22.132.8/29 10.22.131.1 0 302 303 i
* 10.22.132.8/29 10.22.131.9 0 302 303 i
```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>* 10.22.132.8/29</td>
<td>10.22.132.9</td>
<td>0</td>
<td>303 i</td>
</tr>
<tr>
<td>* i10.22.132.240/28</td>
<td>10.22.132.1</td>
<td>0</td>
<td>100 303 i</td>
</tr>
<tr>
<td>*&gt; 10.22.132.240/28</td>
<td>10.22.132.9</td>
<td>0</td>
<td>303 i</td>
</tr>
<tr>
<td>10.22.134.0/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.8/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.16/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.24/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.32/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.40/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.48/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.56/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.64/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.134.80/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 i</td>
</tr>
<tr>
<td>10.22.135.0/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 305 i</td>
</tr>
<tr>
<td>10.22.135.8/29</td>
<td>10.22.132.9</td>
<td></td>
<td>303 304 305 i</td>
</tr>
</tbody>
</table>

Total RIB entries = 30
**show ip bgp summary**

Use the `show ip bgp summary` command to display a summary of the BGP route table.

**Syntax Description**

| `summary` | Displays a summary of the status for all BGP. |

**Default Values**

No default value necessary for this command.

**Functional Notes**

Entries that are not filtered by prefix lists are marked with an asterisk (*) to show they are valid. Entries that are deemed the best path to advertised route are marked with a caret (^).

**Usage Examples**

The following sample output of the `show ip bgp summary` command shows a summarized list of the configured BGP neighbors as well as their status and statistics.

```
ProCurve#show ip bgp summary
BGP router identifier 192.168.3.1, local AS number 304
8 network entries, 5 paths, and 23 BGP path attribute entries

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>V</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>State/PfxRcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.22.131.1</td>
<td>4</td>
<td>302</td>
<td>95</td>
<td>104</td>
<td>0</td>
<td>0</td>
<td>01:30:06</td>
<td>9</td>
</tr>
<tr>
<td>10.22.131.9</td>
<td>4</td>
<td>302</td>
<td>97</td>
<td>105</td>
<td>0</td>
<td>0</td>
<td>01:30:07</td>
<td>21</td>
</tr>
<tr>
<td>10.22.132.9</td>
<td>4</td>
<td>303</td>
<td>200</td>
<td>179</td>
<td>0</td>
<td>0</td>
<td>02:43:09</td>
<td>21</td>
</tr>
<tr>
<td>10.22.134.1</td>
<td>4</td>
<td>304</td>
<td>166</td>
<td>178</td>
<td>0</td>
<td>0</td>
<td>02:43:15</td>
<td>3</td>
</tr>
<tr>
<td>10.22.134.10</td>
<td>4</td>
<td>304</td>
<td>174</td>
<td>179</td>
<td>0</td>
<td>0</td>
<td>02:43:24</td>
<td>7</td>
</tr>
<tr>
<td>10.22.134.26</td>
<td>4</td>
<td>304</td>
<td>172</td>
<td>174</td>
<td>0</td>
<td>0</td>
<td>02:41:43</td>
<td>10</td>
</tr>
<tr>
<td>10.22.134.34</td>
<td>4</td>
<td>304</td>
<td>164</td>
<td>174</td>
<td>0</td>
<td>0</td>
<td>02:41:40</td>
<td>4</td>
</tr>
</tbody>
</table>
```
**show ip cache**

Use the `show ip cache` command to display the fast cache table.

**Syntax Description**

No subcommands.

**Default Values**

No default necessary for this command.

**Usage Examples**

The following example shows sample output from the `show ip cache` command:

```
ProCurve>enable
ProCurve#show ip cache

<table>
<thead>
<tr>
<th>DESTINATION</th>
<th>INTERFACE</th>
<th>NEXT HOP</th>
<th>USE COUNT</th>
<th>MAC ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.17.6.52</td>
<td>Loopback</td>
<td>172.20.0.1</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td>172.22.77.80</td>
<td>eth 0/1</td>
<td>10.17.254.254</td>
<td>0</td>
<td>00:12:79:11:BA:32</td>
</tr>
<tr>
<td>10.17.255.255</td>
<td>Loopback</td>
<td>172.20.0.1</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>
```
**show ip community-list [community-list-name]**

Use the `show ip community-list` command to display any or all defined community lists in the router configuration.

**Syntax Description**

<community-list-name> Optional. Specifies the name of the community list you wish to display. If this parameter is omitted, all defined community lists will be displayed.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example shows two community lists, one of which permits all routes containing community number 10:67, and another which permits routes containing community number 10:68 and the internet community number, but denies routes containing community number 10:45.

ProCurve#show ip community-list
ip community-list CommList1:
   permit 10:67
ip community-list CommList2:
   permit 10:68 internet
deny 10:45
show ip dhcp-client lease <interface>

Use the `show ip dhcp-client lease` command to display all Dynamic Host Client Protocol (DHCP) lease information for interfaces that have dynamically assigned IP addresses.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;interface&gt;</th>
<th>Optional. Displays the information for the specified interface. Type <code>show ip dhcp-client lease ?</code> for a complete list of applicable interfaces.</th>
</tr>
</thead>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output from the `show dhcp-client lease` command:

```
ProCurve>enable
ProCurve#show dhcp-client lease

Interface: ethernet 0/1  
Temp IP address: 10.100.23.64 Mask: 255.255.255.0  
   DHCP Lease server: 10.100.23.207 State: Bound (3)  
   Lease: 120 seconds  
Temp default gateway address: 10.100.23.1  
   Client-ID: N/A
```
show ip dhcp-server binding <client ip address>

Use the show ip dhcp-server binding command to display the Dynamic Host Client Protocol (DHCP) server client table with associated information.

**Syntax Description**

<client ip address> Optional. Specifies a particular client IP address.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output from the show ip dhcp-server binding command:

ProCurve>enable
ProCurve#show ip dhcp-server binding

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Client Id</th>
<th>Lease Expiration</th>
<th>Client Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.100.23.64</td>
<td>01:00:12:79:00:8f:b3</td>
<td>Aug 15 2002 11:02 AM</td>
<td>Router</td>
</tr>
</tbody>
</table>
show ip igmp groups <group-address>

Use the show ip igmp groups command to display the multicast groups that have been registered by directly connected receivers using IGMP. If no group address is specified, all groups are shown with this command.

Syntax Description

<group-address> Optional. Displays IP address of a multicast group.

Default Values

No default value necessary for this command.

Usage Examples

The following is sample output from this command:

ProCurve>enable
ProCurve#show ip igmp groups

IGMP Connected Group Membership

<table>
<thead>
<tr>
<th>Group Address</th>
<th>Interface</th>
<th>Uptime</th>
<th>Expires</th>
<th>Last Reporter</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.0.1.50</td>
<td>Loopback100</td>
<td>00:42:57</td>
<td>00:02:50</td>
<td>172.23.23.1</td>
</tr>
<tr>
<td>172.1.1.1</td>
<td>Ethernet0/1</td>
<td>00:05:26</td>
<td>00:02:51</td>
<td>1.1.1.2</td>
</tr>
<tr>
<td>172.1.1.1</td>
<td>Loopback100</td>
<td>00:42:57</td>
<td>00:02:51</td>
<td>172.23.23.1</td>
</tr>
</tbody>
</table>
show ip igmp interface <interface>

Use the `show ip igmp interface` command to display multicast-related information per-interface. If no interface is specified, this command shows information for all interfaces.

**Syntax Description**

| <interface> | Optional. Designates the display of information for a specific interface (in the format `type slot/port`). Enter the `show ip igmp interface ?` command for a complete list of interfaces. |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output from this command:

```
ProCurve>enable
ProCurve#show ip igmp interface
eth 0/1 is UP
   Ip Address is 10.22.120.47, netmask is 255.255.255.0
IGMP is enabled on interface
   Current IGMP version is 2
   IGMP query interval is 60 seconds
   IGMP querier timeout is 120 seconds
   IGMP max query response time is 10 seconds
   Last member query count is 2
   Last member query response interval is 1000 ms
IGMP activity: 548 joins, 0 leaves
   IGMP querying router is 0.0.0.0
   IGMP helper address is disabled
```
show ip interfaces [<interface> | brief]

Use the `show ip interfaces` command to display the status information for all IP interfaces (or a specific interface).

**Note**  To view secondary IP addresses, use the `show running-config` command.

**Syntax Description**
- `<interface>` Optional. Displays status information for a specific interface. If no interface is entered, status information for all interfaces is displayed. Type `show ip interfaces ?` for a complete list of applicable interfaces.
- `brief` Use this optional keyword to display an abbreviated version of interface statistics for all IP interfaces.

**Default Values**
No default value necessary for this command.

**Usage Examples**
The following is a sample output of the `show ip interfaces` command:

```
ProCurve>enable
ProCurve#show ip interfaces

eth 0/1 is UP, line protocol is UP
    Ip address is 10.10.10.1
    Netmask is 255.255.255.0
    MTU is 1500
    Fastcaching is Enabled
    RIP Authentication is Disabled
    RIP Tx uses global version value
    RIP Rx uses global version value
```
**show ip local policy**

Use the `show ip local policy` command to display information about the route-map used for local policy-based routing.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output from this command:

ProCurve>enable
ProCurve#show ip local policy

Local policy routing is enabled, using route-map equal
route-map equal, permit, sequence 10
Match clauses:
  length 150 200
Set clauses:
  ip next-hop 10.10.11.254
Policy routing matches: 0 packets, 0 bytes
route-map equal, permit, sequence 20
Match clauses:
  ip address (access-lists): 101
Set clauses:
  ip next-hop 10.10.11.14
Policy routing matches: 2 packets, 172 bytes
**show ip mroute [**<group-address>** | **<interface>**] [**summary** | **all**]**

Use the `show ip mroute` command to display IP multicasting routing table information.

**Syntax Description**

- `<group-address>`: Optional. Displays IP address of a multicast group.
- `<interface>`: Optional. Designates the display of parameters for a specific interface (in the format `type slot/port`). For example: `eth 0/1`.
- `summary`: Optional. Displays a single-line summary for each entry in the IP multicast routing table.
- `all`: Optional: Displays all multicast routes, including those not used to forward multicast traffic.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output from this command:

```
ProCurve>enable
ProCurve#show ip mroute

IP Multicast Routing Table
Timers: Uptime/Expires

(*, 10.2.170.3), 01:03:19/00:00:00
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    fr 1.20, Forward, 01:03:19/00:01:48
(*, 10.1.1.1), 00:00:01/00:02:58, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: <-- because lo 100 ifc is joined
    Loopback100, Forward/Dense, 00:00:01/00:00:00

(*, 10.0.1.50), 00:00:01/00:02:58, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback100, Forward/Dense, 00:00:01/00:00:00
```
The following is sample output from the `show ip mroute all` command:

ProCurve>enable
ProCurve#show ip mroute all
IP Multicast Routing Table
Flags: S - Sparse, C - Connected, P - Pruned, J - Join SPT, T - SPT-bit Set,
F - Register, R - RP-bit Set
Timers: Uptime/Expires

(*, 10.1.0.1), 01:17:34/00:03:25, RP 192.168.0.254, Flags: SC
Forwarding Entry: Yes
Incoming interface: tunnel 2, RPF nbr 172.16.2.10
Outgoing interface list:
  eth 0/1, Forward, 01:17:34/00:03:25
**show ip ospf**

Use the `show ip ospf` command to display general information regarding OSPF processes.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output from the `show ip ospf` command:

```
ProCurve>enable
ProCurve#show ip ospf

Summary of OSPF Process with ID: 192.168.72.101
  Supports only single Type Of Service routes (TOS 0)
  SPF delay timer: 5 seconds, Hold time between SPFs: 10 seconds
  LSA interval: 240 seconds
  Number of external LSAs: 0, Checksum Sum: 0x0
  Number of areas: 0, normal: 0, stub: 0, NSSA: 0
```
show ip ospf database

Use the **show ip ospf database** command to display information from the OSPF database regarding a specific router. There are several variations of this command which you can use to obtain information about different OSPF link state advertisements. The variations are shown below:

- `show ip ospf <area-id> database`
- `show ip ospf <area-id> database adv-router <ip address>`
- `show ip ospf <area-id> database database-summary`
- `show ip ospf <area-id> database external <link-state-id>`
- `show ip ospf <area-id> database external <link-state-id> adv-router <ip address>`
- `show ip ospf <area-id> database network <link-state-id>`
- `show ip ospf <area-id> database network <link-state-id> adv-router <ip address>`
- `show ip ospf <area-id> database router <link-state-id>`
- `show ip ospf <area-id> database router <link-state-id> adv-router <ip address>`
- `show ip ospf <area-id> database summary <link-state-id>`
- `show ip ospf <area-id> database summary <link-state-id> adv-router <ip address>`

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;area id&gt;</code></td>
<td>Optional. Displays area ID number associated with the OSPF address range. This range is defined in the network router configuration command used to define the particular area. See <code>network &lt;ip address&gt; &lt;wildcard&gt; area &lt;area id&gt;</code> on page 1243 for more information.</td>
</tr>
<tr>
<td><code>&lt;link-state-id&gt;</code></td>
<td>Optional. Identifies the portion of the Internet environment that is being described by the advertisement. The value needed in this field is tied to the advertisement’s LS type.</td>
</tr>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Enter in the form <code>&lt;A.B.C.D&gt;</code>.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.
Functional Notes

The link state ID differs depending on whether the link state advertisement in question describes a network or a router.

If describing a network, this ID is one of the following:

- The network’s IP address. This is true for type 3 summary link advertisements and in autonomous system external link advertisements.
- An address obtained from the link state ID. If the network link advertisement’s link state ID is masked with the network’s subnet mask, this will yield the network’s IP address.

If describing a router, this ID is always the router’s OSPF router ID.

Usage Examples

The following example shows the database link state summary for all areas.

ProCurve>enable
ProCurve#show ip ospf database
OSPF router with ID: 0.0.0.0
show ip ospf interface <interface type> <interface number>

Use the show ip ospf interface command to display OSPF information for a specific interface.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface type&gt;</td>
<td>Optional. Specifies the interface type. Type show ip ospf interface ? for a complete list of applicable interfaces.</td>
</tr>
<tr>
<td>&lt;interface number&gt;</td>
<td>Optional. Specifies the interface number.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example shows OSPF information for the ppp 1 interface.

ProCurve>enable
ProCurve#show ip ospf interface ppp 1
**show ip ospf neighbor** `<interface type> <interface number> <neighbor id>`

Use the `show ip ospf neighbor` command to display OSPF neighbor information for a specific interface.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;interface type&gt;</code></th>
<th>Optional. Specifies the interface type (i.e., eth, ppp, etc.).</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;interface number&gt;</code></td>
<td>Optional. Specifies the interface number.</td>
</tr>
<tr>
<td><code>&lt;neighbor id&gt;</code></td>
<td>Optional. Specifies a specific neighbor's router ID.</td>
</tr>
<tr>
<td><code>detail</code></td>
<td>Optional. Enter this keyword to display details on all neighbors.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

ProCurve>enable
ProCurve#show ip ospf neighbor
**show ip ospf summary-address**

Use the `show ip ospf summary-address` command to display a list of all summary address redistribution information for the system.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

```
ProCurve>enable
ProCurve#show ip ospf summary-address
```
show ip pim-sparse [interfaces <interface> | neighbor | rp-map | rp-set | state | traffic]

Use the `show ip pim-sparse` command to display Protocol Independent Multicast (PIM) configuration information. Sparse mode or PIM-SM is a routing protocol used to establish and maintain the multicast distribution tree. Routers can participate in the shared tree (RPT) rooted at the rendezvous point (RP) router or the shortest-path tree (SPT) rooted at a multicast source. PIM-SM also establishes both shared trees and shortest-path trees.

**Syntax Description**

- **interface <interface>**
  - Displays PIM-SM configuration and status information for a specific interface. Type `show ip pim-sparse interface ?` to display a list of applicable interfaces.

- **neighbor**
  - Displays neighbor adjacency information.

- **rp-map**
  - Displays list of statically configured RP candidates. The group address is 224.0.0.0/4 when no access group was applied to the `rp-address` command (refer to `rp-address <ip address> access-group <access-list-name>` on page 1251). Otherwise it is the name of the access group.

- **rp-set**
  - Displays list of statically configured RP candidates. The group address is 224.0.0.0/4 when no access group was applied to the `rp-address` command (refer to `rp-address <ip address> access-group <access-list-name>` on page 1251). Otherwise it is the name of the access group.

- **state**
  - Displays multicast route PIM state information.

- **traffic**
  - Displays active PIM-SM control traffic statistics.

**Default Values**

No default necessary for this command.

**Usage Examples**

The following example shows sample output from the `show ip pim-sparse` command:

ProCurve>enable
ProCurve#show ip pim-sparse
Global PIM Sparse Mode Settings
   Join/Prune interval: 60, SPT threshold: 1
The following example shows sample output from the `show ip pim-sparse interfaces` command:

ProCurve>enable
ProCurve#show ip pim-sparse interface

eth 0/1 is UP
   PIM Sparse
   DR: itself
   Local Address: 192.168.1.254
   Hello interval (sec): 30, Neighbor timeout (sec): 105
   Propagation delay (ms): 500, Override interval (ms): 2500

tunnel 1 is UP
   PIM Sparse
   DR: 172.16.1.10
   Local Address: 172.16.1.9
   Hello interval (sec): 30, Neighbor timeout (sec): 105
   Propagation delay (ms): 500, Override interval (ms): 2500

tunnel 2 is UP
   PIM Sparse
   DR: 172.16.2.10
   Local Address: 172.16.2.9
   Hello interval (sec): 30, Neighbor timeout (sec): 105
   Propagation delay (ms): 500, Override interval (ms): 2500

The following example shows sample output from the `show ip pim-sparse neighbor` command:

ProCurve>enable
ProCurve#show ip pim-sparse neighbor

<table>
<thead>
<tr>
<th>Port</th>
<th>Neighbor</th>
<th>Holdtime(sec)</th>
<th>Age(sec)</th>
<th>Uptime(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel 1</td>
<td>172.16.1.10</td>
<td>105</td>
<td>19</td>
<td>241908</td>
</tr>
<tr>
<td>tunnel 2</td>
<td>172.16.2.10</td>
<td>105</td>
<td>23</td>
<td>241913</td>
</tr>
</tbody>
</table>
The following example shows sample output from the `show ip pim-sparse rp-map` command:

```
ProCurve>enable
ProCurve#show ip pim-sparse rp-map
Number of group-to-RP mappings: 5
Group address   RP address
-------------------
225.1.0.1        192.168.0.254
225.1.0.2        192.168.0.254
225.1.0.3        192.168.0.254
```

The following example shows sample output from the `show ip pim-sparse rp-map set` command:

```
ProCurve>enable
ProCurve#show ip pim rp-map set
Group address   Static-RP-address
--------------------------
224.0.0.0/4       192.168.0.254
MCAST_ACL_1      192.168.1.254
MCAST_ACL_2      192.168.2.254
MCAST_ACL_3      192.168.3.254
```

The following example shows sample output from the `show ip pim-sparse state` command:

```
ProCurve>enable
ProCurve#show ip pim-sparse state
PIM-SM State Table
Flags: S - Sparse, C - Connected, P - Pruned, J - Join SPT, T - SPT-bit Set,
F - Register, R - RP-bit Set
Timers: Uptime/Expires

(*, 225.1.0.1), 02:42:03/00:03:04, RP 192.168.0.254, Flags: SC
  Forwarding Entry: Yes
  Incoming interface: tunnel 2, RPF nbr 172.16.2.10
  Upstream Join/Prune State: Joined
  Register State: No Info
  RegStop Timer (sec): stopped
  Join/Prune Timer (sec): 57
  Override Timer (sec): stopped
  Multicast Border Router: 0.0.0.0
  Packets Forwarded: 2
  Outgoing interface list:
```
eth 0/1, Forward, 02:42:03/00:03:03
   Downstream Join/Prune State: Join
   Assert Winner State: No Info
   Assert Timer (sec): stopped
   Assert Winner: 0.0.0.0
   Assert Winner Metric: infinity
   Local Membership: Yes
   Forwarding State: Forwarding
Inherited output list:
   eth 0/1

The following example shows sample output from the `show ip pim-sparse traffic` command:

```
ProCurve>enable
ProCurve#show ip pim-sparse traffic

<table>
<thead>
<tr>
<th></th>
<th>Rx</th>
<th></th>
<th>Tx</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Port:</td>
<td>Rx</td>
<td></td>
<td>Tx</td>
<td></td>
</tr>
<tr>
<td>eth 0/1</td>
<td>7</td>
<td>8334</td>
<td>J/P: 0</td>
<td>0</td>
</tr>
<tr>
<td>Hello:</td>
<td>0</td>
<td>0</td>
<td>RegStop: 0</td>
<td>0</td>
</tr>
<tr>
<td>Register:</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assert:</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tunnel 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hello:</td>
<td>8327</td>
<td>8333</td>
<td>J/P: 0</td>
<td>57</td>
</tr>
<tr>
<td>Register:</td>
<td>0</td>
<td>0</td>
<td>RegStop: 0</td>
<td>0</td>
</tr>
<tr>
<td>Assert:</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tunnel 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hello:</td>
<td>8323</td>
<td>8334</td>
<td>J/P: 0</td>
<td>11949</td>
</tr>
<tr>
<td>Register:</td>
<td>0</td>
<td>0</td>
<td>RegStop: 0</td>
<td>0</td>
</tr>
<tr>
<td>Assert:</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hello:</td>
<td>16657</td>
<td>25001</td>
<td>J/P: 0</td>
<td>12006</td>
</tr>
<tr>
<td>Register:</td>
<td>0</td>
<td>0</td>
<td>RegStop: 0</td>
<td>0</td>
</tr>
<tr>
<td>Assert:</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
show ip policy

Use the show ip policy command to display which route-map is associated with which interface for policy-based routing.

Syntax Description
No subcommands.

Default Values
No default value necessary for this command.

Usage Examples
The following is sample output from this command:

ProCurve>enable
ProCurve#show ip policy

<table>
<thead>
<tr>
<th>Interface</th>
<th>Route-map</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth 0/2</td>
<td>equal</td>
</tr>
<tr>
<td>eth 0/3</td>
<td>AAA-02/06/04-14:01:26.619-1-AppSpec (Dynamic)</td>
</tr>
</tbody>
</table>
**show ip policy-class** `<policyname>`

Use the `show ip policy-class` command to display the configured session limit and specific host IP addresses of all current sessions. See `ip policy-class <policyname>` on page 400 for information on configuring access policies. Variations of this command include:

- `show ip policy-class`
- `show ip policy-class <policyname>`
- `show ip policy-class host-sessions`
- `show ip policy-class <policyname> host-sessions`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>host-sessions</code></td>
<td>Optional. Displays specific host IP addresses of all current sessions</td>
</tr>
<tr>
<td><code>&lt;policyname&gt;</code></td>
<td>Optional. Displays policy class information for a specific policy class.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output from the `show ip policy-class` command:

```
ProCurve>enable
ProCurve#show ip policy-class

Maximum policy-sessions: 17400

Policy-class "Private":
  136 current sessions (5800 max)
  Entry 1 - allow list self self
  Entry 2 - nat source list wizard-ics interface ppp 1 overload

Policy-class "Public":
  0 current sessions (5800 max)
```
The following is a sample output from the \texttt{show ip policy-class host-sessions} command:

\begin{verbatim}
ProCurve>enable
ProCurve#show ip policy-class host-sessions

Policy-class "Private":
100 policy-sessions allowed per source address.

\begin{tabular}{ll}
\textbf{Src IP Address} & \textbf{Sessions} \\
192.168.1.100 & 1 \\
192.168.1.101 & 35 \\
192.168.1.121 & 100 (maximum allowed)
\end{tabular}

Policy-class "Public":
No limit for policy-sessions allowed per host.

The following is a sample output from the \texttt{show ip policy-class <policyname> host-sessions} command for the policy class named \texttt{Private}:

\begin{verbatim}
ProCurve>enable
ProCurve#show ip policy-class Private host-sessions

Policy-class "Private":
100 policy-sessions allowed per source address.

\begin{tabular}{ll}
\textbf{Src IP Address} & \textbf{Sessions} \\
192.168.1.100 & 1 \\
192.168.1.101 & 35 \\
192.168.1.121 & 100 (maximum allowed)
\end{tabular}
\end{verbatim}
show ip policy-sessions <policyname> [all]

Use the show ip policy-sessions command to display a list of current policy class associations. See ip policy-class <policyname> on page 400 for information on configuring access policies.

Syntax Description

| <policyname> | Optional. Displays policy class associations for a specific policy class. |
| all          | Optional. Displays all policy sessions, including active associations (through which the firewall is allowed to pass traffic) and associations flagged for deletion (through which the firewall is forbidden to pass traffic). Associations flagged for deletion will usually be freed within a few seconds of timeout or deletion, depending on packet congestion; servicing of packets is given priority. New traffic matching an association will create a new active association, provided the traffic still matches a policy-class allow or NAT entry. |

Default Values

No default value necessary for this command.

Usage Examples

The following is sample output from the show ip policy-sessions command:

ProCurve>enable
ProCurve#show ip policy-sessions

<table>
<thead>
<tr>
<th>Protocol (TTL) [in crypto map] -&gt; [out crypto map] Destination policy-class</th>
<th>Src IP Address</th>
<th>Src Port</th>
<th>Dest IP Address</th>
<th>Dst Port</th>
<th>NAT IP Address</th>
<th>NAT Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy class “Public”:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp (13)</td>
<td>192.168.1.142</td>
<td>2621</td>
<td>192.168.19.2</td>
<td>1</td>
<td>10.10.10.1</td>
<td>3000</td>
</tr>
<tr>
<td>tcp (13)</td>
<td>192.168.1.142</td>
<td>2622</td>
<td>192.168.19.2</td>
<td>2</td>
<td>10.10.10.1</td>
<td>3001</td>
</tr>
<tr>
<td>tcp (13)</td>
<td>192.168.1.142</td>
<td>2623</td>
<td>192.168.19.2</td>
<td>3</td>
<td>10.10.10.1</td>
<td>3002</td>
</tr>
<tr>
<td>tcp (13)</td>
<td>192.168.1.142</td>
<td>2624</td>
<td>192.168.19.2</td>
<td>4</td>
<td>10.10.10.1</td>
<td>3003</td>
</tr>
</tbody>
</table>
The following is sample output from the `show ip policy-sessions all` command:

```
ProCurve>enable
ProCurve#show ip policy-sessions all
Protocol (TTL) [in crypto map] -> [out crypto map] Destination policy-class
Src IP Address  Src Port Dest IP Address Dst Port NAT IP Address NAT Port
-------------------------------------------------------------------------------------------------------------------------------
Policy class "Public":
tcp (0) - inactive
  192.168.1.142  1025  192.168.19.2  3135  10.10.10.1  3605
  192.168.1.142  1028  192.168.19.2  3138  10.10.10.1  3606
  192.168.1.142  1029  192.168.19.2  3139  10.10.10.1  3607
  192.168.1.142  1036  192.168.19.2  3146  10.10.10.1  3608
```
**show ip policy-stats** *<policyname>*

Use the `show ip policy-stats` command to display a list of current policy class statistics. See `ip policy-class <policyname>` on page 400 for information on configuring access policies.

**Syntax Description**

| <policyname> | Optional. Enter a specific policy class name to display information for a single policy. |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example displays a list of current policy class statistics:

```
ProCurve>enable
ProCurve#show ip policy-stats
```
show ip prefix-list [detail | summary] <listname>

Use the show ip prefix-list command to display BGP prefix list information.

Syntax Description

detail Shows a listing of the prefix list rules and their hit counts.
summary Shows information about the entire prefix list.
<listname> Specifies to display information for a particular prefix list.

Default Values

No default values are necessary for this command.

Functional Notes

If the show ip prefix-list command is issued with no arguments, a listing of the prefix-list rules but no hit count statistics is displayed.

Usage Examples

The following example displays information about the prefix list test:

ProCurve#show ip prefix-list test
ip prefix-list test: 4 entries
    seq 5 permit 0.0.0.0/0 ge 8 le 8
    seq 10 deny 0.0.0.0/0 ge 9 le 9
    seq 15 permit 0.0.0.0/0 ge 10 le 10
    seq 20 deny 0.0.0.0/0 ge 11
show ip protocols

Use the show ip protocols command to display IP routing protocol parameters and statistics.

Syntax Description
No subcommands.

Default Values
No default value necessary for this command.

Usage Examples
The following is a sample output from the show ip protocols command:

ProCurve>enable
ProCurve#show ip protocols
Sending updates every 30 seconds, next due in 8 seconds
  Invalid after 180 seconds, hold down time is 120 seconds
  Redistributing: rip
  Default version control: send version 2, receive version 2
Interface   Send Ver.  Rec Ver.
  eth 0/1     2          2
  ppp 1       2          2
Routing for networks:
  1.1.1.0/24
show ip route [connected | ospf | rip | static | table | bgp | summary | summary realtime | <ip address> <subnet>]

Use the `show ip route` command to display the contents of the IP route table.

**Syntax Description**

- **connected**: Optional. Displays only the IP routes for directly connected networks.
- **ospf**: Optional. Displays only the IP routes associated with OSPF.
- **rip**: Optional. Displays only the IP routes that were dynamically learned through RIP.
- **static**: Optional. Displays only the IP routes that were statically entered.
- **table**: Optional. Displays a condensed version of the IP route table.
- **bgp**: Optional. Displays only the IP routes associated with BGP.
- **summary**: Optional. Displays a summary of all IP route information.
- **summary realtime**: Optional. Displays full-screen output in realtime. See the **Functional Notes** below for more information.
- **<ip address> <subnet>**: Displays only the IP routes to destinations within the given address and subnet.

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `terminal length` command; refer to `terminal length <lines>` on page 280).
Usage Examples

The following is a sample output from the `show ip route` command:

ProCurve>enable
ProCurve#show ip route rip

Codes: C - connected S - static R - RIP O - OSPF IA - OSPF inter area
       N1 - OSPF NSSA external type 1 N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1 E2 - OSPF external type 2

Gateway of last resort is 10.15.227.29 to network 0.0.0.0

R   10.15.223.0/30 [120/1] via 10.15.227.41, ppp 3
R   10.15.223.2/32 [120/1] via 10.15.227.41, ppp 3
R   10.15.226.0/28 [120/1] via 10.15.227.41, ppp 3
R   10.15.226.16/28 [120/1] via 10.15.227.41, ppp 3
R   10.15.226.48/28 [120/1] via 10.15.227.29, ppp 1
R   10.15.226.96/28 [120/1] via 10.15.227.29, ppp 1

The following example shows how to display IP routes learned via BGP. The values in brackets after a BGP route entry represent the entry’s administrative distance and metric:

ProCurve>enable
ProCurve#show ip route bgp

Codes: C - connected, S - static, R - RIP, O - OSPF, B - BGP
       IA - OSPF inter area, N1 - OSPF NSSA external type 1
       N2 - OSPF NSSA external type 2, E1 - OSPF external type 1
       E2 - OSPF external type 2

Gateway of last resort is 10.15.43.17 to network 0.0.0.0

B   1.0.0.0/8 [30/0] via 10.15.43.17, fr 1.17
B   2.0.0.0/9 [30/0] via 10.15.43.17, fr 1.17
B   2.128.0.0/10 [30/0] via 10.15.43.17, fr 1.17
B   2.192.0.0/11 [30/0] via 10.15.43.17, fr 1.17
B   2.224.0.0/12 [30/0] via 10.15.43.17, fr 1.17
B   2.240.0.0/13 [30/0] via 10.15.43.17, fr 1.17
B   2.248.0.0/14 [30/0] via 10.15.43.17, fr 1.17
**show ip traffic [realtime]**

Use the `show ip traffic` command to display all IP traffic statistics.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>realtime</td>
<td>Displays full-screen output in real-time. See the <em>Functional Notes</em> below for more information.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `terminal length` command; refer to `terminal length <lines>` on page 280).

**Usage Examples**

The following is a sample output from the `show ip traffic` command:

```
ProCurve>enable
ProCurve#show ip traffic

IP statistics:
  Routing discards: 0
  Rcvd:  15873 total, 7617 delivered
    0 header errors, 0 address errors
    0 unknown protocol, 0 discards
    0 checksum errors, 0 bad hop counts
  Sent:  8281 generated, 4459 forwarded
    0 no routes, 0 discards
  Frags: 0 reassemble required, 0 reassembled, 0 couldn't reassemble
    0 created, 0 fragmented, 0 couldn't fragment

UDP statistics:
  Rcvd: 3822 total, 0 checksum errors, 0 no port
  Sent: 3822 total

TCP statistics:
  Retrans Timeout Algorithm: 0
```
Min retrans timeout (ms): 0
Max retrans timeout (ms): 0
Max TCP Connections: 0
0 active opens, 64 passive opens, 0 failed attempts
5 establish resets, 1 establish current
3795 segments received, 4459 segments sent, 26 segments retransmitted
show isdn-group <group id>

Use the show isdn group command to display integrated services digital network (ISDN) group information.

Syntax Description

| <group id> | Displays information for a specific ISDN group. Valid range: 1 to 255. |

Default Values

No default value necessary for this command.

Usage Examples

The following example displays information for ISDN group 5:

ProCurve>enable
ProCurve#show isdn-group 5
show lldp

Use the show lldp command to display LLDP timer configuration.

Syntax Description

No subcommands.

Default Values

No default values are necessary for this command.

Usage Examples

The following example shows a sample LLDP timer configuration:

ProCurve>enable
ProCurve#show lldp
Global LLDP information:
Sending LLDP packets every 30 seconds
Sending TTL of 120 seconds
**show lldp device** `<system name>`

Use the `show lldp device` command to display specific neighbor information about a given neighbor.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;system name&gt;</code></td>
<td>Specifies the system name of the neighbor to display.</td>
</tr>
</tbody>
</table>

**Default Values**

No default values are necessary for this command.

**Functional Notes**

If there is more than one neighbor with the same system name, all neighbors with that system name will be displayed.

**Usage Examples**

The following example shows specific information about a neighbor for the system name **Router**:

ProCurve#`show lldp device Router`

Chassis ID: 00:12:79:02:DD:2A (MAC Address)
- System Name: Router
- Device Port: eth 0/1 (Locally Assigned)
- Holdtime: 30
- Platform: 3305
- Software: Version: 08.00.22.sw1.D, Date: Mon Nov 01 10:28:55 2004
- Capabilities: Bridge, Router
- Enabled Capabilities: Router
- Local Port: eth 0/2
- Management Addresses:
  - Address Type: IP version 4, Address: 10.23.10.10
  - Interface Type: Interface Index, Interface Id: 2
**show lldp interface** *<interface>*

Use the `show lldp interface` command to display LLDP configuration and statistics for interfaces on this device.

**Syntax Description**

| `<interface>` | Displays the information for the specified interface. Type `show lldp interface ?` for a complete list of applicable interfaces. |

**Default Values**

No default values are necessary for this command.

**Usage Examples**

The following example shows LLDP configuration and statistics for the Ethernet 0/1 interface:

```
ProCurve#show lldp interface ethernet 0/1
eth 0/1 (TX/RX)
  0 packets input
    0 input errors
    0 TLV errors, 0 TLVs Discarded
    0 packets discarded
  8799 packets output
  0 neighbor ageouts
ProCurve#
```
**show lldp neighbors [interface <interface type> <interface>] [detail | realtime]**

Use the `show lldp neighbors interface` command to display information about neighbors of this device learned about via LLDP.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>interface &lt;interface type&gt;</strong></td>
<td>Displays a summary of all neighbors learned about through interfaces of the specified type (e.g., <code>eth</code>).</td>
</tr>
<tr>
<td><code>&lt;interface&gt;</code></td>
<td>Displays a summary of all neighbors learned about through a specified interface (e.g., <code>eth 0/1</code>). Type <code>show lldp neighbors interface ?</code> for a complete list of applicable interfaces.</td>
</tr>
<tr>
<td><strong>detail</strong></td>
<td>Optional. Shows detailed neighbor information for the specified interface or interface type.</td>
</tr>
<tr>
<td><strong>realtime</strong></td>
<td>Displays full-screen output in real-time. Refer to the Functional Notes below for more information.</td>
</tr>
</tbody>
</table>

**Default Values**

No default values necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `terminal length` command; refer to `terminal length <lines>` on page 280).

**Usage Examples**

The following example shows detailed information about a device’s neighbors:

```
ProCurve#show lldp neighbors interface eth 0/2 detail
Chassis ID: 00:12:79:02:DD:2A (MAC Address)
System Name: Router
Device Port: eth 0/1 (Locally Assigned)
Holdtime: 38
Platform: 3305
Software: Version: 08.00.22.sw1.D, Date: Mon Nov 01 10:28:55 2004
Capabilities: Bridge, Router
Enabled Capabilities: Router
Local Port: eth 0/2
Management Addresses:
  Address Type: IP version 4, Address: 10.23.10.10
Interface Type: Interface Index, Interface Id: 2
```
show lldp neighbors statistics

Use the **show lldp neighbors statistics** command to display statistics about LLDP neighbor table actions.

**Syntax Description**

No subcommands.

**Default Values**

There are no default values necessary for this command.

**Functional Notes**

This command shows information about the changes in this device’s neighbor table. The information displayed indicates the last time a neighbor was added to or removed from the table as well as the number of times neighbors were inserted into or deleted from the table.

**System Last Change Time**

Shows the time at which the most recent change occurred in the neighbor table.

**Inserts**

Shows the number of times neighbors have been added to the table.

**Deletes**

Shows how many times neighbors have been deleted from the table because an interface was shut down.

**Drops**

Shows how many times the insertion of a new neighbor into the table failed because the table was full.

**Age Outs**

Shows how many times neighbors have been removed from the table because no new updates were received from that neighbor before its time-to-live timer expired.

**Usage Examples**

The following example shows sample output for this command:

```
ProCurve>enable
ProCurve#show lldp neighbors statistics
System Last Change Time        Inserts  Deletes  Drops  Age Outs
10-15-2004 14:24:56            55       3    1      1
```
show memory [heap | realtime]

Use the `show memory` command to display statistics regarding memory including memory allocation and buffer use statistics. Shows how memory is in use (broken down by memory size) and how much memory is free.

**Syntax Description**

- **heap**: Shows how much memory is in use (broken down by memory block size) and how much memory is free.
- **realtime**: Displays full-screen output in real-time. See the *Functional Notes* below for more information.

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `terminal length` command; refer to `terminal length <lines>` on page 280).

**Usage Examples**

The following example is a sample output from the `show memory heap` command:

```
ProCurve>enable
ProCurve#show memory heap
Memory Heap:
   HeapFree: 2935792
   HeapSize: 8522736
Block Managers:

    Mgr  Size  Used  Free  Max-Used
       0    0     58   0     58
       1    16   1263  10   1273
       2    48   1225   2   1227
       3    112   432   2   434
       4    240   140   3   143
       5    496    72   2    74
       6   1008    76   1    26
       7   2032    25   1    26
       8   4080    2   1    3
       9   8176    31   1    32
      10  16368     8   0     8
```
show modules

Use the show modules command to display a list of the modules currently installed in the system.

Syntax Description

No subcommands.

Default Value

No default value necessary for this command.

Usage Examples

The following is a sample output from the show modules command:

ProCurve>enable
ProCurve#show modules

<table>
<thead>
<tr>
<th>Slot</th>
<th>Port</th>
<th>Type</th>
<th>Part Number</th>
<th>Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1–2</td>
<td>E1/E1 WAN</td>
<td>J8456A</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1–2</td>
<td>E1/E1 WAN</td>
<td>J8456A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1–8</td>
<td>Octal E1</td>
<td>J8463A</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>Empty</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
show output-startup

Use the show output-startup command to display startup configuration output line-by-line. This output can be copied into a text file and then used as a configuration editing tool.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following is a sample output from the show output-startup command:

ProCurve>enable
ProCurve#show output-startup

!  
ProCurve#!
ProCurve#hostname "UNIT_2"
UNIT_2#no enable password
UNIT_2#!
UNIT_2#ip subnet-zero
UNIT_2#ip classless
UNIT_2#ip routing
UNIT_2#!
UNIT_2#event-history on
UNIT_2#no logging forwarding
UNIT_2#logging forwarding priority-level info
UNIT_2#no logging email
UNIT_2#no logging email
etc....
**show port-auth supplicant [interface ethernet <slot/port> | summary]**

Use the `show port-auth` command to display supplicant information pertaining to port authentication. The supplicant is the port that will receive services from the port authenticator.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>interface ethernet &lt;slot/port&gt;</code></td>
<td>Optional. Shows port authorization supplicant information related to a specific Ethernet interface.</td>
</tr>
<tr>
<td><code>summary</code></td>
<td>Optional. Shows only basic information about each applicable interface.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example displays supplicant information for Ethernet interface 0/2:

```
ProCurve>enable
ProCurve#show port-auth supplicant interface eth 0/2
```

**Interface: eth 0/2**

- Local Supplicant mode is enabled
- Username: User1
- Password: securePass
- Authorization Status: yes
- Supp State Machine: CONNECTED
show pppoe

Use the **show pppoe** command to display all point-to-point over Ethernet (PPPoE) settings and associated parameters.

**Syntax Description**
No subcommands.

**Default Values**
No default value necessary for this command.

**Usage Examples**
The following example enters the Enable Command mode and uses the **show** command to display pppoe information:

ProCurve>**enable**
ProCurve#**show pppoe**
ppp 1
   Outgoing Interface: eth 0/1
   Outgoing Interface MAC Address: 00:12:79:00:85:20
   Access-Concentrator Name Requested: FIRST VALID
   Access-Concentrator Name Received: 13021109813703-LRVLGSROS20W_IFITL
   Access-Concentrator MAC Address: 00:10:67:00:1D:B8
   Session Id: 64508
   Service Name Requested: ANY
   Service Name Available: 
   PPPoE Client State: Bound (3)
   Redial retries: unlimited
   Redial delay: 10 seconds
   Backup enabled all day on the following days:
      Sunday Monday Tuesday Wednesday Thursday Friday Saturday
   Backup phone number list:
      | Number | Call Type | min/max DS0s | Backup I/F |
      |--------|-----------|--------------|------------|
      | 5551212| analog    | 1/1          | ppp 2      |
show processes [cpu | cpu realtime | history | queue]

Use the show processes command to display process statistic information.

Syntax Description

cpu
Displays informations about processes that are currently active.

cpu realtime
Displays full-screen CPU output in real-time. See the Functional Notes below for more information.

history
Displays the process switch history.

queue
Displays process queue utilization.

Default Values

No default value necessary for this command.

Functional Notes

Use the realtime argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the terminal length command; refer to terminal length <lines> on page 280).

Usage Examples

The following is a sample output from the show processes cpu command:

ProCurve>enable
ProCurve#show processes cpu
processes cpu
System load: 7.07%  Min: 0.00%  Max: 85.89%
Context switch load: 0.21%

<table>
<thead>
<tr>
<th>Task</th>
<th>Invoked</th>
<th>Exec</th>
<th>Time</th>
<th>Runtime</th>
<th>Load %</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Idle</td>
<td>0 W</td>
<td>129689</td>
<td>1971</td>
<td>927923</td>
</tr>
<tr>
<td>1</td>
<td>FrontPanel</td>
<td>249 W</td>
<td>9658</td>
<td>165</td>
<td>3202</td>
</tr>
<tr>
<td>3</td>
<td>Stack Usage</td>
<td>11 W</td>
<td>485</td>
<td>305</td>
<td>325</td>
</tr>
<tr>
<td>4</td>
<td>Q Test 1</td>
<td>10 W</td>
<td>50</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Q Test 2</td>
<td>11 W</td>
<td>50</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>Clock</td>
<td>20 W</td>
<td>1443</td>
<td>24</td>
<td>55</td>
</tr>
<tr>
<td>11</td>
<td>PacketRouting</td>
<td>250 W</td>
<td>31656</td>
<td>10</td>
<td>3871</td>
</tr>
<tr>
<td>12</td>
<td>Thread Pool</td>
<td>50 W</td>
<td>161</td>
<td>159</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>IKE</td>
<td>10 W</td>
<td>2</td>
<td>341</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>RouteTableTick</td>
<td>50 W</td>
<td>49</td>
<td>874</td>
<td>874</td>
</tr>
</tbody>
</table>

....etc.
show qos map

The **show qos map** command outputs information about the QoS map. This information differs based on how a particular map entry is defined. Variations of this command include the following:

```
show qos map
show qos map <map name>
show qos map <map name> <sequence number>
show qos map interface <interface ID>
```

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;map name&gt;</code></th>
<th>Enter the name of a defined QoS map.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;sequence number&gt;</code></td>
<td>Enter one of the map’s defined sequence numbers.</td>
</tr>
<tr>
<td><code>interface &lt;interface ID&gt;</code></td>
<td>Displays QoS map information for a specific interface (e.g., Frame Relay, PPP, or ATM). Enter the <strong>show qos map interface ?</strong> command for a complete list of interfaces.</td>
</tr>
</tbody>
</table>

**Default Values**

No defaults necessary for this command.

**Usage Example**

```
ProCurve#show qos map
qos map priority
map entry 10
  match IP packets with a precedence value of 6
  priority bandwidth: 400 (kilobits/sec) burst: default
  packets matched by map: 125520

map entry 20
  match ACL icmp
  packets matched by map: 99

map entry 30
  match RTP packets on even destination ports between 16000 and 17000
  packets matched by map: 0

map entry 50
  match ACL tcp
  packets matched by map: 4326

map entry 60
  match IP packets with a dscp value of 2
  set dscp value to 6
  packets matched by map: 0
```
map entry 70
match NetBEUI frames being bridged by the router
priority bandwidth: 150 (kilobits/sec) burst: default
packets matched by map: 0

qos map tcp_map
map entry 10
  match ACL tcp
  priority bandwidth: 10 (kilobits/sec) burst: default
  set precedence value to 5
  packets matched by map: 0
map entry 20
  match IP packets with a precedence value of 3
  priority bandwidth: 50 (kilobits/sec) burst: default
  packets matched by map: 0

The following example shows the “priority” qos map and all entries in that map:
ProCurve#show qos map priority
qos map priority
map entry 10
  match IP packets with a precedence value of 6
  priority bandwidth: 400 (kilobits/sec) burst: default
  packets matched by map: 125520
map entry 20
  match ACL icmp
  packets matched by map: 99
map entry 30
  match RTP packets on even destination ports between 16000 and 17000
  packets matched by map: 0

map entry 50
  match ACL tcp
  packets matched by map: 4326
map entry 60
  match IP packets with a dscp value of 2
  set dscp value to 6
  packets matched by map: 0
map entry 70
  match NetBEUI frames being bridged by the router
  priority bandwidth: 150 (kilobits/sec) burst: default
  packets matched by map: 0
The following example shows a particular QoS map entry (in this case map entry 10):

ProCurve#show qos map priority 10
qos map priority
map entry 10
   match IP packets with a precedence value of 6
   priority bandwidth: 400 (kilobits/sec) burst: default
   packets matched by map: 125520

The following examples show QoS map interface stats associated with the map defined for an interface:

ProCurve#show qos map interface frame-relay 1
fr 1
qos-policy out: priority

map entry 10
   match IP packets with a precedence value of 6
   budget 145/10000 bytes (current/max)
   priority bandwidth: 400 (kilobits/sec)
   packets matched on interface: 27289
   packets dropped: 98231
map entry 20
   not configured for rate limiting
map entry 30
   not configured for rate limiting
map entry 50
   not configured for rate limiting
map entry 60
   not configured for rate limiting
map entry 70
   match NetBEUI frames being bridged by the router
   budget 3750/3750 bytes (current/max)
   priority bandwidth: 150 (kilobits/sec)
   packets matched on interface: 0
   packets dropped: 0
show queue <interface>

Use the show queue command to display conversation information associated with an interface queue. This command shows summary and per-conversation information.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;interface&gt;</th>
<th>Displays the queueing information for the specified interface. Type the show queue ? command to display a list of valid interface types.</th>
</tr>
</thead>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following is a sample output from the show queue command:

ProCurve>enable
ProCurve#show queue fr 1

Queueing method: weighted fair
Output queue: 18/25/200/64/1027 (size/highest/max total/threshold/drops)
    Conversations 2/4/256 (active/max active/max total)

    (depth/weight/highest/discards) 12/256/33/0
    Conversation 10, linktype: ip, length: 67
    source: 10.100.23.11, destination: 10.200.2.125, id: 0x0000, ttl: 47,
    TOS: 0 prot: 17 (udp), source port 99, destination port 99

    (depth/weight/highest/discards) 6/256/25/0
    Conversation 23, linktype: ip, length: 258
    source: 10.100.23.11, destination: 10.200.2.125, id: 0x0000, ttl: 47,
    TOS: 0 prot: 6 (tcp), source port 16, destination port 16
show queuing [fair]

Use the show queuing command to display information associated with configured queuing methods.

Syntax Description

| fair | Optional keyword used to display only information on the weighted fair queuing configuration. |

Default Values

No default value necessary for this command.

Usage Examples

The following is a sample output from the show queuing command:

ProCurve>enable
ProCurve#show queuing

<table>
<thead>
<tr>
<th>Interface</th>
<th>Discard threshold</th>
<th>Conversation subqueues</th>
</tr>
</thead>
<tbody>
<tr>
<td>fr 1</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>fr 2</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>ppp 1</td>
<td>64</td>
<td>256</td>
</tr>
</tbody>
</table>
show radius statistics

Use the show radius statistics command to display various statistics from the RADIUS subsystem. These statistics include number of packets sent, number of invalid responses, number of timeouts, average packet delay, and maximum packet delay. Statistics are shown for both authentication and accounting packets.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following is an example output using the show radius statistics command:

ProCurve>enable
ProCurve#show radius statistics

<table>
<thead>
<tr>
<th></th>
<th>Auth.</th>
<th>Acct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of packets sent:</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Number of invalid responses:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number of timeouts:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Average delay:</td>
<td>2 ms</td>
<td>0 ms</td>
</tr>
<tr>
<td>Maximum delay:</td>
<td>3 ms</td>
<td>0 ms</td>
</tr>
</tbody>
</table>
show route-map [<name>]

Use the show route-map command to display any route maps that have been configured in the router. It displays any match and set clauses associated with the route map, as well as the number of incoming routes that have matched each route map.

Syntax Description

<name> Optional. Displays only the route map matching the specified name.

Default Values

By default, this command displays all defined route maps.

Usage Examples

In the example below, all route maps in the router are displayed.

ProCurve>enable
ProCurve#show route-map
route-map RouteMap1, permit, sequence 10
  Match clauses:
    community (community-list filter): CommList1
Set clauses:
  local-preference 250
BGP Filtering matches: 75 routes
route-map RouteMap1, permit, sequence 20
  Match clauses:
    community (community-list filter): CommList2
Set clauses:
  local-preference 350
BGP Filtering matches: 87 routes
route-map RouteMap2, permit, sequence 10
  Match clauses:
    ip address (access-lists): 192.168.1.1
Set clauses:
  metric 100
BGP Filtering matches: 10 routes
route-map RouteMap2, permit, sequence 20
  Match clauses:
    ip address (access-lists): 192.168.2.1
Set clauses:
  metric 200
BGP Filtering matches: 12 routes

In the example below, only the route map with the name "RouteMap2" is displayed.

ProCurve>enable
ProCurve#show route-map RouteMap2
route-map RouteMap2, permit, sequence 10
  Match clauses:
   ip address (access-lists): 192.168.1.1
  Set clauses:
   metric 100
BGP Filtering matches: 10 routes
route-map RouteMap2, permit, sequence 20
  Match clauses:
   ip address (access-lists): 192.168.2.1
  Set clauses:
   metric 200
BGP Filtering matches: 12 routes
show running-config

Use the `show running-config` command to display a text print of all the non-default parameters contained in the current running configuration file. Specific portions of the running-config may be displayed, based on the command entered. Variations of this command include the following:

- **show running-config**
- **show running-config access-lists**
- **show running-config access-lists verbose**
- **show running-config checksum**
- **show running-config interface `<interface type> <interface id>`**
- **show running-config interface `<interface type> <interface id> verbose`**
- **show running-config ip-crypto**
- **show running-config ip-crypto verbose**
- **show running-config policy-class**
- **show running-config policy-class verbose**
- **show running-config qos-map**
- **show running-config qos-map verbose**
- **show running-config router bgp**
- **show running-config router bgp verbose**
- **show running-config router ospf**
- **show running-config router ospf verbose**
- **show running-config router pim-sparse**
- **show running-config router pim-sparse verbose**
- **show running-config router rip**
- **show running-config router rip verbose**
- **show running-config verbose**

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>access-lists</code></td>
<td>Displays the current running configuration for all configured IP access lists.</td>
</tr>
<tr>
<td><code>checksum</code></td>
<td>Optional. Displays the encrypted Message Digest 5 (MD5) version of the running configuration.</td>
</tr>
<tr>
<td><code>interface</code></td>
<td>Displays the current running configuration for a particular interface. Type <code>show running-config interface</code> ? for a list of valid interfaces.</td>
</tr>
<tr>
<td><code>&lt;interface id&gt;</code></td>
<td>Specifies any valid slot/port interface (e.g., 0/1).</td>
</tr>
<tr>
<td><code>ip crypto</code></td>
<td>Displays the current running configuration for all IPSec VPN settings.</td>
</tr>
<tr>
<td><code>policy-class</code></td>
<td>Displays the current running configuration for all configured policy classes.</td>
</tr>
<tr>
<td><code>qos-map</code></td>
<td>Displays the current running configuration for all configured QoS maps.</td>
</tr>
<tr>
<td><code>router bgp</code></td>
<td>Optional. Displays the current bgp configuration.</td>
</tr>
<tr>
<td><code>router ospf</code></td>
<td>Optional. Displays the current ospf configuration.</td>
</tr>
<tr>
<td><code>router pim-sparse</code></td>
<td>Optional. Displays the current global PIM-SM configuration.</td>
</tr>
<tr>
<td><code>router rip</code></td>
<td>Optional. Displays the current RIP configuration.</td>
</tr>
<tr>
<td><code>verbose</code></td>
<td>Optional. Displays the entire running configuration to the terminal screen (versus only the non-default values).</td>
</tr>
</tbody>
</table>
Default Values
No default value necessary for this command.

Usage Examples
The following is a sample output from the show running-config command:

ProCurve>enable
ProCurve#show running-config
Building configuration...
!
no enable password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
!
ip policy-timeout tcp all-ports 600
ip policy-timeout udp all-ports 60
ip policy-timeout icmp 60
!
interface eth 0/1........
show sip [resources | statistics | user-registration]

Use the show sip command to display Session Initiation Protocol (SIP) statistical and registration information.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resources</td>
<td>Displays SIP server resource information.</td>
</tr>
<tr>
<td>statistics</td>
<td>Displays SIP server statistic information.</td>
</tr>
<tr>
<td>user-registration</td>
<td>Displays local SIP server registration information.</td>
</tr>
</tbody>
</table>

Default Values

No default necessary for this command.

Usage Examples

The following example shows sample output from the show sip statistics command:

ProCurve>enable
ProCurve#show sip statistics

Invites transmitted:  36
Invites received:  26

Invite Retransmits transmitted:  11
Invite Retransmits received:  0

Non-Invites transmitted:  1869
Non-Invites received:  1911

Non-Invite Retransmits transmitted:  12
Non-Invite Retransmits received:  41

Responses transmitted:  1982
Responses received:  3535

Response Retransmits transmitted:  45
Response Retransmits received:  0
The following example shows sample output from the `show sip user-registration` command:

ProCurve>**enable**
ProCurve#**show sip user-registration**

<table>
<thead>
<tr>
<th>EXT.</th>
<th>TYPE</th>
<th>IP ADDRESS</th>
<th>PORT</th>
<th>EXP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1022</td>
<td>SIP - Generic</td>
<td>172.22.9.41</td>
<td>6378</td>
<td>2445</td>
</tr>
<tr>
<td>1430</td>
<td>SIP - Generic</td>
<td>10.17.10.12</td>
<td>8412</td>
<td>1400</td>
</tr>
<tr>
<td>1644</td>
<td>SIP - Generic</td>
<td>10.17.114.1</td>
<td>6036</td>
<td>1424</td>
</tr>
<tr>
<td>1833</td>
<td>SIP - Generic</td>
<td>172.30.15.131</td>
<td>7296</td>
<td>47</td>
</tr>
<tr>
<td>2002</td>
<td>SIP - Generic</td>
<td>10.17.20.24</td>
<td>5060</td>
<td>2593</td>
</tr>
</tbody>
</table>

Total phones registered: 5
show sip location [dynamic | static]

Use the `show sip location` command to display Session Initiation Protocol (SIP) statistical and registration information.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dynamic</td>
<td>Displays SIP location database dynamic entries.</td>
</tr>
<tr>
<td>static</td>
<td>Displays SIP location database static entries.</td>
</tr>
</tbody>
</table>

**Default Values**

No default necessary for this command.

**Usage Examples**

The following example shows sample output from the `show sip location static` command:

```
ProCurve>enable
ProCurve#show sip location static

User  IP Address Port Expires Source
-----------------------------------
Test   10.1.1.1   5060   0  User Config
```
show snmp

Use the **show snmp** command to display the system Simple Network Management Protocol (SNMP) parameters and current status of SNMP communications.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is an example output using the **show snmp** command for a system with SNMP disabled and the default Chassis and Contact parameters:

ProCurve> **show snmp**

Chassis: Chassis ID
Contact Name:
Contact Phone:
Contact Email:
Contact Pager:
Management URL:
Management URL Label:
0 Rx SNMP packets
  0 Bad community names
  0 Bad community uses
  0 Bad versions
  0 Silent drops
  0 Proxy drops
  0 ASN parse errors

show sntp

Use the show sntp command to display the system Simple Network Time Protocol (SNTP) parameters and current status of SNTP communications.

Syntax Description
No subcommands.

Default Values
No default value necessary for this command.

Usage Examples
The following example displays SNTP parameters and current status:

ProCurve> show sntp
show spanning-tree <bridgegroup#> [realtime]

Use the `show spanning-tree` command to display the status of the spanning-tree protocol.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;bridgegroup#&gt;</td>
<td>Optional. Display spanning-tree for a specific bridge group.</td>
</tr>
<tr>
<td>realtime</td>
<td>Shows full screen spanning-tree protocol status in real-time.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `terminal length` command).

**Usage Examples**

The following is an example output using the `show spanning-tree` command:

```
ProCurve> enable
ProCurve# show spanning-tree
Spanning Tree enabled protocol ieee
Root ID  Priority  32768
  Address  00:12:79:00:88:41
  We are the root of the spanning tree
  Hello Time  2 sec  Max Age  20 sec  Forward Delay  15 sec
Bridge ID  Priority  32768
  Address  00:12:79:00:88:41
  Hello Time  2 sec  Max Age  20 sec  Forward Delay  15 sec
  Aging Time  300

<table>
<thead>
<tr>
<th>Interface</th>
<th>Role</th>
<th>Sts</th>
<th>Cost</th>
<th>Prio.Nbr</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth 0/2</td>
<td>Desg</td>
<td>FWD</td>
<td>19</td>
<td>128.2</td>
<td>P2p</td>
</tr>
<tr>
<td>eth 0/3</td>
<td>Desg</td>
<td>FWD</td>
<td>19</td>
<td>128.3</td>
<td>P2p</td>
</tr>
<tr>
<td>eth 0/4</td>
<td>Desg</td>
<td>FWD</td>
<td>19</td>
<td>128.4</td>
<td>P2p</td>
</tr>
<tr>
<td>giga-eth 0/1</td>
<td>Desg</td>
<td>FWD</td>
<td>4</td>
<td>128.25</td>
<td>P2p</td>
</tr>
<tr>
<td>giga-eth 0/2</td>
<td>Desg</td>
<td>FWD</td>
<td>4</td>
<td>128.26</td>
<td>P2p</td>
</tr>
</tbody>
</table>
```
**show startup-config**

Use the `show startup-config` command to display a text printout of the startup configuration file stored in flash memory.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample output of the `show startup-config` command:

```
ProCurve>enable
ProCurve#show startup-config
!
!
no enable password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
!
ip policy-timeout tcp all-ports 600
ip policy-timeout udp all-ports 60
ip policy-timeout icmp 60
!
!
!
interface eth 0/1
speed auto
no ip address
shutdown
!
interface bri 1/2
shutdown
```
!  
!  
ip access-list standard MatchAll
    permit host 10.3.50.6  
    permit 10.200.5.0 0.0.0.255
!  
!  
ip access-list extended UnTrusted
    deny   icmp 10.5.60.0 0.0.0.255 any source-quench  
    deny   tcp any any
!  
no ip snmp agent
!  
!  
!
show startup-config checksum

Use the `show startup-config checksum` command to display the MD5 checksum of the unit’s startup configuration.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Functional Notes**

This command is used in conjunction with the `show running-config checksum` command to determine whether the configuration has changed since the last time it was saved.

**Usage Examples**

The following example displays the MD5 checksum of the unit’s startup configuration:

```
ProCurve#show startup-config checksum
10404D5DAB3FE35E307B6A79AC6AC8C0
ProCurve#
```

```
ProCurve#show running-config checksum
10404D5DAB3FE35E307B6A79AC6AC8C0
ProCurve#
```
show system

The **show system** command shows the system version, timing source, power source, and alarm relay status.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is sample output for the **show system** command:

ProCurve>**enable**
ProCurve#**show system**
show tacacs+ statistics

Use the show tacacs+ statistics command to display terminal access controller access control system (TACACS+) client statistics.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following is sample output for the show tacacs+ statistics command:

ProCurve>enable
ProCurve#show tacacs+ statistics

Packets sent: 0 0 0
Invalid responses: 0 0 0
Timeouts: 0 0 0
Average delay: 0ms 0ms 0ms
Maximum delay: 0ms 0ms 0ms
Socket Opens: 0
Socket Closes: 0
show tcp info [realtime] <control block>

Use the show tcp info command to display TCP control block information in the SROS. This information is for troubleshooting and debug purposes only. For more detailed information, you can optionally specify a particular TCP control block. When a particular TCP control block is specified, the system provides additional information regarding crypto map settings that the show tcp info command does not display.

Syntax Description

<table>
<thead>
<tr>
<th>realtime</th>
<th>Displays full-screen output in real-time. See the Functional Notes below for more information.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;control block&gt;</td>
<td>Optional. Specify a particular TCP control block for more detailed information.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Functional Notes

Use the realtime argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the terminal length command; refer to terminal length <lines> on page 280).

Usage Examples

The following is a sample from the show tcp info command:

ProCurve>enable
ProCurve#show tcp info
TCP TCB Entries
ID STATE LSTATE OSTATE TYPE FLAGS RPORT LPORT SWIN SRT INTERFACE
0 FREE FREE FREE SRVR 0 0 0 0 0 NONE
1 LISTEN FREE FREE CONN 0 0 21 0 0 NONE
2 LISTEN FREE FREE CONN 0 0 80 0 0 NONE
3 LISTEN FREE FREE CONN 0 0 23 0 0 NONE
4 LISTEN FREE FREE CONN 0 0 5761 0 0 NONE
5 FREE FREE FREE SRVR 0 0 0 0 0 NONE
etc.
**show tech [terminal]**

Use the `show tech` command to create a file `showtech.txt` located in flash memory that contains pertinent system information collected from various `show` commands. Instead of writing to a file, display the created file on the terminal screen using the `terminal` option.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>terminal</td>
<td>Optional. The terminal option displays the <code>showtech.txt</code> file output to the terminal screen real-time.</td>
</tr>
</tbody>
</table>

**Default Value**

No default value necessary for this command.

**Functional Notes**

The `show tech` command runs a system file that creates a `showtech.txt` file in flash memory that contains terminal displays from the following `show` commands:

- `show version`
- `show modules`
- `show flash`
- `show cflash`
- `show running-config verbose`
- `show interfaces`
- `show atm pvc`
- `show dial-backup interfaces`
- `show frame-relay lmi`
- `show frame-relay pvc`
- `show ip bgp neighbors`
- `show ip bgp summary`
- `show ip ospf neighbor`
- `show ip ospf summary-address`
- `show ip mroute`
- `show ip bridge`
- `show spanning-tree`
- `show ip interfaces`
- `show connections`
- `show arp`
- `show ip traffic`
- `show tcp info`
- `show ip protocols`
- `show ip route`
- `show ip access-lists`
- `show event-history`
- `show output-startup`
show processes cpu
show buffers
show buffers users
show memory heap
show debugging

Usage Examples

The following creates a showtech.txt file and displays it to the terminal screen:

ProCurve>enable
ProCurve#show tech

Opening and applying file.....
Done.

ProCurve#show file flash showtech.txt
Using 57693 bytes

ProCurve#!
ProCurve#!
ProCurve#!
ProCurve# show version
ProCurve Secure Router 7102dl
SROS Version: J03.01.01
  Checksum: 5C8D29BE, built on: Mon Jul 25 16:14:46 2005
Boot ROM version J03.01.01
  Checksum: 49C7, built on: Mon Jul 25 16:15:52 2005
Copyright (c) 2005-2005, Hewlett-Packard, Co.
Platform: ProCurve Secure Router 7102dl
Serial number US449TR019
Flash: 33554432 bytes  DRAM: 134217727 bytes

System uptime is 0 days, 0 hours, 14 minutes, 40 seconds

Current system image: "CFLASH:/SROS.BIZ"
Current configuration-file: "CFLASH:/startup-config"
Configured system image path:
Primary: "CFLASH:/SROS.BIZ"
Configured configuration-file path:
Primary: "CFLASH:/startup-config"
Backup: "NONVOL:/startup-config"
ProCurve#!
ProCurve#!
ProCurve# show modules

<table>
<thead>
<tr>
<th>Slot</th>
<th>Port</th>
<th>Type</th>
<th>Part Number</th>
<th>Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1–2</td>
<td>E1/E1 WAN</td>
<td>J8456A</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1–2</td>
<td>E1/E1 WAN</td>
<td>J8456A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1–8</td>
<td>Octal E1</td>
<td>J8463A</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>Empty</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

ProCurve#!
ProCurve#!
ProCurve# show flash
287413 J01_02B-boot.biz
3775 startup-config
5166 startup-config.bak

........ etc.
show thresholds

Use the show thresholds command to display thresholds currently crossed for all DS1 interfaces.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following is sample output of the show thresholds command.

ProCurve>enable
ProCurve#show thresholds

t1 1/1:
    SEFS 15 min threshold exceeded
    UAS 15 min threshold exceeded
    SEFS 24 hr threshold exceeded
    UAS 24 hr threshold exceeded

t1 1/2:
    No thresholds exceeded
**show udp info [realtime] <session id>**

Use the `show udp info` command to display User Datagram Protocol (UDP) session information.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>realtime</code></td>
<td>Displays full-screen output in real-time. See the <code>Functional Notes</code> below for more information.</td>
</tr>
<tr>
<td><code>&lt;session id&gt;</code></td>
<td>Specifies ID of session to display (Range: 0 to 31).</td>
</tr>
</tbody>
</table>

**Default Values**

No default necessary for this command.

**Functional Notes**

Use the `realtime` argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the `F` key) or exit realtime mode (by pressing `Ctrl-C`). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the `#terminal length` command).

**Usage Examples**

The following example shows sample output from the `show udp info` command:

```
ProCurve>enable
ProCurve#show udp info
UDP Session Entries

<table>
<thead>
<tr>
<th>ID</th>
<th>Local Port</th>
<th>IP Address</th>
<th>Socket</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>520</td>
<td>0.0.0.0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.0.0.0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>161</td>
<td>0.0.0.0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>127.0.0.1</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>0.0.0.0</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>127.0.0.1</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>127.0.0.1</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>127.0.0.1</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>127.0.0.1</td>
<td>19</td>
</tr>
</tbody>
</table>
```
show users [realtime]

Use the show users command to display the name (if any) and state of users authenticated by the system. Displayed information includes:

- Connection location (for remote connections this includes TCP information)
- Username of authenticated user
- Current state of the login (in process or logged in)
- Current enabled state
- Time the user has been idle on the connection

Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>realtime</td>
<td>Displays full-screen output in real-time. See the Functional Notes below for more information.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Functional Notes

Use the realtime argument for this command to display full-screen output in real-time. Information is continuously updated on the console until you either freeze the data (by pressing the F key) or exit realtime mode (by pressing Ctrl-C). If there is not enough room on the screen for all available data, the information will truncate at the bottom of the screen. In order to maximize the amount of data displayed, increase the terminal length (using the #terminal length command).

Usage Examples

The following is a sample of show users output:

ProCurve>enable
ProCurve#show users
- CONSOLE 0 'user' logged in and enabled
  Idle for 00:00:00
- TELNET 0 (172.22.12.60:3998) 'password-only' logged in (not enabled)
  Idle for 00:00:14
- FTP (172.22.12.60:3999) 'user' logged in (not enabled)
  Idle for 00:00:03
**show version**

Use the `show version` command to display the current SROS version information.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample `show version` output:

ProCurve>**enable**
ProCurve#**show version**

ProCurve Secure Router 7203dl
SROS Version: J04.01
  Checksum: 74305239, built on: Fri Dec 09 10:00:32 2005
Boot ROM version J04.01
  Checksum: 4353, built on: Fri Dec 09 10:00:35 2005
Copyright (c) 2005-2005, Hewlett-Packard, Co.
Platform: ProCurve Secure Router 7203dl
Serial number US449TS040
Flash: 33554432 bytes  DRAM: 268435455 bytes

System uptime is 0 days, 6 hours, 55 minutes, 24 seconds

Current system image file:   "CFLASH:/SROS.BIZ"
Current configuration-file: CFLASH:/startup-config"
Configured system image path:
  Primary:                  "CFLASH:/SROS.BIZ"
  Backup:                   "NONVOL:/SROS.BIZ"
Configured configuration-file path:
  Primary:                  "CFLASH:/startup-config"
  Backup:                   "NONVOL:/startup-config"

ProCurve#
show version [cflash | flash] <filename>

Use the show version cflash and show version flash commands to display the current SROS version information for a specified .biz file. When this command is entered, the system opens the .biz file specified and returns the current SROS version information.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cflash</td>
<td>Specifies a .biz file located in the compact flash memory.</td>
</tr>
<tr>
<td>flash</td>
<td>Specifies a .biz file located in flash memory.</td>
</tr>
<tr>
<td>&lt;filename&gt;</td>
<td>Specifies the exact filename of the .biz file for the system to determine the version information.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following is a sample show version cflash output:

ProCurve>enable
ProCurve#show version cflash SROS.BIZ

Version: 03.01.00
sip check-sync

Use the sip check-sync command to send a check-sync notification to all IP phones registered to the unit. When an IP phone receives this check-sync notification, the phone will check for possible configuration changes stored on the server.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Usage Examples

The following example notifies all IP phones to check for a change in configuration:

ProCurve>enable
ProCurve#sip check-sync
**telnet <address>**

Use the `telnet` command to open a Telnet session (through the SROS) to another system on the network.

**Syntax Description**

<address> Specifies the IP address of the remote system.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example opens a Telnet session to a remote system (10.200.4.15):

```
ProCurve>enable
ProCurve#telnet 10.200.4.15

User Access Login: 
Password: 
```
**terminal length <lines>**

The `terminal length` command sets the number of rows (lines) for a terminal session. This command is only valid for the current terminal session and returns to the default value (24 rows) when the session closes. Use the `no` form of this command to return to the default terminal length.

**Syntax Description**

| <lines> | Number of rows (lines) for the terminal session (Range: 0 to 480). |

**Note**

*Setting the terminal length to zero disables paging.*

**Default Values**

The default setting for this command is 24 rows.

**Usage Examples**

The following example sets the number of rows to 30.

```
ProCurve>enable
ProCurve#terminal length 30
```
**traceroute** `<address> source <address>`

Use the `traceroute` command to display the IP routes a packet takes to reach the specified destination.

**Syntax Description**

- `<address>` Optional. Specifies the IP address of the remote system to trace the routes to.
- `source <address>` Optional. Specifies the IP address of the interface to use as the source of the trace.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following is a sample `traceroute` output:

```
ProCurve>enable
ProCurve#traceroute 192.168.0.1

Type CTRL+C to abort.
Tracing route to 192.168.0.1 over a maximum of 30 hops

    1  22ms  20ms  20ms  192.168.0.65
    2  23ms  20ms  20ms  192.168.0.1

ProCurve#
```
**undebug all**

Use the `undebug all` command to disable all activated debug messages.

**Syntax Description**

No subcommands.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example disabled all activated debug messages:

```
ProCurve>enable
ProCurve#undebug all
```
wall <message>

Use the wall command to send messages to all users currently logged in to the SROS unit.

**Syntax Description**

No subcommands.

**Default Values**

No defaults necessary for this command.

**Usage Examples**

The following example sends the message “Reboot in 5 minutes if no objections” to the CLI screen of everyone currently connected:

ProCurve>enable
ProCurve#wall Reboot in 5 minutes if no objections
write [erase | memory | network | terminal]

Use the `write` command to save the running configuration to the unit’s flash memory or a Trivial File Transfer Protocol (TFTP) server. Also use the `write` command to clear flash memory or to display the running configuration on the terminal screen. Entering the `write` command with no other arguments copies your configuration changes to the unit’s flash memory. Once the save is complete, the changes are retained even if the unit is shut down or suffers a power outage.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>erase</td>
<td>Optional. Erase the configuration files saved to the unit’s flash memory.</td>
</tr>
<tr>
<td>memory</td>
<td>Optional. Save the current configuration to flash memory. See `copy running-config [cflash &lt;filename&gt;</td>
</tr>
<tr>
<td>network</td>
<td>Optional. Save the current configuration to the network TFTP server. See <code>copy tftp &lt;destination&gt;</code> on page 79 for more information.</td>
</tr>
<tr>
<td>terminal</td>
<td>Optional. Display the current configuration on the terminal screen.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example saves the current configuration to the unit’s flash memory:

```
ProCurve>enable
ProCurve#write memory
```
GLOBAL CONFIGURATION MODE COMMAND SET

To activate the Global Configuration mode, enter the `configuration` command at the Enable security mode prompt. For example:

ProCurve>`enable
ProCurve>`configure terminal
ProCurve>(config)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `aaa accounting commands` begin on page 288
- `aaa authentication commands` begin on page 291
- `aaa authorization commands` begin on page 295
- `aaa group server [radius | tacacs+] <listname>` on page 297
- `aaa on` on page 298
- `aaa processes <threads>` on page 300
- `arp <ip address> <mac address> arpa` on page 301
- `autosynch-mode` on page 302
- `banner [exec | login | motd] <character> <message> <character>` on page 304
- `boot config [cflash | flash] <filename> [cflash | flash] <backup filename>` on page 305
- `boot system [cflash | flash] <filename> [no-backup | [cflash | flash] <backup filename>] [verify]` on page 306
- `bridge <group#> protocol ieee` on page 307
- `clock [auto-correct-dst | no-auto-correct-dst]` on page 308
- `clock set <time> <day> <month> <year>` on page 309
- `clock timezone <text>` on page 310
- `crypto commands` begin on page 312
- `data-call [authentication protocol | sent authentication protocol] [chap | pap]` on page 328
- `data-call [mtu <number> | multilink]` on page 329
- `enable password [md5] <password>` on page 330
- `event-history` on page 331
event-history priority [error | fatal | info | notice | warning] on page 332
exception report [file-name <filename>] on page 335
ftp authentication <listname> on page 336
hostname <name> on page 337
interface commands begin on page 338
ip access-list commands begin on page 350
ip as-path-list <listname> on page 356
ip classless on page 357
ip community-list <listname> on page 358
ip crypto on page 359
ip default-gateway <ip address> on page 360
ip dhcp-server commands begin on page 361
ip domain commands begin on page 366
ip firewall commands begin on page 369
ip forward-protocol udp <port number> on page 387
ip ftp commands begin on page 389
ip host <name> <address> on page 392
ip http [access-class <listname> in | authentication <listname> | secure-access-class <listname> in | secure-server <TCP port> | server <TCP port> | session-limit <limit> | session-timeout <time>] on page 393
ip igmp join <group-address> on page 394
ip load-sharing [per-destination | per-packet] on page 395
ip local policy route-map <map-name> on page 396
ip mcast-stub helper-address <ip address> on page 397
ip multicast-routing on page 398
ip name-server <server-address1-6> on page 399
ip policy-class <policyname> on page 400
ip policy-timeout <protocol> <range> <port> <seconds> on page 407
ip prefix-list <listname> description <"text"> on page 409
ip prefix-list <listname> seq <sequence#> [permit | deny] <network/len> [le <le-value> | ge <ge-value>] on page 410
ip radius source-interface <interface> on page 411
ip route <ip address> <subnet mask> [<interface or ip address> | null 0] <administrative distance> on page 412
ip routing on page 413
ip rtp firewall-traversal [policy-timeout <seconds>] on page 414
ip scp server on page 415
ip sip commands begin on page 416
ip snmp agent on page 420
ip snmp source-interface <interface> on page 421
ip [ssh-server <port> | telnet-server <port>] on page 422
ip subnet-zero on page 423
ip tftp server [access-class <access-class> in] on page 425
ip tftp source-interface <interface> on page 426
isdn-group on page 427
isdn-number-template on page 428
line [console | telnet | ssh] <line-number> <ending number> on page 430
lldp [minimum-transmit-interval l reinitialization-delay l transmit-interval l ttl-multiplier] <numeric value> on page 432
logging commands begin on page 433
mac address-table aging-time <aging time> on page 446
mac address-table static <mac address> bridge <bridge id> interface <interface> on page 447
modem countrycode [<countrycode>] on page 448
qos map <mapname> <sequence number> on page 451
radius-server on page 452
radius-server host on page 454
route-map <map-name> [ permit | deny ] <sequence number> on page 455
router bgp <AS number> on page 456
router ospf on page 457
router pim-sparse on page 458
router rip on page 459
safe-mode <reload timer> <threshold time> on page 461
service password-encryption on page 462
snmp-server commands begin on page 463
snmp retry-timeout <time> on page 474
snmp server <address or hostname> version <1-3> on page 475
snmp wait-time <time> on page 476
spanning-tree commands begin on page 477
tacacs-server on page 486
thresholds [BES | CSS | DM | ES | LCV | LES | PCV | SEFS | SES | UAS] [15Min | 24Hr] <threshold count> on page 487
username <username> password <password> on page 489
aaa accounting commands <level> [<listname> | default] [none | stop-only] [group <groupname> | group tacacs+]

Use **aaa accounting commands** to set parameters for AAA accounting. For more detailed information on AAA functionality, refer to the *Technology Review* section of the command *aaa on* on page 298.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;level&gt;</td>
<td>Specifies the commands enable level. (1=unprivileged, 15 = privileged).</td>
</tr>
<tr>
<td>&lt;listname&gt;</td>
<td>Specifies the name of the list.</td>
</tr>
<tr>
<td>default</td>
<td>Uses the default accounting list.</td>
</tr>
<tr>
<td>none</td>
<td>Disables accounting.</td>
</tr>
<tr>
<td>stop-only</td>
<td>Records stop-only when service terminates.</td>
</tr>
<tr>
<td>group &lt;groupname&gt;</td>
<td>Uses the specified group of remote servers for accounting.</td>
</tr>
<tr>
<td>group tacacs+</td>
<td>Uses the TACACS+ server for accounting.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, accounting is not enabled.

**Usage Examples**

The following example creates a list called myList and sets accounting for Level 1 commands at stop-only activities:

```
ProCurve(config)#aaa accounting commands 1 myList stop-only group tacacs+
```

**Note**

*To complete this command, Telnet must be applied to the lines. See Line (Telnet) Interface Config Command Set on page 503 for more detailed instructions.*
aaa accounting [suppress null-username]

Use the **aaa accounting suppress null-username** command to stop sending accounting records for usernames set to null. For more detailed information on AAA functionality, refer to the *Technology Review* section of the command **aaa on** on page 298.

**Syntax Description**

| suppress | Refrain from sending accounting records for null usernames. |

**Default Values**

By default, this command is disabled, which means the accounting records for null usernames are sent to the server.

**Usage Examples**

The following command causes the unit to refrain from sending accounting records for users with null usernames:

```
ProCurve(config)#aaa accounting suppress null-username
```
aaa accounting update [newinfo | periodic <minutes>]

Use the aaa accounting update command to specify when accounting records are sent to the server. For more detailed information on AAA functionality, refer to the Technology Review section of the command aaa on on page 298.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newinfo</td>
<td>Sends all new accounting records immediately.</td>
</tr>
<tr>
<td>periodic &lt;minutes&gt;</td>
<td>Periodically sends all accounting records to the server.</td>
</tr>
</tbody>
</table>

Default Values

By default, accounting records are sent every 5 minutes.

Usage Examples

The following command sets the unit to send accounting records every 600 minutes to the server:

ProCurve(config)#aaa accounting update periodic 600
aaa authentication [banner | fail-message | password-prompt | username-prompt] <string>

Use the `aaa authentication` command to control various features of the AAA subsystem authentication process. For more detailed information on AAA functionality, refer to the Technology Review section of the command `aaa on` on page 298.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>banner &lt;string&gt;</strong></td>
<td>Sets the banner shown before user authentication is attempted. The banner can be multiple lines. Enter a delimiter (such as :) to begin recording the typed text message used for the banner. The message must end with the same delimiter to indicate that the message is complete. The text delimiters are not displayed to the screen during operation.</td>
</tr>
<tr>
<td><strong>fail-message &lt;string&gt;</strong></td>
<td>Sets the message shown if user authentication fails. The message can be multiple lines. Enter a delimiter (such as :) to begin recording the typed text message displayed after a failed authentication attempt. The message must end with the same delimiter to indicate that the message is complete. The text delimiters are not displayed to the screen during operation.</td>
</tr>
<tr>
<td><strong>password-prompt &lt;string&gt;</strong></td>
<td>Sets the prompt for the user's password. The prompt is a single line. Enclose the string in quotation marks.</td>
</tr>
<tr>
<td><strong>username-prompt &lt;string&gt;</strong></td>
<td>Sets the prompt for the user's name. The prompt is a single line. Enclose the string in quotation marks.</td>
</tr>
</tbody>
</table>

### Default Values

- **banner**: User Access Verification
- **fail-message**: Authentication Failed
- **password-prompt**: Password:
- **username-prompt**: Username:

### Usage Examples

The following example defines a banner of “Welcome to the ProCurve Secure Router”:

```
ProCurve(config)#aaa authentication banner :
Enter TEXT message. End with the character ':'.
Welcome to the ProCurve Secure Router:.
```

The following example defines an authentication failed message of “Authentication Failed. Contact IT for further assistance.”:

```
ProCurve(config)#aaa authentication fail-message :
Enter TEXT message. End with the character ':'.
Authentication Failed. Contact IT for further assistance:.
```
The following example defines a password prompt of “PW:”:

ProCurve(config)#aaa authentication password-prompt “PW:”
ProCurve(config)#

The following example defines a Username prompt of “User:”:

ProCurve(config)#aaa authentication username-prompt “User:”
ProCurve(config)#
aaa authentication enable default [none | line | enable | group <groupname> | group radius | group tacacs+]

Use the `aaa authentication enable default` command to create (or change) the list of fallback methods used for privileged mode access authentication. For more detailed information on AAA functionality, refer to the Technology Review section of the command `aaa on` on page 298.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Access automatically granted.</td>
</tr>
<tr>
<td>line</td>
<td>Uses the line password for authentication.</td>
</tr>
<tr>
<td>enable</td>
<td>Uses the enable password for authentication.</td>
</tr>
<tr>
<td>group &lt;groupname&gt;</td>
<td>Uses the specified group of remote servers for authentication.</td>
</tr>
<tr>
<td>group radius</td>
<td>Uses all defined RADIUS servers for authentication.</td>
</tr>
<tr>
<td>group tacacs+</td>
<td>Uses all defined TACACS+ servers for authentication.</td>
</tr>
</tbody>
</table>

**Default Values**

If there is no default methods list configured, the default behavior is to use the enable password for the unit. If there is no password configured, consoles are allowed access (this prevents a lock-out).

**Functional Notes**

A user is authenticated by trying the list of methods from first to last until a method succeeds or fails. If a method is unable to complete, the next method is tried. The group falls through if the servers in the remote group could not be found.

Note that enable access is a password-only process. The local user database cannot be used and the username given to any remote RADIUS server is `$enab15$`. The only list name allowed is `default`.

**Usage Examples**

The following example specifies using the line password as the first method for enable authentication and using the enable password as the second:

```
ProCurve(config)#aaa authentication enable default line
```
aaa authentication login [<listname> | default] [none | line | enable | local | group <groupname> | group radius | group tacacs+]

Use the `aaa authentication login` command to create (or change) a named list with the ability to have a chain of fallback authentication methods for user authentication. Available methods for the fallback authentication methods are: no authentication (which grants login access without authentication), line password, enable password, local database, and defined group of servers. The defined server groups may be TACACS+ or RADIUS servers. Use the `no` form of this command to remove a configured login. For more detailed information on AAA functionality, refer to the `Technology Review` section of the command `aaa` on page 298.

Syntax Description

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;listname&gt;</code></td>
<td>Specifies a named login list.</td>
</tr>
<tr>
<td>default</td>
<td>Specifies the default list used to authenticate users when no other list is assigned.</td>
</tr>
<tr>
<td>none</td>
<td>Access automatically granted.</td>
</tr>
<tr>
<td>line</td>
<td>Uses line password (Telnet 0-4 or console 0-1) for authentication.</td>
</tr>
<tr>
<td>enable</td>
<td>Uses enable password for authentication.</td>
</tr>
<tr>
<td>local</td>
<td>Uses local user database for authentication.</td>
</tr>
<tr>
<td>group &lt;groupname&gt;</td>
<td>Uses specified group of remote servers for authentication.</td>
</tr>
<tr>
<td>group radius</td>
<td>Uses defined RADIUS servers for authentication.</td>
</tr>
<tr>
<td>group tacacs+</td>
<td>Uses defined TACACS+ servers for authentication.</td>
</tr>
</tbody>
</table>

Default Values

The login list named `default` is the default list used to authenticate users when no other list is assigned.

Functional Notes

A user is authenticated by trying the list of methods from first to last until a method succeeds or fails. If a method is unable to complete, the next method is tried. The local user database falls through to the next method if the username does not appear in the database. The group falls through if the servers in the remote group could not be found. See the command `radius-server` on page 452 or `tacacs-server` on page 486 for information on defining server groups.

Usage Examples

The following example creates a named list called `myList` and specifies using the local database as the first method, `myGroup` as the second method, and line password as the third method for login authentication:

```
ProCurve(config)#aaa authentication login myList local group myGroup line
```

The following command sets the default authentication list for logins to use the local database as the first fallback method:

```
ProCurve(config)#aaa authentication login default local
```
aaa authorization commands <level> [<listname> | default] [group <groupname> | group tacacs+ | if-authenticated | none]

Use aaa authorization commands to create (or change) a list of methods for user authorization. Use the no form of this command to remove a configured list of authorization methods. For more detailed information on AAA functionality, refer to the Technology Review section of the command aaa on on page 298.

Syntax Description

<level> Specifies the commands enable level. (1=unprivileged, 15 = privileged).
$listname> Specifies the name of the authorization list.
default Specifies the default authorization list and applies it implicitly across all lines.
group <groupname> Uses the specified group of remote servers for authorization.
group tacacs+ Uses all defined TACACS+ servers for authorization.
if-authenticated Succeeds if user has authenticated.
none Access automatically granted.

Default Values

The authorization list named default is the default list used to authorize commands when no other list is assigned to the line.

Usage Examples

The following command creates a list called myList to authorize unprivileged commands (which succeeds only if the user has been authenticated successfully):

ProCurve(config)#aaa authorization commands 1 myList if-authenticated

The following command uses the default list to authorize privileged (level 15) commands against the defined TACACS+ servers:

ProCurve(config)#aaa authorization commands 15 default group tacacs+
aaa authorization [config-command | console]

Use the **aaa authorization** to enable or disable authorization for configuration mode commands and for console mode. Use the **no** form of this command to return to the default setting. For more detailed information on AAA functionality, refer to the *Technology Review* section of the command `aaa on` on page 298.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config-command</td>
<td>Enables authorization for configuration mode commands. Only level 1 (unprivileged) and level 15 (privileged) commands are supported.</td>
</tr>
<tr>
<td>console</td>
<td>Allows authorization to be applied to the console.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, authorization for console is disabled. However, configuration mode commands are authorized by default.

**Usage Examples**

The following example enables authorization of configuration mode commands:

```
ProCurve(config)#aaa authorization config-command
```

The following example enables authorization of console commands:

```
ProCurve(config)#aaa authorization console
```
aaa group server [radius | tacacs+] <listname>

Use the `aaa group server` command to group pre-defined RADIUS and TACACS+ servers into named lists. Use the `no` form of this command to remove a configured server group. For more detailed information on AAA functionality, refer to the Technology Review section of the command `aaa on` on page 298.

**Syntax Description**

- `radius`: Groups defined RADIUS servers.
- `tacacs+`: Groups TACACS+ server.
- `<listname>`: Specifies the name of the list.

**Default Values**

No default value necessary for this command.

**Functional Notes**

Use the `radius-server` command to specify RADIUS servers before adding them to a group. Likewise, use the `tacacs-server` command to specify TACACS+ servers before adding them to a group. These commands enter a mode for adding individual servers to the named group. Refer to Radius Group Command Set on page 1298 or TACACS+ Group Configuration Command Set on page 1300 for more information.

The default group cannot be changed and includes all RADIUS servers in the order they were specified by the `radius-server` commands. The same is true of TACACS+ servers specified by the `tacacs-server` commands.

**Usage Examples**

The following example creates the named list `myServers` and enters the RADIUS group:

```
ProCurve(config)#aaa group server radius myServers
ProCurve(config-sg-radius)#
```

The following example creates the named list `myServers` and enters the TACACS+ group:

```
ProCurve(config)#aaa group server tacacs myServers
ProCurve(config-sg-tacacs+)#
```
aaa on

Use the aaa on command to activate the AAA subsystem. Use the no form of this command to deactivate AAA.

Syntax Description

No subcommands.

Default Values

By default, AAA is not activated.

Functional Notes

By default, the AAA subsystem is turned off and authentication follows the line technique (local, line, etc.). Once activated, the AAA lists override the methods specified in the line command.

Technology Review

AAA stands for authentication, authorization, and accounting. The SROS AAA subsystem currently supports authentication. Authentication is the means by which a user is granted access to the device (router). For instance, a username/password is authenticated before the user can use the CLI. VPN clients can also verify username/password before getting access through the device.

There are several methods that can be used to authenticate a user:

- **NONE** Instant access
- **LINE-PASSWORD** Use the line password (telnet 0-4 or console 0-1)
- **ENABLE-PASSWORD** Use the enable password
- **LOCAL-USERS** Use the local user database
- **GROUP <groupname>** Use a group of remote RADIUS or TACACS+ servers

The AAA system allows the user to create a named list of these methods to try in order (in case one fails, it falls to the next one). This named list is then attached to a portal (telnet 0-4 or console 0-1). When a user telnets in or accesses the terminal, the AAA system uses the methods from the named list to authenticate the user.

The AAA system must be turned on to be active. By default it is off. Use the aaa on command to activate the AAA system.

If a portal is not explicitly assigned a named list, the name default is automatically assigned to it. The user can customize the default list just like any other list. If no default list is configured, the following default behavior applies (defaults are based on portal):

- Instant access (NONE) is assigned to the CONSOLE using the default list (when the list has not been configured).
- The local user database is used for TELNETS using the default list (when the list has not been configured).
- No access is granted for FTP access using the default list (when the list has not been configured).
Methods fail (and therefore cause the system to proceed to the next configured method) under circumstances such as the following:

- LINE and ENABLE passwords fall through if there is no LINE or ENABLE password configured.
- LOCAL USERS fall through if the given user is not in the database.
- RADIUS or TACACS+ servers fall through if the given server(s) cannot be contacted on the network.

Example

For a default list defined with the order [LINE, ENABLE, LOCAL, and GROUP mygroup], the following statements are true:

- If there is no LINE password, the list falls through to the ENABLE password.
- If there is no ENABLE password, the AAA system prompts the user for a username and password for the local user database.
- If the given user is not in the local list, the username and password are handed to the remote servers defined in mygroup.
- A failure at any point (password not matching) denies access.

If the AAA process falls through the list completely, system behavior is based on portal:

- CONSOLE access is granted if the process falls completely through (this prevents a lock-out condition).
- TELNET and FTP are denied access.

Usage Examples

The following example activates the AAA subsystem:

ProCurve(config)#aaa on
aaa processes <threads>

Use the aaa processes command to set the number of threads available to the AAA subsystem. Use the no form of this command to return to the default setting. For more detailed information on AAA functionality, refer to the Technology Review section of the command aaa on on page 298.

Syntax Description

<threads> Specifies the number of threads available to the AAA subsystem. Range: 1 to 64.

Default Values

By default, this is set to 1 process.

Functional Notes

Increasing this number may speed up simultaneous authentication at the cost of system resources (e.g., memory).

Usage Examples

The following example specifies five available threads for the AAA subsystem:

ProCurve(config)#aaa processes 5
**arp** `<ip address> <mac address> arpa`

Use this command to enter static entries into the address resolution protocol (ARP) table.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>arpa</strong></td>
<td>Sets the standard address resolution protocol for this interface.</td>
</tr>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Specifies the IP address.</td>
</tr>
<tr>
<td><code>&lt;mac address&gt;</code></td>
<td>Specifies the MAC address.</td>
</tr>
</tbody>
</table>

**Default Values**

The default for this command is **arpa**.

**Usage Examples**

The following example enables standard ARP for the VLAN interface:

```
ProCurve(config)#interface vlan 1
ProCurve(config-interface-vlan 1)#arp 192.168.22.253 00:12:79:00:00:01 arpa
```
**autosynch-mode**

Use the `autosynch` command to configure the SROS to synchronize the `startup-config` and `SROS.BIZ` files located in the system flash memory and the compact flash card. Use the `no` form of this command to disable this feature.

**Syntax Description**

No subcommands.

**Default Values**

The AutoSynch™ feature is disabled by default.

**Functional Notes**

The AutoSynch™ feature configures the system to synchronize the `startup-config` and `SROS.BIZ` files located in the system flash memory and the compact flash card. When enabled, the system compares the two files in the two locations and replaces the files located in the system flash memory with the ones from the compact flash card (regardless of which set of files is more current). This allows the customer to maintain the version of the operating system, and the configuration for that operating system, at the desired level. To accomplish this, a synchronization check is performed on the system any time there is a change in `startup-config` or `SROS.BIZ` on the compact flash card.

The autosynch feature allows for quick installation and updates of routers by inserting a compact flash card containing the desired software (must be renamed from the desired operating system software, such as J03.01.biz to `SROS.BIZ`) and startup configuration file (must be named `startup-config`) into a router with AutoSynch™ enabled. The ProCurve Secure Routers automatically boot from the compact flash (and secondarily from internal flash). After booting, with AutoSynch™ enabled, the router will synchronize the files in system flash memory with the desired files from compact flash.

**Caution**

Deleting the `SROS.BIZ` and `startup-config` files from the compact flash card (using the `erase` command) deletes the files from the system flash memory as well.
Functional Notes

Status commands associated with the AutoSynch™ feature include `show version flash SROS.BIZ` and `show autosynch-status`.

The `show version flash SROS.BIZ` command opens the specified .biz file and returns the current SROS version information.

```
ProCurve>enable
#show version flash SROS.BIZ

Version:J03.01.00
```

The `show autosynch-status` command displays the current AutoSynch™ configuration and the statistics for the `SROS.BIZ` and `startup-config` files (if AutoSynch™ is enabled).

```
ProCurve>enable
#show autosynch-status

AutoSynch: Mode - Enabled
AutoSynch: SROS.BIZ synched
AutoSynch: startup-config synched
```

Usage Examples

The following example enables the AutoSynch™ feature:

```
ProCurve(config)#autosynch-mode
```
Use the `banner` command to specify messages to be displayed in certain situations. Use the `no` form of this command to delete a previously configured banner.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exec</code></td>
<td>Creates a message to be displayed when any exec-level process takes place.</td>
</tr>
<tr>
<td><code>login</code></td>
<td>Creates a message to be displayed before the username and password login prompts.</td>
</tr>
<tr>
<td><code>motd</code></td>
<td>Creates a message-of-the-day (MOTD) banner.</td>
</tr>
<tr>
<td><code>&lt;character&gt;</code></td>
<td>Banner text delimiter character. Press <code>Enter</code> after the delimiter to begin input of banner text.</td>
</tr>
<tr>
<td><code>&lt;message&gt;</code></td>
<td>Specifies the text message you wish to display. End with the character that you chose as your delimiter.</td>
</tr>
</tbody>
</table>

### Default Values

By default, no banners are configured.

### Functional Notes

Banners appear in the following order (if configured):

- MOTD banner appears at initial connection.
- Login banner follows the MOTD banner.
- Exec banner appears after successful log in.

### Usage Examples

The following example configures the system to display a message of the day:

```
ProCurve(config)#banner motd "The system will be shut down today from 7PM to 11PM"
```
boot config [cflash | flash] <filename> [cflash | flash] <backup filename>

Use the boot config command to modify system boot parameters.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cflash</td>
<td>Specifies primary/backup configuration file located in compact flash memory.</td>
</tr>
<tr>
<td>flash</td>
<td>Specifies primary/backup configuration file located in flash memory.</td>
</tr>
<tr>
<td>&lt;filename&gt;</td>
<td>Specifies the filename of the configuration file (filenames are case-sensitive).</td>
</tr>
<tr>
<td>&lt;backup filename&gt;</td>
<td>Specifies a name for the backup configuration file.</td>
</tr>
</tbody>
</table>

Default Values

No default is necessary for this command.

Usage Examples

The following example specifies the file myconfig, located in flash memory, as the system boot file:

ProCurve(config)#boot config flash myconfig
**boot system [cflash | flash] <filename> [no-backup | [cflash | flash] <backup filename>] [verify]**

Use the `boot config` command to specify the system software loaded at startup.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cflash</td>
<td>Specifies primary/backup file located in compact flash memory.</td>
</tr>
<tr>
<td>flash</td>
<td>Specifies primary/backup file located in flash memory.</td>
</tr>
<tr>
<td>&lt;filename&gt;</td>
<td>Specifies the filename of the software (filenames are case-sensitive) - software files should have a .biz or .BIZ extension.</td>
</tr>
<tr>
<td>no-backup</td>
<td>Specifies that no backup software is to be saved to the system.</td>
</tr>
<tr>
<td>&lt;backup filename&gt;</td>
<td>Specifies a name for the backup software.</td>
</tr>
<tr>
<td>verify</td>
<td>Specifies a verification of the software checksum.</td>
</tr>
</tbody>
</table>

**Default Values**

No default is necessary for this command.

**Functional Notes**

Detailed instructions for upgrading the SROS and loading files into memory are found on your ProCurve SROS Documentation CD.

**Usage Examples**

The following example specifies the file `SROS.BIZ`, stored in flash memory, as the startup software:

```
ProCurve(config)#boot system flash SROS.BIZ
```
bridge <group#> protocol ieee

The **bridge protocol ieee** command configures a bridge group for the IEEE Spanning Tree Protocol. Use the **no** form of this command (with the appropriate arguments) to delete this setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group#&gt;</td>
<td>Specifies bridge group number (1 to 255) (using the bridge command).</td>
</tr>
<tr>
<td>ieee</td>
<td>IEEE 802.1 Ethernet spanning-tree protocol.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all configured bridge interfaces implement **ieee** spanning-tree protocol.

**Usage Examples**

The following example deletes the bridge protocol setting for bridge-group 17:

```
ProCurve(config)#no bridge 17 protocol ieee
```
clock [auto-correct-dst | no-auto-correct-dst]

The clock auto-correct-dst command allows the unit to automatically correct for Daylight Saving Time (DST). Use the clock no-auto-correct-dst command to disable this feature.

Syntax Description

<table>
<thead>
<tr>
<th>auto-correct-DST</th>
<th>Configures the unit to automatically correct for DST.</th>
</tr>
</thead>
<tbody>
<tr>
<td>no-auto-correct-DST</td>
<td>Disables DST correction.</td>
</tr>
</tbody>
</table>

Default Values

By default DST correction takes place automatically.

Functional Notes

Depending on the clock timezone chosen (see clock timezone <text> on page 310 for more information) one-hour DST correction may be enabled automatically. You may override this default using this command.

Usage Examples

The following example allows for automatic DST correction:

ProCurve(config)#clock auto-correct-dst

The following example overrides the one-hour offset for DST:

ProCurve(config)#clock no-auto-correct-dst
clock set <time> <day> <month> <year>

Use the **clock set** command to configure the system software clock. For the command to be valid, all fields must be entered. Refer to the **Usage Example** below for an example.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;time&gt;</td>
<td>Sets the time (in 24-hour format) of the system software clock in the format HH:MM:SS (hours:minutes:seconds).</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>Sets the current day of the month (valid range: 1 to 31).</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>Sets the current month (valid range: January to December). You need only enter enough characters to make the entry unique. This entry is not case-sensitive.</td>
</tr>
<tr>
<td>&lt;year&gt;</td>
<td>Sets the current year (valid range: 2000 to 2100).</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example sets the system software clock for 3:42 pm, April 22 2006:

```
ProCurve(config)#clock set 15:42:00 22 Apr 2006
```
The **clock timezone** command sets the unit’s internal clock to the timezone of your choice. This setting is based on the difference in time (in hours) between Greenwich Mean Time (GMT) or Central Standard Time (CST) and the timezone for which you are setting up the unit. Use the **no** form of this command to disable this feature.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;text&gt;</code></td>
<td>Specifies the difference in time (in hours) between Greenwich Mean Time (GMT) or Central Standard Time (CST) and the timezone for which you are setting up the unit.</td>
</tr>
</tbody>
</table>

### Default Values

No default value is necessary for this command.

### Functional Notes

The following list shows sample cities and their timezone codes.

- clock timezone +1-Amsterdam
- clock timezone +1-Belgrade
- clock timezone +1-Brussels
- clock timezone +1-Sarajevo
- clock timezone +1-West-Africa
- clock timezone +10-Brisbane
- clock timezone +10-Canberra
- clock timezone +10-Guam
- clock timezone +10-Hobart
- clock timezone +10-Vladivostok
- clock timezone +11
- clock timezone +12-Auckland
- clock timezone +12-Fiji
- clock timezone +13
- clock timezone +2-Athens
- clock timezone +2-Bucharest
- clock timezone +2-Cairo
- clock timezone +2-Harare
- clock timezone +8-Beijing
- clock timezone +8-Irkutsk
- clock timezone +8-Kuala-Lumpur
- clock timezone +8-Perth
- clock timezone +8-Taipei
- clock timezone +9-Adelaide
- clock timezone +9:30-Adelaide
- clock timezone +9:30-Darwin
- clock timezone +9-Beijing
- clock timezone +9:30-Perth
- clock timezone +9:30-Taipei
- clock timezone +9-Beijing
- clock timezone +9:30-Perth
- clock timezone +9:30-Taipei
- clock timezone +9:30-Beijing
- clock timezone +9:30:30-Perth
- clock timezone +9:30:30-Taipei
- clock timezone +9:30:30-Beijing
- clock timezone +9:30:30-Perth
- clock timezone +9:30:30-Taipei
- clock timezone +10-Hobart
- clock timezone +10-Vladivostok
- clock timezone +11
- clock timezone +12-Auckland
- clock timezone +12-Fiji
- clock timezone +13
- clock timezone +2-Athens
- clock timezone +2-Bucharest
- clock timezone +2-Cairo
- clock timezone +2-Harare
- clock timezone +3-Brasilia
- clock timezone +3-Buenos-Aires
- clock timezone +3:30-Brasilia
- clock timezone +3:30-Buenos-Aires
- clock timezone +3:30:30-Brasilia
- clock timezone +3:30:30-Buenos-Aires
- clock timezone +4-Brasilia
- clock timezone +4:30-Brasilia
- clock timezone +4:30-Buenos-Aires
- clock timezone +5-Brasilia
- clock timezone +5:30-Brasilia
- clock timezone +5:30-Buenos-Aires
- clock timezone +5:30-Brasilia
- clock timezone +6:30-Brasilia
- clock timezone +6:30-Buenos-Aires
- clock timezone +6:30-Brasilia
- clock timezone +7-Brasilia
- clock timezone +7:30-Brasilia
- clock timezone +7:30-Buenos-Aires
- clock timezone +7:30-Brasilia
- clock timezone +8-Brasilia
- clock timezone +8:30-Brasilia
- clock timezone +8:30-Buenos-Aires
- clock timezone +8:30-Brasilia
- clock timezone +9-Brasilia
- clock timezone +9:30-Brasilia
- clock timezone +9:30-Buenos-Aires
- clock timezone +9:30-Brasilia
- clock timezone +10-Brasilia
- clock timezone +10:30-Brasilia
- clock timezone +10:30-Buenos-Aires
- clock timezone +10:30-Brasilia
- clock timezone +11-Brasilia
- clock timezone +11:30-Brasilia
- clock timezone +11:30-Buenos-Aires
- clock timezone +11:30-Brasilia
- clock timezone +12-Brasilia
- clock timezone +12:30-Brasilia
- clock timezone +12:30-Buenos-Aires
- clock timezone +12:30-Brasilia
- clock timezone +13-Brasilia
- clock timezone +13:30-Brasilia
- clock timezone +13:30-Buenos-Aires
- clock timezone +13:30-Brasilia
- clock timezone +14-Brasilia
- clock timezone +14:30-Brasilia
- clock timezone +14:30-Buenos-Aires
- clock timezone +14:30-Brasilia
- clock timezone +15-Brasilia
- clock timezone +15:30-Brasilia
- clock timezone +15:30-Buenos-Aires
- clock timezone +15:30-Brasilia
- clock timezone +16-Brasilia
- clock timezone +16:30-Brasilia
- clock timezone +16:30-Buenos-Aires
- clock timezone +16:30-Brasilia
- clock timezone +17-Brasilia
- clock timezone +17:30-Brasilia
- clock timezone +17:30-Buenos-Aires
- clock timezone +17:30-Brasilia
- clock timezone +18-Brasilia
- clock timezone +18:30-Brasilia
- clock timezone +18:30-Buenos-Aires
- clock timezone +18:30-Brasilia
- clock timezone +19-Brasilia
- clock timezone +19:30-Brasilia
- clock timezone +19:30-Buenos-Aires
- clock timezone +19:30-Brasilia
- clock timezone +20-Brasilia
- clock timezone +20:30-Brasilia
- clock timezone +20:30-Buenos-Aires
- clock timezone +20:30-Brasilia
- clock timezone +21-Brasilia
- clock timezone +21:30-Brasilia
- clock timezone +21:30-Buenos-Aires
- clock timezone +21:30-Brasilia
- clock timezone +22-Brasilia
- clock timezone +22:30-Brasilia
- clock timezone +22:30-Buenos-Aires
- clock timezone +22:30-Brasilia
- clock timezone +23-Brasilia
- clock timezone +23:30-Brasilia
- clock timezone +23:30-Buenos-Aires
- clock timezone +23:30-Brasilia

*Note: Depending on the **clock timezone** chosen, one-hour Daylight Savings Time (DST) correction may be enabled automatically. See clock [auto-correct-dst | no-auto-correct-dst] on page 308 for more information.*
### Functional Notes (cont’d)

<table>
<thead>
<tr>
<th>Clock Timezone</th>
<th>Clock Timezone</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2-Helsinki</td>
<td>-3-Greenland</td>
</tr>
<tr>
<td>+2-Jerusalem</td>
<td>-3:30</td>
</tr>
<tr>
<td>+3-Baghdad</td>
<td>-4-Atlantic-Time</td>
</tr>
<tr>
<td>+3-Kuwait</td>
<td>-4-Caracas</td>
</tr>
<tr>
<td>+3-Moscow</td>
<td>-4-Santiago</td>
</tr>
<tr>
<td>+3-Nairobi</td>
<td>-5</td>
</tr>
<tr>
<td>+3:30</td>
<td>-5-Bogota</td>
</tr>
<tr>
<td>+4-Abu-Dhabi</td>
<td>-5-Eastern-Time</td>
</tr>
<tr>
<td>+4-Baku</td>
<td>-6-Central-America</td>
</tr>
<tr>
<td>+4:30</td>
<td>-6-Central-Time</td>
</tr>
<tr>
<td>+5-Ekaterinburg</td>
<td>-6-Mexico-City</td>
</tr>
<tr>
<td>+5-Islamabad</td>
<td>-6-Saskatchewan</td>
</tr>
<tr>
<td>+5:30</td>
<td>-7-Arizona</td>
</tr>
<tr>
<td>+5:45</td>
<td>-7-Mountain-Time</td>
</tr>
<tr>
<td>+6-Almaty</td>
<td>-8</td>
</tr>
<tr>
<td>+6-Astana</td>
<td>-9</td>
</tr>
<tr>
<td>+6-Sri-Jay</td>
<td>0-(UTC) Universal Coordinated Time</td>
</tr>
<tr>
<td>+6:30</td>
<td>GMT-Casablanca</td>
</tr>
<tr>
<td>+7-Bangkok</td>
<td>GMT-Dublin</td>
</tr>
<tr>
<td>+7-Kranoyarsk</td>
<td></td>
</tr>
</tbody>
</table>

### Usage Examples

The following example sets the timezone for Santiago, Chile.

ProCurve>enable
ProCurve#clock timezone -4-Santiago
crypto ca authenticate <name>

Use the `crypto ca authenticate` command to initiate CA authentication procedures.

**Syntax Description**

```
<name>
```

Specifies CA profile by alphanumeric string of up to 32 characters.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

The type of authentication procedure is based on the `enrollment` command and its settings. See `enrollment terminal` on page 1124 and `enrollment url <url>` on page 1125 for more information. When `enrollment` is set to `terminal`, the CA authentication process is done manually, as shown in the example which follows (see **Usage Examples** for this command).

**Usage Examples**

The following example initiates the CA authentication process:

```
ProCurve(config)#crypto ca authenticate testCAprofile
```

Enter the base 64 encoded CA certificate. End with two consecutive carriage returns or the word "quit" on a line by itself:

```
-----BEGIN X509 CERTIFICATE-----
MIIDEDCCAs6gAwIBAgICAXIwCwYHKoZIzgEAwUAMFoxCzAJBgNVBAYTAlZJMSQw
IgYDVQQKEtUTU0ggQ29tbXVuaWNhdGlvbnMgU2VjdXJpdHkxETAPBgNVBAsTCFdl
Y8b0ZXNoMzR0Mi8EAYDVQQDEwwXZWh0dGhkMDoGEzIwMjAwHhcNMDMwMjEwMTYxNTEw
ekMDMxMjEwMDYwMDExWjByMQswCQYDVQQGEwJGMTcwHhcNMDMyNzI2MTIxMDgx
MTExMTQwMjAmMDMyNzI2MTIxMDgxWjEYMA0GCSqGSIb3DQECCAqEAx9awIodH+l
75+CDWt0W9TGyWm/cC2f+e0kTc6eKRqQ+Zzyg4Ux0QsT6XZGJx+tGqOmoG6JW1
-----END X509 CERTIFICATE-----
```

-----BEGIN CERTIFICATE-----
MIIFEDCCAs6gAwIBAgICAXIwCwYHKoZIzgEAwUAMFoxCzAJBgNVBAYTAlZJMSQw
IgYDVQQKEtUTU0ggQ29tbXVuaWNhdGlvbnMgU2VjdXJpdHkxETAPBgNVBAsTCFdl
Y8b0ZXNoMzR0Mi8EAYDVQQDEwwXZWh0dGhkMDoGEzIwMjAwHhcNMDMwMjEwMTYxNTEw
ekMDMxMjEwMDYwMDExWjByMQswCQYDVQQGEwJGMTcwHhcNMDMyNzI2MTIxMDgx
MTExMTQwMjAmMDMyNzI2MTIxMDgxWjEYMA0GCSqGSIb3DQECCAqEAx9awIodH+l
75+CDWt0W9TGyWm/cC2f+e0kTc6eKRqQ+Zzyg4Ux0QsT6XZGJx+tGqOmoG6JW1
-----END CERTIFICATE-----
quit

Hash: 4e904504dc4e5b95e08129430e2a0b97ceef0ad1394f905b42df2db8f751be0244a711bb06edd4a2f07dd640c187f14c16fa0bed28e038b286741a880539d6ed06a68b7e324bfde6f3d0b1783d94e598d4943f5988a7a0f27f6b6b932dc0410376247160752853858db7a1951245c6b414b109effc430e177623720de56f4

* Do you accept this certificate? [y]y
crypto ca certificate chain <name>

Use the crypto ca certificate chain command to enter the Certificate Configuration for the specified CA. See Certificate Configuration Command Set on page 1131 for more information.

Syntax Description

| <name> | Specifies CA profile by alphanumeric string of up to 32 characters. |

Default Values

No defaults necessary for this command.

Functional Notes

Typically used only in the running-config and startup-config to restore certificates.

Usage Examples

The following example enters the Certificate Configuration for the CA profile MyProfile:

ProCurve(config)#crypto ca certificate chain MyProfile
ProCurve(config-cert-chain)#
crypto ca enroll <name>

Use the crypto ca enroll command to begin CA enrollment procedures.

Syntax Description

| <name> | Specifies CA profile by alphanumeric string of up to 32 characters. |

Default Values

No defaults necessary for this command.

Functional Notes

The type of enrollment procedure is based on the enrollment command and its settings. See enrollment terminal on page 1124 and enrollment url <url> on page 1125 for more information. This command initiates a dialog that is used to fill in the parameters that make up an enrollment request to be forwarded to a certificate authority. Note that some of the parameters (such as IP address) may be filled in using the values supplied in the crypto ca profile (in which case, the enrollment dialog will not prompt for those parameters). Once all required parameters are defined using the dialog, this command assembles them into an enrollment request to be sent to a certificate authority (including the generation of public and private keys). See crypto ca profile for more information.

If enrollment is set to terminal, you may view the request on the terminal screen.

If enrollment is set to url, the request is sent automatically to the certificate authority using the URL specified by the enrollment url command.

Usage Examples

The following example shows a typical enrollment dialog:

ProCurve(config)#crypto ca enroll MyProfile

**** Press CTRL+C to exit enrollment request dialog. ****
* Enter signature algorithm (RSA or DSS) [rsa]:rsa
* Enter the modulus length to use [512]:1024
* Enter the subject name as an X.500 (LDAP) DN:CN=Rrouter,C=US,L=Roseville,S=CA
  --The subject name in the certificate will be CN=CN=Rrouter,C=US,L=Roseville,S=CA.
* Include an IP address in the subject name [n]:y
* Enter IP address or name of interface to use:10.200.1.45
* Include fully qualified domain name [n]:y
* Enter the fully qualified domain name to use:FullyQualifiedDomainName
* Include an email address [n]:y
* Enter the email address to use:myemail@email.com
Generating request (including keys)....
crypto ca import <name> certificate

Use the **crypto ca import certificate** command to import a certificate manually via the console terminal.

**Syntax Description**

<name> Specifies CA profile by alphanumeric string of up to 32 characters.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

Puts CLI in mode where the certificate can be entered manually. Enter `quit` and a carriage return (or simply enter two consecutive carriage returns) to exit this mode. Abort this mode by pressing `Ctrl-C`. This command only applies if the enrollment command is set to `terminal`. See `enrollment terminal` on page 1124.

**Usage Examples**

The following example imports a certificate via the console terminal:

```
ProCurve(config)#crypto ca import MyProfile certificate
Enter the PM-encoded certificate. End with two consecutive carriage returns or the word "quit" on a line by itself:

-----BEGIN CERTIFICATE-----
MIIDWTCCAwOgAwIBAgIKFLCsOgAAAAAATjANBgkqhkiG9w0BAQUFADBjMQswCQYD
VQQGewJVUzEQMA4GA1UECBMHOQxQETAPBgNVBAoTC0lJb3J MPIcKmEwMDA1MB8GA1UE
CwwJU2VydmljZSIEcmVvdXQgQXVzdHJhbWYgY3JlZSB5b3UgUHJvZ3JhbWUgQ29yZ
ZCGwcG9wczEVbmx5IHN0dXJ0Q29yZzEyMDQgalementa4qIO0GAIy8dBYQaIw4qZIy
8dBYCggZG9wczEVbmd1cnNlIENyZWF0ZSBQb3JncmVzc2l0ZWQgY29yc3Rpb24gUGFy
Z2VyZ2V0LmNybDAoLjIA==

-----END CERTIFICATE-----
Success!
```
crypto ca import <name> crl

Use the crypto ca import crl command to import a CRL manually via the console terminal.

Syntax Description

- **<name>**: Specifies CA profile by alphanumeric string of up to 32 characters.

Default Values

No defaults necessary for this command.

Functional Notes

Puts CLI in a mode where the CRL can be entered manually. Enter quit and a carriage return (or simply enter two consecutive carriage returns) to exit this mode. This command only applies if the enrollment command is set to terminal. See enrollment terminal on page 1124.

Usage Examples

The following allows you to manually paste in the CA's CRL:

```
ProCurve(config)#crypto ca import MyProfile crl
```
**crypto ca profile** `<name>`

Use the `crypto ca profile` command to define a CA and to enter the CA Profile Configuration. See *CA Profile Configuration Command Set* on page 1120 for more information.

**Syntax Description**

| `<name>` | Specifies CA profile by alphanumeric string of up to 32 characters. |

**Default Values**

No defaults necessary for this command.

**Functional Notes**

Use this to specify the type of enrollment, as well as enrollment request parameters. See the *Functional Notes* of the command `crypto ca enroll <name>` on page 315 for more information.

**Usage Examples**

The following example creates the CA profile called *MyProfile* and enters the CA Profile Configuration for that certificate authority:

```
ProCurve(config)#crypto ca profile MyProfile
Configuring New CA Profile MyProfile.
ProCurve(ca-profile)#
```
crypto ike

Use the `crypto ike` command to define the system-level local ID for IKE negotiations and to enter the IKE Client or IKE Policy command sets.

Variations of this command include the following:

- `crypto ike client configuration pool <poolname>`
- `crypto ike local-id address`
- `crypto ike policy <policy priority>`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>client configuration pool &lt;poolname&gt;</code></td>
<td>Creates a local pool named the <code>&lt;poolname&gt;</code> of your choice and enters the IKE Client. Clients that connect via an IKE policy that specifies this pool-name will be assigned values from this pool. See the section for more information.</td>
</tr>
<tr>
<td><code>local-id address</code></td>
<td>Sets the local ID during IKE negotiation to be the IP address of the interface from which the traffic exits. This setting can be overridden on a per-policy basis using the <code>local-id</code> command in the IKE Policy (see `local-id [address</td>
</tr>
<tr>
<td><code>policy &lt;policy priority&gt;</code></td>
<td>Creates an IKE policy with the <code>&lt;policy priority&gt;</code> of your choice and enters the IKE Policy. See IKE Policy Command Set on page 1167 for more information.</td>
</tr>
</tbody>
</table>

**Default Values**

There are no default settings for this command.

**Usage Examples**

The following example creates an IKE policy with a policy priority setting of 1 and enters the IKE Policy for that policy:

```
ProCurve(config)#crypto ike policy 1
ProCurve(config-ike)#
```

**Technology Review**

The following example configures an SROS product for VPN using IKE aggressive mode with pre-shared keys. The SROS product can be set to initiate IKE negotiation in main mode or aggressive mode. The product can be set to respond to IKE negotiation in main mode, aggressive mode, or any mode. In this example, the device is configured to initiate in aggressive mode and to respond to any mode.

This example assumes that the SROS product has been configured with a WAN IP Address of 172.16.45.57 on interface `ppp 1` and a LAN IP Address of 10.10.10.254 on interface `ethernet 0/1`. The Peer Private IP Subnet is 10.10.20.0.

For more detailed information on VPN configuration, refer to the VPN Configuration Guide located on the...
provided with your unit.
Step 1:
Enter the Global configuration mode (i.e., config terminal mode).

ProCurve>**enable**
ProCurve#**configure terminal**

Step 2:
Enable VPN support using the **ip crypto** command. This command allows crypto maps to be applied to interfaces, and enables the IKE server to listen for IKE negotiation sessions on UDP port 500.

ProCurve(config)#**ip crypto**

Step 3:
Set the local ID. During IKE negotiation, local-ids are exchanged between the local device and the peer device. In the SROS, the default setting for all local-ids is configured by the **crypto ike local-id** command. The default setting is for all local-ids to be the IPv4 address of the interface over which the IKE negotiation is occurring. In the future, a unique system-wide Hostname or Fully Qualified Domain Name could be used for all IKE negotiation.

ProCurve(config)#**crypto ike local-id address**

Step 4:
Create IKE policy. In order to use IKE negotiation, an IKE policy must be created. Within the system, a list of IKE policies is maintained. Each IKE policy is given a priority number in the system. That priority number defines the position of that IKE policy within the system list. When IKE negotiation is needed, the system searches through the list, starting with the policy with priority of 1, looking for a match to the peer IP address.

An individual IKE policy can override the system local-id setting by having the **local-id** command specified in the IKE policy definition. This command in the IKE policy is used to specify the type of local-id and the local-id data. The type can be of IPv4 address, Fully Qualified Domain Name, or User-Specified Fully Qualified Domain Name.

An IKE policy may specify one or more peer IP addresses that will be allowed to connect to this system. To specify multiple unique peer IP addresses, the **peer A.B.C.D** command is used multiple times within a single IKE policy. To specify that all possible peers can use a default IKE policy, the **peer any** command is given instead of the **peer A.B.C.D** command inside of the IKE policy. The policy with the **peer any** command specified will match to any peer IP address (and therefore should be given the highest numerical priority number). This will make the policy the last one to be compared against during IKE negotiation.

ProCurve(config)#**crypto ike policy 10**
ProCurve(config-ike)#**no local-id**
ProCurve(config-ike)#**peer 172.16.15.129**
ProCurve(config-ike)#**initiate aggressive**
ProCurve(config-ike)#**respond anymode**
ProCurve(config-ike)#attribute 10
ProCurve(config-ike-attribute)#encryption 3des
ProCurve(config-ike-attribute)#hash sha
ProCurve(config-ike-attribute)#authentication pre-share
ProCurve(config-ike-attribute)#group 1
ProCurve(config-ike-attribute)#lifetime 86400

Step 5:
Define the remote-id settings. The `crypto ike remote-id` command is used to define the remote-id for a peer connecting to the system, specify the preshared-key associated with the specific remote-id, and (optionally) determine that the peer matching this remote-id should not use mode config (by using the `no-mode-config` keyword). See `crypto ike remote-id` on page 323 for more information.

ProCurve(config)#crypto ike remote-id address 172.16.15.129 preshared-key mysecret123

Step 6:
Define the transform-set. A transform-set defines the encryption and/or authentication algorithms to be used to secure the data transmitted over the VPN tunnel. Multiple transform-sets may be defined in a system. Once a transform-set is defined, many different crypto maps within the system can reference it. In this example, a transform-set named `highly_secure` has been created. This transform-set defines ESP with Authentication implemented using 3DES encryption and SHA1 authentication.

ProCurve(config)#crypto ipsec transform-set highly_secure esp-3des esp-sha-hmac
ProCurve(cfg-crypto-trans)#mode tunnel

Step 7:
Define an ip-access list. An Extended Access Control List is used to specify which traffic needs to be sent securely over the VPN tunnel. The entries in the list are defined with respect to the local system. The source IP address will be the source of the traffic to be encrypted. The destination IP address will be the receiver of the data on the other side of the VPN tunnel.

ProCurve(config)#ip access-list extended corporate_traffic
ProCurve(config-ext-nacl)#permit ip 10.10.10.0 0.0.0.255 10.10.20.0 0.0.0.255 log
    deny ip any any

Step 8:
Create crypto map. A Crypto Map is used to define a set of encryption schemes to be used for a given interface. A crypto map entry has a unique index within the crypto map set. The crypto map entry will specify whether IKE is used to generate encryption keys or if manually specified keys will be used. The crypto map entry will also specify who will be terminating the VPN tunnel, as well as which transform-set or sets will be used to encrypt and/or authenticate the traffic on that VPN tunnel. It also specifies the lifetime of all created IPSec Security Associations.

ProCurve(config)#crypto map corporate_vpn 1 ipsec-ike
ProCurve(config-crypto-map)#match address corporate_traffic
ProCurve(config-crypto-map)#set peer 172.16.15.129
ProCurve(config-crypto-map)#set transform-set highly_secure
ProCurve(config-crypto-map)#set security-association lifetime kilobytes 8000
ProCurve(config-crypto-map)#set security-association lifetime seconds 28800
ProCurve(config-crypto-map)#no set pfs

Step 9:
Configure public interface. This process includes configuring the IP address for the interface and applying the appropriate crypto map to the interface. Crypto maps are applied to the interface on which encrypted traffic will be transmitted.

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip address 172.16.45.57 255.255.255.248
ProCurve(config-ppp 1)#crypto map corporate_vpn
ProCurve(config-ppp 1)#no shutdown

Step 10:
Configure private interface to allow all traffic destined for the VPN tunnel to be routed to the appropriate gateway.

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip address 10.10.10.254 255.255.255.0
ProCurve(config-eth 0/1)#no shutdown
ProCurve(config-eth 0/1)#exit
crypto ike remote-id

Use the crypto ike remote-id command to specify the remote ID and to associate a pre-shared key with the remote ID.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>Specifies a remote ID of IPv4 type.</td>
</tr>
<tr>
<td>any</td>
<td>Wildcard that allows any remote ID (type and value).</td>
</tr>
<tr>
<td>asn1-dn</td>
<td>Specifies an Abstract Syntax Notation Distinguished Name as the remote ID.</td>
</tr>
<tr>
<td>crypto map</td>
<td>Optional. Specifies the crypto map this remote ID corresponds to.</td>
</tr>
<tr>
<td>fqdn</td>
<td>Specifies a fully qualified domain name (e.g., procurve.com) as the remote ID.</td>
</tr>
<tr>
<td>ike policy</td>
<td>Optional. Specifies the IKE policy this remote ID corresponds to.</td>
</tr>
<tr>
<td>user-fqdn</td>
<td>Specifies a user fully qualified domain name or email address as the remote ID.</td>
</tr>
<tr>
<td>preshared-key</td>
<td>Associates a pre-shared key with this remote ID.</td>
</tr>
<tr>
<td>no-mode-config</td>
<td>Optional keyword used to specify that the peer matching this remote ID should not use mode config.</td>
</tr>
<tr>
<td>no-xauth</td>
<td>Optional keyword used to specify that the peer matching this remote ID should not use xauth.</td>
</tr>
<tr>
<td>nat-t [v1 l v2] [allow l force l disable]</td>
<td>Optional keyword that denotes whether peers matching this remote ID should allow, disable, or force NAT traversal versions 1 and 2.</td>
</tr>
</tbody>
</table>

Default Values

There are no default settings for this command.

Functional Notes

The fqdn and user-fqdn <WORD> line can include wildcard characters. The wildcard characters are "*" for a 0 or more character match and "?" for a single character match. Currently, the "?" cannot be set up using the CLI, but it can be transferred to the unit via the startup-config.

Example for user-fqdn:
john*@domain.com
will match:
johndoe@domain.com
johnjohn@myemail.com
Example for **fqdn**:

*.*domain.com

will match:

www.domain.com
ftp.domain.com
one.www.domain.com

The **address** remote ID can be in the form of a single host address or in the form of an IP address wildcard.

Example for **address** type:

**crypto ike remote id address 10.10.10.0 0.0.0.255**

will match:

10.10.10.1
10.10.10.2
and all IP addresses in the form of 10.10.10.X (where X is 0-255)

The **asn1-dn** <WORD> line can include wildcard characters. The wildcard characters are "*" for a 0 or more character match and "?" for a single character match. Currently, the "?" cannot be set up using the CLI, but it can be transferred to the unit via the startup-config.

Example for typical **asn1-dn** format with no wildcards:

**crypto ike remote-id asn1-dn "CN=MyRouter, C=US, S=CA, L=Roseville, O-HP, OU=TechSupport"**

(matches only remote ID strings with all fields exactly the same)

Example for typical **asn1-dn** format with wildcards used to match a string within a field:

**crypto ike remote-id asn1-dn "CN=*, C=*, S=*, L=*, O=*, OU=*"**

(matches any asn1-dn remote ID string from a peer)

Example for typical **asn1-dn** format with wildcards used to match a portion of the remote ID:

**crypto ike remote-id asn1-dn "CN=*, C=US, S=CA, L=Roseville, O=HP, OU=***"**

(matches any remote ID string with the same values for the C, S, L, and O fields, and any values in the CN and OU fields)

Example for typical **asn1-dn** format with wildcards used to match a portion of a field:

**crypto ike remote-id asn1-dn "CN=My*, C=US, S=CA, L=Roseville, O=HP, OU=TechSupport"**

(matches remote ID strings with all fields exactly the same, but with any CN field beginning with "My")

### Usage Examples

The following example assigns a remote ID of 63.97.45.57 and associates the pre-shared key **mysecret** with the remote ID:

ProCurve(config)#**crypto ike remote-id address 63.97.45.57 preshared-key mysecret**
crypto ipsec transform-set <setname> <parameters>

Use the `crypto ipsec transform-set` command to define the transform configuration for securing data (e.g., esp-3des, esp-sha-hmac, etc.). The transform-set is then assigned to a crypto map using the map’s `set transform-set` command. See `set transform-set <setname1 - setname6>` on page 1145.

**Syntax Description**

- `<setname>` Specifies a name for the transform-set you are about to define.
- `<parameters>` Assigns a combination of up to three security algorithms. This field is a valid combination of the following:
  - ah-md5-hmac, ah-sha-hmac
  - esp-des, esp-3des, esp-aes-128-cbc, esp-aes-192-cbc, esp-aes-256-cbc, esp-null
  - esp-md5-hmac, esp-sha-hmac

**Default Values**

There are no default settings for this command.

**Functional Notes**

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms.

If no transform-set is configured for a crypto map, the entry is incomplete and will have no effect on the system.

**Usage Examples**

The following example first creates a transform-set (`Set1`) consisting of two security algorithms (up to three may be defined), and then assigns the transform-set to a crypto map (`Map1`):

```
ProCurve(config)#crypto ipsec transform-set Set1 esp-3des esp-sha-hmac
ProCurve(config-crypto-trans)#exit

ProCurve(config)#crypto map Map1 1 ipsec-ike
ProCurve(config-crypto-map)#set transform-set Set1
```
crypto map

Use the `crypto map` command to define crypto map names and numbers and to enter the associated (either Crypto Map IKE or Crypto Map Manual).

Variations of this command include the following:

- `crypto map <mapname> <mapindex> ipsec-ike`
- `crypto map <mapname> <mapindex> ipsec-manual`

**Note**

For VPN configuration example scripts, refer to the technical support note *VPN Configuration Guide* located on the ProCurve SROS Documentation CD provided with your unit.

**Syntax Description**

- `<mapname>` Names the crypto map. You can assign the same name to multiple crypto maps, as long as the map index numbers are unique.
- `<mapindex>` Assigns a crypto map sequence number.
- `ipsec-ike` Specifies the Crypto Map IKE (see *Crypto Map IKE Command Set* on page 1135). This supports IPSec entries that will use IKE to negotiate keys.

- `ipsec-manual` Specifies the Crypto Map Manual (see *Crypto Map IKE Command Set* on page 1135). This supports manually configured IPSec entries.

**Default Values**

There are no default settings for this command.

**Functional Notes**

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms (see `crypto ipsec transform-set <setname> <parameters>` on page 325).

Crypto map entries do not directly contain the selectors used to determine which data to secure. Instead, the crypto map entry refers to an access control list. An access control list is assigned to the crypto map using the `match address` command (see `ike-policy <policy number>` on page 1139).

If no transform-set or access-list is configured for a crypto map, the entry is incomplete and will have no effect on the system.

When you apply a crypto map to an interface (using the `crypto map` command within the interface’s), you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.
Usage Examples

The following example creates a new IPSec IKE crypto map called **testMap** with a map index of **10**:

```
ProCurve(config)#crypto map testMap 10 ipsec-ike
ProCurve(config-crypto-map)#
```

Technology Review

A crypto map entry is a single policy that describes how certain traffic is to be secured. There are two types of crypto map entries: ipsec-manual and ipsec-ike. Each entry is given an index, which is used to sort the ordered list. When a non-secured packet arrives on an interface, the crypto map set associated with that interface is processed in order. If a crypto map entry matches the non-secured traffic, the traffic is discarded.

When a packet is to be transmitted on an interface, the crypto map set associated with that interface is processed in order. The first crypto map entry that matches the packet will be used to secure the packet. If a suitable SA (security association) exists, that is used for transmission. Otherwise, IKE is used to establish an SA with the peer. If no SA exists, and the crypto map entry is “respond only”, the packet is discarded.

When a secured packet arrives on an interface, its SPI (security parameter index) is used to look up an SA. If an SA does not exist, or if the packet fails any of the security checks (bad authentication, traffic does not match SA selectors, etc.), it is discarded. If all checks pass, the packet is forwarded normally.
data-call [authentication protocol | sent authentication protocol] [chap | pap]

Use the data-call authentication protocol and data-call sent authentication protocol commands to set the pre-authentication defaults for inbound demand routing calls. Use the no form of these commands to return to the default settings. For more detailed information on CHAP and PAP, refer to the Technology Review section of the command ppp authentication [chap | pap] on page 1040.

Syntax Description

<table>
<thead>
<tr>
<th>authentication protocol</th>
<th>Sets the authentication protocol expected for inbound calls.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sent authentication protocol</td>
<td>Sets the authentication protocol sent for inbound calls.</td>
</tr>
<tr>
<td>chap</td>
<td>Configures CHAP authentication.</td>
</tr>
<tr>
<td>pap</td>
<td>Configures PAP authentication.</td>
</tr>
</tbody>
</table>

Default Values

By default, there is no configuration for authentication.

Functional Notes

There are certain PPP parameters that must be known before PPP can negotiate an inbound call when using demand routing. To ensure PPP convergence, it is recommended (in most cases) that demand routing interfaces use the same settings as those specified in the data-call commands. If the PPP parameters do not match the authenticated user, the link is renegotiated.

Usage Examples

The following example sets the authentication protocol expected for incoming calls to CHAP. The router will then authenticate the peer using CHAP:

ProCurve(config)#data-call authentication protocol chap

The following example sets the authentication protocol sent for incoming calls to PAP. This router may be authenticated by the peer using PAP:

ProCurve(config)#data-call sent authentication protocol pap
**data-call [mtu <number> | multilink]**

Use the `data-call` commands to set the pre-authentication defaults for maximum transmit unit (MTU) size or to enable multilink for inbound demand routing calls. Use the `no` form of each command to return to the factory default settings. See the `mtu <size>` on page 1038 for more detailed syntax descriptions.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtu &lt;number&gt;</td>
<td>Sets the maximum size for the transmit unit. Valid range: 64 to 1520.</td>
</tr>
<tr>
<td>multilink</td>
<td>Enables the negotiation of multilink MRU size for inbound calls.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the MTU size is 1500 and multilink is disabled.

**Functional Notes**

There are certain PPP parameters that must be known before PPP can negotiate an inbound call when using demand routing. To ensure PPP convergence, it is recommended (in most cases) that demand routing interfaces use the same settings as those specified in the `data-call` commands. The `data-call mtu <number>` command sets the MTU and controls the negotiated maximum receive unit (MRU) size during incoming calls for link control protocol (LCP) negotiation. If the PPP parameters do not match the authenticated user, the link is renegotiated.

**Usage Examples**

The following example specifies an MTU of 1200 on the demand routing interface:

```
ProCurve(config)#data-call MTU 1200
```

The following example enables multilink for inbound demand routing calls:

```
ProCurve(config)#data-call multilink
```
enable password [md5] <password>

Use the `enable password` command to define a password (with optional encryption) for accessing the Enable mode. Use the `no enable password` command to remove a configured password.

**Note**  To prevent unauthorized users from accessing the configuration functions of your device, immediately install an Enable-level password.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>md5</td>
<td>Optional. Specifies Message Digest 5 (md5) as the encryption protocol to use when displaying the enable password during show commands. If the md5 keyword is not used, encryption is not used when displaying the enable password during show commands.</td>
</tr>
<tr>
<td>&lt;password&gt;</td>
<td>String (up to 30 characters in length) to use as the Enable Security mode password.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there is no configured enable password.

**Usage Examples**

To provide extra security, the SROS can encrypt the enable password when displaying the current configuration. For example, the following is a `show configuration` printout (password portion) with an unencrypted enable password (PASSWORD):

```
!
enable password PASSWORD
!
```

Alternately, the following is a `show configuration` printout (password portion) with an enable password of password using md5 encryption:

```
!
enable password md5 encrypted 5aa5fbae7d01a90e79fb57705ce74676
!
```
event-history on

Use the `event-history on` command to enable event logging for the SROS system. Event log messages will not be recorded unless this command has been issued (regardless of the `event-history priority` configured). The event log may be displayed using the `show event-history` command. Use the `no` form of this command to disable the event log.

Syntax Description

No subcommands.

Default Values

By default, the SROS event logging capabilities are disabled.

Functional Notes

The event history provides useful information regarding the status of the system and individual port states. Use the event history as a troubleshooting tool when identifying system issues. The following is a sample event history log.

ProCurve#show event-history
Using 526 bytes
2002.07.12 15:34:01 T1.t1 1/1 Yellow
2002.07.12 15:34:01 INTERFACE_STATUS.t1 1/1 changed state to down.
2002.07.12 15:34:02 T1.t1 1/1 No Alarms
2002.07.12 15:34:02 INTERFACE_STATUS.t1 1/1 changed state to up.
2002.07.12 15:34:03 INTERFACE_STATUS.eth 0/1 changed state to up.
2002.07.12 15:34:10 OPERATING_SYSTEM Warm Start
2002.07.12 15:34:12 PPP.NEGOTIATION LCP up
2002.07.12 15:34:12 PPP.NEGOTIATION IPCP up

Usage Examples

The following example enables the SROS event logging feature:

ProCurve(config)#event-history on
event-history priority [error | fatal | info | notice | warning]

Use the `event-history priority` command to set the threshold for events stored in the event history. All events with the specified priority or higher will be kept for viewing in the local event log. The event log may be displayed using the `show event-history` command. Use the `no` form of this command to keep specified priorities from being logged.

**Syntax Description**

Sets the minimum priority threshold for logging messages to the event history. The following priorities are available (ranking from lowest to highest):

**Error**
When selected, events with `error` and `fatal` priorities are logged.

**Fatal**
When selected, only events with a `fatal` priority are logged.

**Info**
When selected, all events are logged.

**Notice**
When selected, events with `notice`, `warning`, `error`, and `fatal` priorities are logged.

**Warning**
When selected, events with `warning`, `error`, and `fatal` priorities are logged.

**Default Values**

By default, no event messages are logged to the event history.

**Functional Notes**

The event history provides useful information regarding the status of the system and individual port states. Use the event history as a troubleshooting tool when identifying system issues. The following is a sample event history log.

**ProCurve#show event-history**

Using 526 bytes
2002.07.12 15:34:01 T1.t1 1/1 Yellow
2002.07.12 15:34:01 INTERFACE_STATUS.t1 1/1 changed state to down.
2002.07.12 15:34:02 T1.t1 1/1 No Alarms
2002.07.12 15:34:02 INTERFACE_STATUS.t1 1/1 changed state to up.
2002.07.12 15:34:03 INTERFACE_STATUS.eth 0/1 changed state to up.
2002.07.12 15:34:10 OPERATING_SYSTEM Warm Start
2002.07.12 15:34:12 PPP.NEGOTIATION LCP up
2002.07.12 15:34:12 PPP.NEGOTIATION IPCP up
Usage Examples
The following example logs all events to the event history:

ProCurve>enable
ProCurve#config terminal
ProCurve(config)#event-history priority info
exception memory minimum <memory>

Use the exception memory minimum command to initiate a reboot when the specified minimum amount of memory is no longer available. This ensures that adequate memory is available to store an exception report. Use the no form of this command to disable rebooting when the minimum memory limitation is violated.

**Caution**  Executing the exception memory minimum command may cause the unit to reboot. Use this command only if advised to by Technical Support.

**Syntax Description**

<memory> Specifies the minimum amount of memory (in bytes) that must be free before a reboot occurs.

**Default Values**

By default, exception memory minimum is disabled.

**Usage Examples**

The following example sets the exception memory minimum to 3 Mb:

ProCurve(config)#exception memory minimum 30000000
exception report [file-name <filename>]

Use the exception report command to change the default output filename for generated exception reports. An exception report contains a snap shot of current system processes to use when troubleshooting.

Syntax Description

| file-name <filename> | Optional. Specifies a filename for the exception report other than the default filename. The specified filename replaces the exception-report in the default filename. |

Default Values

By default, the exception report filename is exception-report-yyyyMMddHHmmss. (The yyyyMMddHHmmss is automatically populated with the actual year, month, day, hour, minutes, and seconds when the report was generated.) Specifying a new filename results in the following format: filename-yyyyMMddHHmmss.

Usage Example

The following example specifies the output filename for an exception report:

ProCurve(config)# exception report file-name thereport
ProCurve(config)# exit
ProCurve# exception report generate
Exception report generated.
ProCurve# sh flash
  1744 startup-config
  45676 thereport-20050708080537
ProCurve# config t
ProCurve(config)# no exception report file-name
ProCurve(config)# exit
Appropriate commands must be issued to preserve configuration.
ProCurve# exception report generate
Exception report generated.
ProCurve# sh flash
  1744 startup-config
  45676 thereport-20050708080537
  45900 exception-report-20050708080552
ftp authentication <listname>

Use the ftp authentication command to attach AAA login authentication lists to the FTP server (see aaa authentication login [<listname> | default] [none | line | enable | local | group <groupname> | group radius | group tacacs+] on page 294 for more information). This list is only used if the AAA subsystem has been activated with the aaa on command.

Syntax Description

| <listname> | Specifies the named list created with the aaa authentication login command. Enter default to use the AAA default login list. |

Default Values

There is no default configuration for the list. If AAA is turned on but no ftp authentication list has been assigned, FTP denies all login attempts.

Usage Examples

The following example attaches the authentication list, MyList, to the FTP server:

ProCurve(config)#ftp authentication MyList

The following example specifies that the SROS use the default AAA login list for FTP authentication:

ProCurve(config)#ftp authentication default
hostname <name>

Creates a name used to identify the unit. This alphanumeric string should be used as a unique description for the unit. This string will be displayed in all prompts.

Syntax Description

| <name> | Identifies the unit by alphanumeric string of up to 32 characters. |

Default Values

By default, the hostname is `ProCurveSRXXXXdl` (where XXXX is the model number of the router). For example, the default for the ProCurve Secure Router 7203dl is `ProCurveSR7203dl`.

Usage Examples

The following example creates a hostname for the SROS device of `HP_RTR` to identify the system as the HP router:

```
ProCurve(config)#hostname HP_RTR
```
interface <port-type> <slot/port>

Activates the Interface Configuration mode for the listed physical interface.

**Syntax Description**

| <port-type> | Identifies the physical port type of the installed Interface Module, Backup Module or Ethernet port. Type `interface ?` for a complete list of valid interfaces. |
| <slot/port>  | Specifies an interface based on its physical location (slot and port). For example, if you have a T1/DSX-1 installed in Slot 1 of an SROS product: |
|             | - The **WAN-T1** port would be specified in the CLI as `t1 1/1`. |
|             | - The **DSX-1** port would be specified as `t1 1/2`. |
|             | - If (for example) a backup module is also installed, then the **backup** port would be specified as `bri 1/3`. |
|             | - If you are specifying a port that is built into the base unit (e.g., the Ethernet port), the slot number is 0. For example, the Ethernet (**LAN**) port would be specified as `ethernet 0/1`. |

**Default Values**

No default values required for this command.

**Usage Examples**

The following example enters the serial interface for a serial module installed in slot 1:

```
ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#
```
interface atm <label> point-to-point

Use the interface frame-relay command to create a virtual Asynchronous Transfer Mode (ATM) interface (or sublink, if specified) that is identified using the entered number label. In addition, entering this command activates the ATM interface command set. The point-to-point keyword (optional) can be used to identify the ATM endpoint as a point-to-point link (versus multipoint). Use the no form of this command to delete a configured virtual ATM interface.

To specify a virtual ATM sub-interface, the following syntax applies:
interface frame-relay <label>.<sublink label>

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;label&gt;</td>
<td>Specifies the numerical virtual ATM interface identifying label (valid range: 1 to 1024)</td>
</tr>
<tr>
<td>&lt;sublink label&gt;</td>
<td>Numerical label for the virtual sublink (valid range: 1-65535)</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Optional. Identifies the ATM interface as a point-to-point link (versus multilink) By default, all created ATM interfaces are point-to-point.</td>
</tr>
</tbody>
</table>

Default Values

By default, there are no configured virtual ATM interfaces or sublinks.

Functional Notes

Creating an endpoint that uses a layer 2 protocol (such as ATM) contains the following steps:

Step 1:
Create the ATM virtual endpoint (using the interface atm command) and set the protocol-specific configuration parameters and/or activate the interface. The following example creates a virtual ATM interface (labeled 1) and activates the interface:

ProCurve(config)#interface atm 1
ProCurve(config-atm 1)#no shutdown

Step 2:
Create the sub-interface and configure the PVC parameters. Using the sub-interface commands, apply access policies to the interface, create bridging interfaces, configure backup, assign an IP address, and set the PVC virtual path identifier (VPI) and virtual channel identifier (VCI) and define encapsulation method. For example, the following creates a ATM sub-interface labeled 22, sets the VPI and VCI to 16 and 32, respectively, specifies AAL5 LLC/SNAP encapsulation, and assigns an IP address of 10.44.69.1/30 to the interface:

ProCurve(config-atm 7)#interface atm 7.22
ProCurve(config-atm 7.22)#pvc 16/32
ProCurve(config-atm 7.22)#encapsulation aal5snap
ProCurve(config-atm 7.22)#ip address 10.44.69.1 255.255.255.252
ProCurve(config-atm 7.22)#no shutdown

Step 3:
Make the association between the layer 2 endpoint and the physical interface using the `bind` command. For example, the following creates a bind (labeled 5) to make an association between the ATM virtual interface (atm 7) and the adsl 1/1 interface.

ProCurve(config)#bind 5 adsl 1/1 atm 7

**Usage Examples**
The following example creates an ATM virtual interface (labeled 1) and enters the ATM Interface Configuration mode:

ProCurve(config)#interface atm 1
ProCurve(config-atm 1)#
**interface frame-relay** `<label>` **point-to-point**

Use the `interface frame-relay` command to create a virtual Frame Relay interface (or sublink, if specified) that is identified using the entered number label. In addition, entering this command activates the Frame Relay interface command set. The `point-to-point` keyword (optional) can be used to identify the Frame Relay endpoint as a point-to-point link (versus multipoint). Use the `no` form of this command to delete a configured virtual Frame Relay interface.

To specify a virtual Frame Relay sub-interface, the following syntax applies:

```
interface frame-relay `<label>`.<`sublink label>`
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;label&gt;</code></td>
<td>Specifies the numerical virtual Frame Relay interface identifying label (valid range: 1 to 1024)</td>
</tr>
<tr>
<td><code>&lt;sublink label&gt;</code></td>
<td>Numerical label for the virtual sublink (valid range: 1-1007)</td>
</tr>
<tr>
<td><code>point-to-point</code></td>
<td>Optional. Identifies the Frame Relay interface as a point-to-point link (versus multipoint). By default, all created Frame Relay interfaces are point-to-point.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no configured virtual Frame Relay interfaces or sublinks.

**Functional Notes**

Creating an endpoint that uses a layer 2 protocol (such as Frame Relay) is generally a four-step process:

**Step 1:**
Create the Frame Relay virtual endpoint (using the `interface frame-relay` command) and set the signaling method (using the `frame-relay lmi-type` command). Also included in the Frame Relay virtual endpoint are all the applicable Frame Relay timers, logging thresholds, encapsulation types, etc. Generally, most Frame Relay virtual interface parameters should be left at their default state. You must activate the interface (using the `no shutdown` command) before the interface is able to pass any data. For example, the following creates a Frame Relay interface labeled 7, sets the signaling method to `ansi`, and activates the interface:

```
ProCurve(config)#interface frame-relay 7
ProCurve(config-fr 7)#frame-relay lmi-type ansi
ProCurve(config-fr 7)#no shutdown
```

**Step 2:**
Create the sub-interface and configure the PVC parameters. Using the sub-interface commands, apply access policies to the interface, create bridging interfaces, configure backup, assign an IP address, and set the PVC data-link control identifier (DLCI). For example, the following creates a Frame Relay sub-interface labeled 22, sets the DLCI to 30, and assigns an IP address of `10.44.69.1/30` to the interface.

```
ProCurve(config-fr 7)#interface fr 7.22
```
ProCurve(config-fr 7.22)#frame-relay interface-dlci 30
ProCurve(config-fr 7.22)#ip address 10.44.69.1 255.255.255.252
ProCurve(config-fr 7.22)#no shutdown

Step 3: (VALID ONLY FOR T1 INTERFACES)
Specify the group of DS0s used for signaling on the T1 interface by creating a tdm-group. Group any number of aggregate DS0s together to create a data pipe for layer 2 signaling. Also use the tdm-group command to specify the per-DS0 signaling rate on the interface. For example, the following creates a tdm-group labeled 9 containing 20 DS0s (each DS0 having a data rate of 56 kbps).

ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#tdm-group 9 timeslots 1-20 speed 56
ProCurve(config-t1 1/1)#exit

Step 4:
Make the association between the layer 2 endpoint and the physical interface using the bind command. For example, the following creates a bind (labeled 5) to make an association between the Frame Relay virtual interface (fr 7) and the tdm-group configured on interface t1 1/1 (tdm-group 9).

ProCurve(config)#bind 5 t1 1/1 9 fr 7

Usage Examples
The following example creates a Frame Relay virtual interface (labeled 1) and enters the Frame Relay Interface Configuration mode:

ProCurve(config)#interface fr 1
ProCurve(config-fr 1)#
**interface hdlc <label>**

Use the `interface hdlc` command to create a virtual high level data link control interface that is identified using the entered number label. In addition, entering this command activates the HDLC interface command set. Use the `no` form of this command to delete a configured virtual HDLC interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;label&gt;</code></td>
<td>Specifies the numerical virtual HDLC interface identifying label (valid range: 1 to 1024)</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no configured HDLC interfaces.

**Functional Notes**

Creating an endpoint that uses a layer 2 protocol (such as HDLC) is generally a four-step process:

**Step 1:**
Create the HDLC virtual endpoint (using the `interface hdlc`) command and enter the HDLC configuration commands.

```
ProCurve(config)#interface hdlc 7
ProCurve(config-hdlc 7)#
```

**Step 2:**
Configure the interface parameters to apply access policies to the interface, create bridging interfaces, configure backup, and assign an IP address. You must activate the interface (using the `no shutdown` command) before the interface is able to pass any data. For example, the following assigns an IP address of `10.44.69.1/30` to the interface.

```
ProCurve(config-hdlc 7)#ip address 10.44.69.1 255.255.255.252
ProCurve(config-hdlc 7)#no shutdown
```

**Step 3:** (VALID ONLY FOR T1 INTERFACES)
Specify the group of DS0s used for signaling on the T1 interface by creating a `tdm-group`. Group any number of aggregate DS0s together to create a data pipe for layer 2 signaling. Also use the `tdm-group` command to specify the per-DS0 signaling rate on the interface. For example, the following creates a `tdm-group` labeled `9` containing 20 DS0s (each DS0 having a data rate of 56 kbps).

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#tdm-group 9 timeslots 1-20 speed 56
ProCurve(config-t1 1/1)#exit
```
Step 4:
Make the association between the layer 2 endpoint and the physical interface using the `bind` command. For example, the following creates a bind (labeled 5) to make an association between the HDLC virtual interface (`hdlc 7`) and the tdm-group configured on interface t1 1/1 (`tdm-group 9`).

```
ProCurve(config)#bind 5 t1 1/1 9 hdlc 7
```

**Usage Examples**

The following example creates a HDLC virtual interface (labeled 1) and enters the HDLC Interface Configuration mode:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#
```
**interface loopback <label>**

Use the `interface loopback` command to create a virtual interface that can be assigned layer 3 and higher properties and is always up unless the router is shut down. Use the `no` form of this command to delete a configured loopback interface.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;label&gt;</code></th>
<th>Specifies the numerical virtual loopback interface identifying label (valid range: 1 to 1024)</th>
</tr>
</thead>
</table>

**Default Values**

By default, there are no configured loopback interfaces.

**Usage Examples**

The following example creates a loopback virtual interface (labeled 1) and enters the Loopback Interface Configuration mode:

ProCurve(config)#`interface loopback 1`
ProCurve(config-loop 1)#
interface ppp <label>

Use the interface ppp command to create a virtual point-to-point protocol (PPP) interface that is identified using the entered number label. In addition, entering this command activates the PPP interface command set. Use the no form of this command to delete a configured virtual PPP interface.

Syntax Description

<label> Specifies the numerical virtual PPP interface identifying label (range: 1 to 1024).

Default Values

By default, there are no configured PPP interfaces.

Functional Notes

Creating an endpoint that uses a layer 2 protocol (such as PPP) is generally a four-step process:

Step 1:
Create the PPP virtual endpoint (using the interface ppp command) and enter the PPP command set.

ProCurve(config)#interface ppp 7
ProCurve(config-ppp 7)#

Step 2:
Configure the interface parameters to apply access policies to the interface, create bridging interfaces, configure backup, and assign an IP address. You must activate the interface (using the no shutdown command) before the interface can pass data. For example, the following assigns an IP address of 172.24.69.1/30 to the interface.

ProCurve(config-ppp 7)#ip address 172.24.69.1 255.255.255.252
ProCurve(config-ppp 7)#no shutdown

Step 3: (VALID ONLY FOR T1 INTERFACES)
Specify the group of DS0s used for signaling on the T1 interface by creating a tdm-group. Group any number of aggregate DS0s together to create a data pipe for layer 2 signaling. Also use the tdm-group command to specify the per-DS0 signaling rate on the interface. For example, the following creates a tdm-group labeled 9 containing 20 DS0s (each DS0 having a data rate of 56 kbps).

ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#tdm-group 9 timeslots 1-20 speed 56
ProCurve(config-t1 1/1)#exit
Step 4:
Make the association between the layer 2 endpoint and the physical interface using the `bind` command. For example, the following creates a bind (labeled 5) to make an association between the PPP virtual interface (`ppp 7`) and the tdm-group configured on interface `t1 1/1` (`tdm-group 9`).

```
ProCurve(config)#bind 5 t1 1/1 9 ppp 7
```

**Usage Examples**

The following example creates a PPP virtual interface (labeled 1) and enters the PPP Interface Configuration mode:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#
```
interface range [e1 | t1] <slot/port> - <slot/port>

Use the interface range command to enter configuration mode for a range of interfaces.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>Specifies the interface type as E1.</td>
</tr>
<tr>
<td>t1</td>
<td>Specifies the interface type as T1.</td>
</tr>
<tr>
<td>&lt;slot/port&gt;</td>
<td>Specifies the slot/port number of the first interface in the desired range of interfaces to be configured, followed by a hyphen (-) or a comma (,).</td>
</tr>
<tr>
<td>&lt;slot/port&gt;</td>
<td>Specifies the slot/port number of the last interface in the desired range of interfaces to be configured.</td>
</tr>
</tbody>
</table>

Default Values

No default value is necessary for this command.

Functional Notes

All configuration changes made in this mode will apply to all interfaces in the range specified.

Usage Examples

The following example selects seven consecutive E1 ports for configuration:

ProCurve(config)#interface range e1 3/1-3/7
ProCurve(config-e1 3/1-7)#

The following example selects nonconsecutive E1 ports for configuration:

ProCurve(config)#interface range e1 3/1-2, 3/4-6, 3/8
ProCurve(config-e1 3/1-2, 3/4-6, 3/8)#
**interface tunnel <id>**

Use the `interface tunnel` command to create a virtual tunnel interface and enters the Tunnel Configuration command set. See *Tunnel Configuration Command Set* on page 1053 for details. Use the `no` form of this command to delete a configured virtual tunnel interface.

**Syntax Description**

<id>  
Specifies the numerical tunnel interface identifying label (valid range: 1 to 1024).

**Default Values**

By default, there are no configured tunnel interfaces.

**Functional Notes**

A tunnel may become operational only under the following conditions:

1. The tunnel must have an IP address defined.
2. A valid source address or interface must be configured.
3. A valid destination address must be configured.
4. The physical interface used as the source for the tunnel must be operational.
5. The tunnel can not be in a recursive routing loop.
6. If keepalives are enabled, keepalive processing must be successful. See `keepalive <period> <retries>` on page 1100 for details.

**Technology Review**

A tunnel interface enables standard point-to-point encapsulation between two links. Each endpoint must have a unique tunnel configured. Tunneling allows an arbitrary payload protocol to be encapsulated within a delivery protocol to provide point-to-point communications. The tunnel alone does not provide encryption or any other means of high security. The tunnel interface is not a physical interface, so traffic will be routed to the tunnel by the routing engine for encapsulation or decapsulation and typically forwarded out a physical interface. A common tunnel implementation is the use of a GRE tunnel to transport IP multicast traffic, such as that used by routing protocols across a link that only has IP unicast connectivity (such as IPSec).

**Usage Examples**

The following example creates a tunnel interface (labeled 1) and enters the Tunnel Configuration mode:

```
ProCurve(config)#interface tunnel 1  
ProCurve(config-tunnel 1)#
```
ip access-list extended <listname>

Use the `ip access-list extended` command to create an empty access list and enter the extended access-list command set. Use the `no` form of this command to delete an access list and all the entries contained in it. For more information on using access lists with the SROS firewall, refer to `ip policy-class <polyciname>` on page 400. The following lists the complete syntax for the `ip access-list extended` commands:

```
ip access-list extended <listname>
    <action> <protocol> <source> <source port> <destination> <destination port>
```

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;listname&gt;</code></td>
<td>Identifies the configured access list using an alphanumeric descriptor. All access list descriptors are case-sensitive.</td>
</tr>
</tbody>
</table>
| `<action>` | `permit` Permits entry to the routing system for specified packets.  
              `deny` Denies entry to the routing system for specified packets.  
              `remark` Associates a descriptive tag (up to 80 alphanumeric characters enclosed in quotation marks) to the access list. Enter a functional description for the list such as “This list blocks all outbound Web traffic.” |
| `<protocol>` | Specifies the data protocol `ip`, `icmp`, `tcp`, `udp`, `ahp`, `esp`, `gre`, or a specific protocol (0 to 255). |
| `<source>` | Specifies the source used for packet matching. Sources can be expressed in one of four ways:  
               1. Using the keyword `any` to match any IP address.  
               2. Using `host <A.B.C.D>` to specify a single host address.  
               3. Using the `<A.B.C.D> <wildcard>` format to match all IP addresses in a range. Wildcard masks work in reverse logic from subnet masks. Specifying 255 in any octet of the wildcard mask equates to a “don’t care”. For example, entering `192.168.0.0. 0.0.0.255` selects all traffic from the 192.168.0.0/24 network.  
               4. Using the keyword `hostname` to match based on a DNS name. The unit must be configured with DNS servers for this function to work. |
| `<source port>` | Optional. The source port is used only when `<protocol>` is `tcp` or `udp`. The following keywords and port numbers are supported for the `<source port>` field:  
               `any` Matches any destination port.  
               `eq <port number>` Matches only packets that contain the specified port number.  
               `gt <port number>` Matches only packets with a port number higher than the one listed.  
               `lt <port number>` Matches only packets with a port number lower than the one listed.  
               `neq <port number>` Matches only packets that do not contain the specified port number.  
               `range <port number>` Matches only packets that contain a port number in the specified range. |
### <port number>

Specifies the port number used by TCP or UDP to pass information to upper layers using the following syntax: `<port number>`. All ports below 1024 are considered well-known ports and are controlled by the Internet Assigned Numbers Authority (IANA). All ports above 1024 are dynamically assigned ports that include registered ports for vendor-specific applications.

### <port name>

The following UDP port numbers can be specified using the associated names:

- **biff** (Port 512)  
- **bootpc** (Port 68)  
- **bootps** (Port 67)  
- **discard** (Port 9)  
- **dnsix** (Port 195)  
- **domain** (Port 53)  
- **echo** (Port 7)  
- **isakmp** (Port 500)  
- **mobile-ip** (Port 434)  
- **nameserver** (Port 42)  
- **netbios-dgm** (Port 138)  
- **netbios-ns** (Port 137)  
- **netbios-ss** (Port 139)  
- **nameserver** (Port 42)  
- **ntp** (Port 123)  
- **pim-auto-rp** (Port 496)  
- **rip** (Port 520)  
- **snmp** (Port 161)  
- **snmptrap** (Port 162)  
- **sunrpc** (Port 111)  
- **syslog** (Port 514)  
- **tacacs** (Port 49)  
- **tftp** (Port 69)  
- **time** (Port 37)  
- **who** (Port 513)  
- **xdmcp** (Port 177)

The following TCP port numbers can be specified using the associated names:

- **bgp** (Port 179)  
- **chargen** (Port 19)  
- **cmd** (Port 514)  
- **daytime** (Port 13)  
- **discard** (Port 9)  
- **domain** (Port 53)  
- **echo** (Port 7)  
- **exec** (Port 512)  
- **finger** (Port 79)  
- **ftp** (Port 21)  
- **gopher** (Port 70)  
- **hostname** (Port 101)  
- **ident** (Port 113)  
- **irc** (Port 194)  
- **klogin** (Port 543)  
- **kshell** (Port 544)  
- **login** (Port 513)  
- **lpd** (Port 515)  
- **nntp** (Port 119)  
- **pop2** (Port 109)  
- **pop3** (Port 110)  
- **smtp** (Port 25)  
- **sunrpc** (Port 111)  
- **syslog** (Port 514)  
- **tacacs** (Port 49)  
- **talk** (Port 517)  
- **telnet** (Port 23)  
- **time** (Port 37)  
- **uucp** (Port 540)  
- **whois** (Port 43)  
- **www** (Port 80)
<destination> Specifies the destination used for packet matching. Destinations can be expressed in one of four ways:
1. Using the keyword any to match any IP address.
2. Using host <A.B.C.D> to specify a single host address.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a range. Wildcard masks work in reverse logic from subnet masks. Specifying 255 in any octet of the wildcard mask equates to a “don’t care”.
4. Using the keyword hostname to match based on a DNS name. The unit must be configured with DNS servers for this function to work.

<destination port> Optional. Specifies the destination port. Only valid when <protocol> is tcp or udp. The same keywords and port numbers/names used for the <source port> field are valid for the <destination port> field. Refer to previously listed <source port> for more details.

<destination port> Optional. Specifies the destination port. Only valid when <protocol> is tcp or udp. (Refer to previously listed <source port> for more details.)

<icmp-type> Optional. Filters packets using ICMP defined (and numbered) messages carried in IP datagrams (used to send error and control information). Valid range is 0 to 255.

<icmp-code> Optional. Filters ICMP packets that are filtered using the ICMP message type (using the <icmp-type> keyword) can also be filtered using the ICMP message code (valid range: 0 to 255). An <icmp-type> must be specified when entering an <icmp-code>.

<icmp-message> Optional. Filters packets using ICMP descriptive message rather than the corresponding type and code associations.

Default Values
By default, all SROS security features are disabled and there are no configured access lists.

Functional Notes
Access control lists (ACLs) are used as packet selectors by other SROS features (firewall, VPN, QoS); by themselves they do nothing. ACLs are composed of an ordered list of entries with an implicit deny all at the end of each list. An ACL entry contains two parts: an action (permit or deny) and a packet pattern. A permit ACL is used to allow packets (meeting the specified pattern) to enter the router system. A deny ACL advances the SROS to the next access policy entry. The SROS provides two types of ACLs: standard and extended. Standard ACLs match based on the source of the packet. Extended ACLs match based on the source and destination of the packet.

ACLs are performed in order from the top of the list down. Generally, the most specific entries should be at the top and the more general at the bottom.
The following commands are contained in the access-list extended:

**remark**
Use the remark command to associate a descriptive tag (up to 80 alphanumeric characters encased in quotation marks) to the access-list. Enter a functional description for the list such as “This list blocks all outbound web traffic”.

**log**
Using the log keyword logs a message (if `debug access-list` is enabled for this access list) when the access list finds a packet match.

**Usage Examples**
The following example creates an access list `AllowIKE` to allow all IKE (UDP Port 500) packets from the 192.168.22.55/24 network:

```plaintext
ProCurve(config)#ip access-list extended AllowIKE
ProCurve(config-ext-nacl)#permit udp 192.168.22.55.0 0.0.0.255 eq 500 any eq 500
```

For more details, refer to the *ProCurve Secure Router 7000dl Series Management and Configuration Guide* for information regarding access-list configuration.
Use the `ip access-list standard` command to create an empty access list and enter the standard access-list command set. Use the `no` form of this command to delete an access list and all the entries contained in it. For more information on using access lists with the SROS firewall, refer to `ip policy-class <policyname>` on page 400. The following lists the complete syntax for the `ip access-list standard` commands:

```
ip access-list standard <listname>
   <action> <source>
```

**Syntax Description**

- `<listname>`
  Identifies the configured access list using an alphanumeric descriptor. All access list descriptors are case-sensitive.

- `<action>`
  - `permit`
    Permits entry to the routing system for specified packets.
  - `deny`
    Denies entry to the routing system for specified packets.
  - `remark`
    Associates a descriptive tag (up to 80 alphanumeric characters enclosed in quotation marks) to the access list. Enter a functional description for the list such as “This list blocks all outbound Web traffic.”

- `<source>`
  Specifies the source used for packet matching. Sources can be expressed in one of four ways:
  1. Using the keyword `any` to match any IP address.
  2. Using `host <A.B.C.D>` to specify a single host address.
  3. Using the `<A.B.C.D> <wildcard>` format to match all IP addresses in a range. Wildcard masks work in reverse logic from subnet masks. Specifying 255 in any octet of the wildcard mask equates to a “don’t care”.
  4. Using the keyword `hostname` to match based on a DNS name. The unit must be configured with DNS servers for this function to work.

**Default Values**

By default, all SROS security features are disabled and there are no configured access lists.

**Functional Notes**

Access control lists (ACLs) are used as packet selectors by different SROS features (firewall, VPN, QoS); by themselves they do nothing. ACLs are composed of an ordered list of entries with an implicit `deny all` at the end of each list. An ACL entry contains two parts: an action (`permit` or `deny`) and a packet pattern. A `permit` ACL is used to match packets (meeting the specified pattern) to enter the router system. A `deny` ACL advances the SROS to the next access policy entry. The SROS provides two types of ACLs: standard and extended. Standard ACLs match based on the source of the packet. Extended ACLs match based on the source and destination of the packet.

ACLs are performed in order from the top of the list down. Generally, the most specific entries should be at the top and the more general at the bottom.
The following commands are contained in the **access-list standard:**

**remark**
Use the remark command to associate a descriptive tag (up to 80 alphanumeric characters encased in quotation marks) to the access-list. Enter a functional description for the list such as “This list blocks all outbound web traffic”.

**log**
Use the log keyword to log a message (if debug access-list is enabled for this access list) when the access list finds a packet match.

### Usage Examples

The following example creates an access list **UnTrusted** to deny all packets from the 192.168.22.248/30 network:

```
ProCurve(config)#ip access-list standard UnTrusted
ProCurve(config-std-nacl)#deny 192.168.22.248 0.0.0.3
```

For more details, refer to the *ProCurve Secure Router 7000dl Series Management and Configuration Guide* for information regarding access-list configuration.
**ip as-path-list <listname>**

Use the `ip as-path-list` command to create IP autonomous system (AS) path lists for route-map use. Use the `no` form of this command to delete the AS path list.

**Syntax Description**

| <listname> | Specifies the name of the AS-path list. See *AS Path List Command Set* on page 1178 for more information on the available options. |

**Default Values**

By default, no as-path lists are defined.

**Functional Notes**

AS path lists are a type of route filter that permits or denies BGP routes based on the AS_PATH attribute. AS path lists define a list of AS specifications (to permit or deny traffic) which can then be referenced in a route map. See the Usage Examples section below.

**Usage Examples**

The following example creates the AS path list `list5` and enters the IP `as-path-list` command mode:

```
ProCurve(config)#ip as-path-list list5
ProCurve(config-as-path-list)#
```
ip classless

Use the `ip classless` command to forward classless packets to the best supernet route available. A classless packet is a packet addressed for delivery to a subnet of a network with no default network route.

**Syntax Description**
No subcommands.

**Default Values**
By default, this command is enabled.

**Functional Notes**
SROS products only function in classless mode. You cannot disable this feature.

**Usage Examples**
The following example enables the system to forward classless packets:

```
ProCurve(config)#ip classless
```
ip community-list <listname>

Use the `ip community-list` command to create a community list for BGP route map use. Use the `no` form of this command to delete a community list.

**Syntax Description**

| <listname> | Specifies the name of the community list to use in the community list attribute for BGP routes. See *Community List Command Set* on page 1232 for more information on the available options. |

**Default Values**

By default, this command is disabled.

**Usage Examples**

The following example creates the community list and enters the `community-list` command mode:

```
ProCurve(config)#ip community-list MyList
```
ip crypto

Use the **ip crypto** command to enable SROS VPN functionality and allow crypto maps to be added to interfaces. Use the **no** form of this command to disable the VPN functionality.

**Note**  
Disabling the SROS security features (using the **no ip crypto** command) does not affect VPN configuration settings (with the exception of the removal of all crypto maps from the interfaces). All other configuration parameters will remain intact, and VPN functionality will be disabled.

For VPN configuration example scripts, refer to the technical support note *VPN Configuration Guide* located on the ProCurve SROS Documentation CD provided with your unit.

**Syntax Description**

**fast-failover**  
Optional. This setting is used when the same crypto map is applied to two different egress interfaces. It allows the quick deletion of IKE and IPSec SAs when the default route policy-class changes.

**Default Values**

By default, all SROS VPN functionality is disabled.

**Functional Notes**

VPN-related settings will not go into effect until you enable VPN functionality using the **ip crypto** command. The SROS allows you to perform all VPN-related configuration prior to enabling **ip crypto**, with the exception of assigning a **crypto map** to an interface. The **no ip crypto** command removes all crypto maps from the interfaces. Enabling **ip crypto** enables the IKE server on UDP port 500. The **no** form of this command disables the IKE server on UDP port 500.

**Usage Examples**

The following example enables VPN functionality:

ProCurve(config)#ip crypto
**ip default-gateway** `<ip address>`

Use the `ip default-gateway` command to specify a default gateway if (and only if) IP routing is NOT enabled on the unit. Use the `ip route` command to add a default route to the route table when using IP routing functionality. See `ip route <ip address> <subnet mask> [interface or ip address] [null 0] <administrative distance>` on page 412 for more information.

**Syntax Description**

| `<ip address>` | Specifies the default gateway IP address in the form of dotted decimal notation (example: 192.22.71.50). |

**Default Values**

By default, there is no configured default-gateway.

**Functional Notes**

Only use the `ip default-gateway` when IP routing is disabled on the router. For all other cases, use the `ip route 0.0.0.0 0.0.0.0 <ip address>` command.

**Usage Examples**

The following example disables IP routing and configures a default gateway for 192.22.71.50:

```
ProCurve(config)#no ip routing
ProCurve(config)#ip default gateway 192.22.71.50
```
**ip dhcp-server database local**

Use the `ip dhcp-server database local` command to configure a DHCP database agent with local bindings. Use the `no` form of this command to disable this option.

**Syntax Description**

No subcommands.

**Default Values**

No default values.

**Usage Examples**

The following example configures the DHCP database agent with local bindings:

```
ProCurve(config)#ip dhcp-server database local
```
**ip dhcp-server excluded-address** `<start ip> <end ip>`

Use the `ip dhcp-server excluded-address` command to specify IP addresses that cannot be assigned to DHCP clients. Use the `no` form of this command to remove a configured IP address restriction.

**Syntax Description**

| `<start ip>` | Specifies the lowest IP address (using dotted decimal notation) in the range OR a single IP address to be excluded. |
| `<end ip>`   | Optional. Specifies the highest IP address (using dotted decimal notation) in the range. This field is not required when specifying a single IP address. |

**Default Values**
By default, there are no excluded IP addresses.

**Functional Notes**
The SROS DHCP server (by default) allows all IP addresses for the DHCP pool to be assigned to requesting clients. This command is used to ensure that the specified address is never assigned by the DHCP server. When static addressed hosts are present in the network, it is helpful to exclude the IP addresses of the host from the DHCP IP address pool. This will avoid IP address overlap.

**Usage Examples**
The following example excludes an IP address of 172.22.5.100 and the range 172.22.5.200 through 172.22.5.250:

```
ProCurve(config)#ip dhcp-server excluded-address 172.22.5.100
ProCurve(config)#ip dhcp-server excluded-address 172.22.5.200 172.22.5.250
```
ip dhcp-server ping packets <#packets>

Use the ip dhcp-server ping packets command to specify the number of ping packets the DHCP server will transmit before assigning an IP address to a requesting DHCP client. Transmitting ping packets verifies that no other hosts on the network are currently configured with the specified IP address. Use the no form of this command to prevent the DHCP server from using ping packets as part of the IP address assignment process.

Syntax Description

<#packets> Specifies the number of DHCP ping packets sent on the network before assigning the IP address to a requesting DHCP client

Default Values

By default, the number of DHCP server ping packets is set to 2 packets.

Functional Notes

Before assigning an IP address to a requesting client, the SROS DHCP server transmits a ping packet on the network to verify there are no other network hosts already configured with the specified address. If the DHCP server receives no reply, the IP address is assigned to the requesting client and added to the DHCP database as an assigned address. Configuring the ip dhcp-server ping packets command with a value of 0 prevents the DHCP server from using ping packets as part of the IP address assignment process.

Usage Examples

The following example configures the DHCP server to transmit 4 ping packets before assigning an address:

ProCurve(config)#ip dhcp-server ping packets 4
ip dhcp-server ping timeout <milliseconds>

Use the **ip dhcp-server ping timeout** command to specify the interval (in milliseconds) the DHCP server will wait for a response to a transmitted DHCP ping packet. The DHCP server transmits ping packets before assigning an IP address to a requesting DHCP client. Transmitting ping packets verifies that no other hosts on the network are currently configured with the specified IP address. Use the **no** form of this command to return to the default timeout interval.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;milliseconds&gt;</th>
<th>Specifies the number of milliseconds (valid range: 1 to 1,000) the DHCP server will wait for a response to a transmitted DHCP ping packet.</th>
</tr>
</thead>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>&lt;milliseconds&gt;</th>
<th>500 milliseconds</th>
</tr>
</thead>
</table>

**Functional Notes**

Before assigning an IP address to a requesting client, the SROS DHCP server transmits a ping packet on the network to verify there are no other network hosts already configured with the specified address. If the DHCP server receives no reply, the IP address is assigned to the requesting client and added to the DHCP database as an assigned address.

**Usage Examples**

The following example configures the DHCP server to wait 900 milliseconds for a response to a transmitted DHCP ping packet before considering the ping a failure:

```
ProCurve(config)#ip dhcp-server ping timeout 900
```
ip dhcp-server pool <name>

Use the `ip dhcp-server pool` command to create a DHCP address pool and enter the DHCP pool. Use the `no` form of this command to remove a configured DHCP address pool. See the section `DHCP Pool Command Set` on page 1279 for more information.

**Syntax Description**

| <name> | Identifies the configured DHCP server address pool by alphanumeric string (up to 32 characters in length) (example SALES). |

**Default Values**

By default, there are no configured DHCP address pools.

**Functional Notes**

Use the `ip dhcp-server pool` to create multiple DHCP server address pools for various segments of the network. Multiple address pools can be created to service different segments of the network with tailored configurations.

**Usage Examples**

The following example creates a DHCP server address pool (labeled SALES) and enters the DHCP server pool:

```
ProCurve(config)#ip dhcp-server pool SALES
ProCurve(config-dhcp)#
```
ip domain-lookup

Use the ip domain-lookup command to enable the IP DNS (domain naming system), allowing DNS-based host translation (name-to-address). Use the no form of this command to disable DNS.

Syntax Description
No subcommands.

Default Values
By default, this command is enabled.

Functional Notes
Use the ip domain-lookup command to enable the DNS client in the router. This will allow the user to input web addresses instead of IP addresses for applications such as ping, Telnet, and traceroute.

Usage Examples
The following example enables DNS:

ProCurve(config)#ip domain-lookup
**ip domain-name <name>**

Use the **ip domain-name** command to define a default IP domain name to be used by the SROS to resolve host names. Use the **no** form of this command to disable this function.

**Syntax Description**

| <name> | Default IP domain name used to resolve unqualified host names. Do not include the initial period that separates the unresolved name from the default domain name. |

**Default Values**

By default, this command is disabled.

**Functional Notes**

Use the **ip domain-name** command to set a default name which will be used to complete any IP host name that is invalid (i.e., any name that is not recognized by the name-server). When this command is enabled, any IP host name that is not initially recognized will have the **ip domain-name** appended to it and the request will be resent.

**Usage Examples**

The following example defines **procurve** as the default domain name:

```
ProCurve(config)#ip domain-name procurve
```
### ip domain-proxy

Use the **ip domain-proxy** command to enable DNS proxy for the router. This enables the router to act as a proxy for other units on the network.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

When this command is enabled, incoming DNS requests will be handled by the router. It will first search its host table for the query, and if it is not found there the request will be forwarded to the servers configured with the **ip name-server** command.

**Usage Examples**

The following example enables DNS proxy:

```
ProCurve(config)#ip domain-proxy
```
ip firewall

Use the ip firewall command to enable SROS security features including access control policies and lists, Network Address Translation (NAT), and the stateful inspection firewall. Use the no form of this command to disable the security functionality.

**Note**

Disabling the SROS security features (using the no ip firewall command) does not affect security configuration. All configuration parameters will remain intact, but no security data processing will be attempted.

Regarding the use of IKE negotiation for VPN with ip firewall enabled, there can be up to six channel groups with 2-8 interfaces per group. Dynamic protocols are not yet supported (only static). A physical interface can be a member of only one channel-group.

**Syntax Description**

No subcommands.

**Default Values**

By default, all SROS security features are disabled.

**Functional Notes**

This command enables firewall processing for all interfaces with a configured policy class. Firewall processing consists of the following functions:

- **Attack Protection**: Detects and discards traffic that matches profiles of known networking exploits or attacks.
- **Session Initiation Control**: Allows only sessions that match traffic patterns permitted by access-control policies to be initiated through the router.
- **Ongoing Session Monitoring and Processing**: Each session that has been allowed through the router is monitored for any irregularities that match patterns of known attacks or exploits. This traffic will be dropped. Also, if NAT is configured, the firewall modifies all traffic associated with the session according to the translation rules defined in NAT access-policies. Finally, if sessions are inactive for a user-specified amount of time, the session will be closed by the firewall.
- **Application Specific Processing**: Certain applications need special handling to work correctly in the presence of a firewall. SROS uses ALGs (application-level gateways) for these applications. The SROS includes several security features to provide controlled access to your network. The following features are available when security is enabled (using the ip firewall command):

1. **Stateful Inspection Firewall**
   The SROS (and your unit) act as an application-level gateway and employ a stateful inspection firewall that protects an organization's network from common cyber attacks including TCP syn-flooding, IP spoofing, ICMP redirect, land attacks, ping-of-death, and IP reassembly problems. In addition, further security is added with use of Network Address Translation (NAT) and Port Address Translation (PAT) capability.
2. Access Policies (ACPs)
SROS access control policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded.

3. Access Lists (ACLs)
Access control lists are used as packet selectors by ACPs; by themselves they do nothing. ACLs are composed of an ordered list of entries. Each entry contains two parts: an action (permit or deny) and a packet pattern. A permit ACL is used to permit packets (meeting the specified pattern) to enter the router system. A deny ACL advances the SROS to the next access policy entry. The SROS provides two types of ACLs: standard and extended. Standard ACLs allow source IP address packet patterns only. Extended ACLs may specify patterns using most fields in the IP header and the TCP or UDP header.

Usage Examples
The following example enables the SROS security features:

ProCurve(config)#ip firewall

Technology Review
Concepts:
Access control using the SROS firewall has two fundamental parts: Access Control Lists (ACLs) and Access Policy Classes (ACPs). ACLs are used as packet selectors by other SROS systems; by themselves they do nothing. ACPs consist of a selector (ACL) and an action (allow, discard, NAT). ACPs integrate both allow and discard policies with NAT. ACPs have no effect until they are assigned to a network interface.

Both ACLs and ACPs are order dependent. When a packet is evaluated, the matching engine begins with the first entry in the list and progresses through the entries until it finds a match. The first entry that matches is executed.

Packet Flow:
Case 1: Packets from interfaces with a configured policy class to any other interface

ACPs are applied when packets are received on an interface. If an interface has not been assigned a policy class, by default it will allow all received traffic to pass through. If an interface has been assigned a policy class but the firewall has not been enabled with the `ip firewall` command, traffic will flow normally from this interface with no firewall processing.

Case 2: Packets that travel in and out a single interface with a configured policy class

These packets are processed through the ACPs as if they are destined for another interface (identical to Case 1).

Case 3: Packets from interfaces without a configured policy class to interfaces with one

These packets are routed normally and are not processed by the firewall. The `ip firewall` command has no effect on this traffic.

Case 4: Packets from interfaces without a configured policy class to other interfaces without a configured policy class

This traffic is routed normally. The `ip firewall` command has no effect on this traffic.

Attack Protection:

When the `ip firewall` command is enabled, firewall attack protection is enabled. The SROS blocks traffic (matching patterns of known networking exploits) from traveling through the device. For some of these attacks, the user may manually disable checking/blocking while other attack checks are always on anytime the firewall is enabled.

The table (on the following pages) outlines the types of traffic discarded by the Firewall Attack Protection Engine. Many attacks use similar invalid traffic patterns; therefore attacks other than the examples listed below may also be blocked by the firewall. To determine if a specific attack is blocked by the SROS firewall, please contact technical support.

<table>
<thead>
<tr>
<th>Invalid Traffic Pattern</th>
<th>Manually Enabled?</th>
<th>OS Firewall Response</th>
<th>Common Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Larger than allowed packets</td>
<td>No</td>
<td>Any packets that are longer than those defined by standards will be dropped.</td>
<td>Ping of Death</td>
</tr>
<tr>
<td>Fragmented IP packets that produce errors when attempting to reassemble</td>
<td>No</td>
<td>The firewall intercepts all fragments for an IP packet and attempts to reassemble them before forwarding to destination. If any problems or errors are found during reassembly, the fragments are dropped.</td>
<td>SynDrop, TearDrop, OpenTear, Nestea, Targa, Newtare, Bonk, Boink</td>
</tr>
<tr>
<td>Invalid Traffic Pattern</td>
<td>Manually Enabled?</td>
<td>OS Firewall Response</td>
<td>Common Attacks</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Smurf Attack</td>
<td>No</td>
<td>The firewall will drop any ping responses that are not part of an active session.</td>
<td>Smurf Attack</td>
</tr>
<tr>
<td>IP Spoofing</td>
<td>No</td>
<td>The firewall will drop any packets with a source IP address that appears to be spoofed. The IP route table is used to determine if a path to the source address is known (out of the interface from which the packet was received). For example, if a packet with a source IP address of 10.10.10.1 is received on interface fr 1.16 and no route to 10.10.10.1 (through interface fr 1.16) exists in the route table, the packet is dropped.</td>
<td>IP Spoofing</td>
</tr>
<tr>
<td>ICMP Control Message Floods and Attacks</td>
<td>No</td>
<td>The following types of ICMP packets are allowed through the firewall: echo, echo-reply, TTL expired, dest. Unreachable, and quench. These ICMP messages are only allowed if they appear to be in response to a valid session. All others are discarded.</td>
<td>Twinge</td>
</tr>
<tr>
<td>Attacks that send TCP URG packets</td>
<td>Yes</td>
<td>Any TCP packets that have the URG flag set are discarded by the firewall.</td>
<td>Winnuke, TCP XMAS Scan</td>
</tr>
<tr>
<td>Falsified IP Header Attacks</td>
<td>No</td>
<td>The firewall verifies that the packet’s actual length matches the length indicated in the IP header. If it does not, the packet is dropped.</td>
<td>Jolt/Jolt2</td>
</tr>
<tr>
<td>Echo</td>
<td>No</td>
<td>All UDP echo packets are discarded by the firewall.</td>
<td>Char Gen</td>
</tr>
<tr>
<td>Land Attack</td>
<td>No</td>
<td>Any packets with the same source and destination IP addresses are discarded.</td>
<td>Land Attack</td>
</tr>
<tr>
<td>Broadcast Source IP</td>
<td>No</td>
<td>Packets with a broadcast source IP address are discarded.</td>
<td></td>
</tr>
<tr>
<td>Invalid TCP Initiation Requests</td>
<td>No</td>
<td>TCP SYN packets that have ack, urg rst, or fin flags set are discarded.</td>
<td></td>
</tr>
<tr>
<td>Invalid TCP Segment Number</td>
<td>No</td>
<td>The sequence numbers for every active TCP session are maintained in the firewall session database. If the firewall received a segment with an unexpected (or invalid) sequence number, the packet is dropped.</td>
<td></td>
</tr>
<tr>
<td>IP Source Route Option</td>
<td>No</td>
<td>All IP packets containing the IP source route option are dropped.</td>
<td></td>
</tr>
</tbody>
</table>
Application Specific Processing:

The following applications and protocols require special processing to operate concurrently with NAT/firewall functionality. The SROS firewall includes ALGs for handling these applications and protocols:

- AOL Instant Messenger (AIM®)
- VPN ALGS: ESP and IKE
- FTP
- H.323: H.245 Q.931 ASN1 PER decoding and Encoding
- ICQ®
- IRC
- Microsoft® Games
- Net2Phone
- PPTP
- Quake®
- Real-Time Streaming Protocol
- SMTP
- HTTP
- CUseeme
- SIP
- L2TP
- PcAnywhere™
- SQL
- Microsoft Gaming Zone

To determine if a specific application requires special processing, contact technical support.
**ip firewall alg [ftp | h323 | pptp | sip]**

Use the `ip firewall alg` command to enable the application level gateway (ALG) for a particular application. Use the `no` form of this command to disable ALG for the application.

### Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftp</td>
<td>Enables the FTP ALG.</td>
</tr>
<tr>
<td>h323</td>
<td>Enables the H323 ALG. H.323 is a generic recommendation from the ITU that sets standards for multimedia communications over networks without guaranteed Quality of Service (QoS)</td>
</tr>
<tr>
<td>pptp</td>
<td>Enables the PPTP ALG.</td>
</tr>
<tr>
<td>sip</td>
<td>Enables the SIP ALG.</td>
</tr>
</tbody>
</table>

### Default Values

By default, the ALG for FTP, H323, PPTP, and SIP are enabled.

### Functional Notes

Enabling the Application Layer Gateway (ALG) for a specific protocol gives the firewall additional information about that complex protocol and causes the firewall to perform additional processing for packets of that protocol. When the ALG is disabled, the firewall treats the complex protocol as any other simple protocol. The firewall needs no special knowledge to work well with simple protocols.

### Warning

Disabling the IP firewall ALG may cause the firewall to block some of the traffic for the specified protocol.

### Session Initiation Protocol (SIP) ALG Information

By default, the SROS SIP ALG is enabled. This ALG allows the firewall to examine the ALL SIP packets it identifies and maintain knowledge of SIP transmissions on the network based on the SIP header. The SIP ALG requires the use of the SIP stack and the SIP proxy server in order to properly route SIP calls and maintain the SIP information. When the SIP ALG is enabled, the SIP stack and SIP proxy server are automatically enabled. For proper SIP operation, the firewall must also be configured to allow for dynamic holes for the RTP/RTCP traffic associated with SIP calls between User Agents (UAs). This functionality must be manually enabled using the `ip rtp firewall-traversal` command.

To completely disable SIP operation in the SROS, the following commands should be entered: `no ip firewallalg sip`, `no ip sip`, `no ip sip proxy`, and `no ip rtp firewall-traversal`. The `no ip firewall alg sip` command disables the SIP ALG. The `no ip sip` command disables the SIP stack and frees all memory allocated to the stack. The `no ip sip proxy` command disables the SIP proxy server. This command is not necessary to disable SIP functionality (because the `no ip sip` command effectively shuts the proxy server down by disabling the stack), but should be entered for a cleaner configuration.
Usage Examples

The following example disables ALG for FTP:

ProCurve(config)#no ip firewall alg ftp

Technology Review

SIP is one protocol in a suite of protocols that was designed to replace H.323 for IP telephony. SIP operates in Layer 7 of the OSI model (Application layer) to create, modify, and terminate sessions between nodes. SIP not only provides recommendations for IP telephony, but multimedia distribution and conferences as well. SIP version 1.0 was defined in RFC2453, and was refined to SIP version 2.0 in RFC3261.

SIP operations occur between SIP UAs and SIP servers. Types of SIP servers include proxy, redirect, registrar, and presence. The part of a SIP UA that sends messages is known as the User Agent Client (UAC). The part of a SIP UA that receives messages is known as a User Agent Server (UAS).

SIP was originally designed for use over UDP. SIP servers, by default, listen on port 5060. Due to security concerns, SIP is now transitioning to TCP and Transport Layer Security (TLS). SIP servers using TLS-over-TCP listen on port 5061. SIP UAs listen on a range of ports. The listening UDP port can be manually changed using the `ip firewall alg sip udp` command.

SIP uses the Session Description Protocol (SDP) to format the SIP message body in order to negotiate a Real-time Transport Protocol (RTP)/Real-time Transport Control Protocol (RTCP) connection between two or more UAs. The ports used for this will always be selected in a pair, with the even port used for RTP and the odd port for RTCP. SIP, because it uses SDP and RTP, causes many problems for standard firewalls. Neither SIP nor RTP are guaranteed to be symmetric, thus causing problems for stateful-inspection firewalls that rely on symmetric flows. SIP and SDP carry IP addresses and ports embedded in the packet and standard NAT implementations only modify the IP and TCP/UDP headers. A true SIP ALG is required to both modify the packets as needed for NAT, but also open holes in the firewall as needed for traffic flow based on the information carried in the SIP header.

Enabling the SROS SIP ALG (using the `ip firewall alg sip` command) configures the firewall to examine the ALL SIP packets it identifies and maintain knowledge of SIP transmissions on the network. Since SIP packet headers include port information for the call setup, the ALG must intelligently read the packets and remember the information. To accomplish this, the SIP ALG enables two other SIP functions, the SIP stack (`ip sip` command) and the SIP proxy server (`ip sip proxy`). This operation allows dynamic configuration of the SIP network, because UAs on the network do not need to be manually added to the router’s location database. If there is a SIP node on the network that transmits traffic, the router will identify the traffic as SIP traffic and maintain the appropriate information. This mode can be considered “transparent-proxy.” A ProCurve Secure Router running in transparent-proxy mode can be added to a previously configured network (without requiring specific SIP location database configuration) and can be expected to intelligently route SIP packets.

As an alternative to running in transparent-proxy mode, the Secure Router SIP proxy server can be configured to restrict SIP knowledge to only nodes entered into the location database (using the `no ip firewall alg sip` command). Just as a router uses an IP route table to determine the destination for IP packets it receives, a SIP proxy server uses the location database to determine the appropriate destination UA. Manually configuring the location database can be cumbersome for a large SIP network. To avoid losing pertinent information in the event of a power loss, use the `ip sip database local` command to create a persistent database on the local router memory that is maintained across a power loss.
ip firewall alg sip udp <port#>

Use the `ip firewall alg sip udp` command to configure the user datagram protocol (UDP) port(s) for Session Initiation Protocol (SIP) application-level gateways (ALG). For more details on SIP functionality in the SROS, refer to the `Functional Notes` and `Technology Review` sections of the command `ip firewall alg [ftp | h323 | pptp | sip]` on page 374. Use the `no` form of this command to return to the default settings.

**Syntax Description**

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>udp &lt;port#&gt;</td>
<td>Sets the UDP port. Valid range: 1 to 65,535. Multiple UDP ports can be entered.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the ALG for SIP is enabled and the UDP port is set to 5060.

**Usage Examples**

The following example enables port 1020 for UDP:

```
ProCurve(config)#ip firewall alg sip udp 1020
```
**ip firewall attack-log threshold** `<value>`

Use the `ip firewall attack-log threshold` command to specify the number of attack mounting attempts the SROS will identify before generating a log message. Use the `no` form of this command to return to the default threshold.

---

**Syntax Description**

`<value>` Specifies the number of attack mounting attempts the SROS will identify before generating a log message (valid range: 0 to 4,294,967,295).

---

**Default Values**

By default, the `ip firewall attack-log threshold` is set to 100.

---

**Usage Examples**

The following example specifies a threshold of 25 attacks before generating a log message:

```
ProCurve(config)#ip firewall attack-log threshold 25
```
ip firewall check reflexive-traffic

Use the `ip firewall check reflexive-traffic` command to enable the SROS stateful inspection firewall to process traffic from a primary subnet to a secondary subnet on the same interface through the firewall. Use the `no` form of this command to disable this feature.

**Note** The SROS security features must be enabled (using the `ip firewall` command) for the stateful inspection firewall to be activated.

**Syntax Description**
No subcommands.

**Default Values**
All SROS security features are disabled by default until the `ip firewall` command is issued at the Global Configuration prompt. In addition, the reflexive traffic check is disabled until the `ip firewall check reflexive-traffic` command is issued.

**Functional Notes**
This command allows the firewall to process traffic from a primary subnet to a secondary subnet on the same interface through the firewall. If enabled, this traffic will be processed through the access-policy on that interface and any actions specified will be executed on the traffic.

**Usage Examples**
The following example enables the SROS reflexive-traffic check:

```
ProCurve(config)#ip firewall check reflexive-traffic
```
ip firewall check rst-seq

Use the `ip firewall check rst-seq` command to enable TCP reset sequence number checking. Use the `no` form of this command to disable this feature.

**Note** The SROS security features must be enabled (using the `ip firewall` command) for the stateful inspection firewall to be activated.

**Syntax Description**

No subcommands.

**Default Values**

All SROS security features are disabled by default until the `ip firewall` command is issued at the Global Configuration prompt. In addition, TCP reset sequence number checking is disabled until the `ip firewall check rst-seq` command is issued.

**Usage Examples**

The following example enables TCP reset sequence number checking:

```
ProCurve(config)#ip firewall check rst-seq
```
ip firewall check syn-flood

Use the **ip firewall check syn-flood** command to enable the SROS stateful inspection firewall to filter out phony TCP service requests and allow only legitimate requests to pass through. Use the **no** form of this command to disable this feature.

**Note**  
The SROS security features must be enabled (using the **ip firewall** command) for the stateful inspection firewall to be activated.

**Syntax Description**
No subcommands.

**Default Values**
All SROS security features are disabled by default until the **ip firewall** command is issued at the Global Configuration prompt. In addition, the SYN-flood check is disabled until the **ip firewall check syn-flood** command is issued.

**Functional Notes**
SYN Flooding is a well-known denial of service attack on TCP-based services. TCP requires a three-way handshake before actual communications begin between two hosts. A server must allocate resources to process new connection requests that are received. A potential intruder is capable of transmitting large amounts of service requests (in a very short period of time), causing servers to allocate all resources to process the phony incoming requests. Using the **ip firewall check syn-flood** command configures the SROS stateful inspection firewall to filter out phony service requests and allow only legitimate requests to pass through.

**Usage Examples**
The following example enables the SROS syn-flood check:

```
ProCurve(config)#ip firewall check syn-flood
```
ip firewall check winnuke

Use the **ip firewall check winnuke** command to enable the SROS stateful inspection firewall to discard all Out of Band (OOB) data (to protect against WinNuke attacks). Use the **no** form of this command to disable this feature.

```
Note
The SROS security features must be enabled (using the ip firewall command) for the stateful inspection firewall to be activated.
```

Syntax Description

No subcommands.

Default Values

All SROS security features are disabled by default until the **ip firewall** command is issued at the Global Configuration prompt. Issuing the **ip firewall** command enables the WinNuke check.

Functional Notes

WinNuke attack is a well-known denial of service attack on hosts running Microsoft Windows® operating systems. An intruder sends Out of Band (OOB) data over an established connection to a Windows user. Windows cannot properly handle the OOB data and the host reacts unpredictably. Normal shut-down of the hosts will generally return all functionality. Using the **ip firewall check winnuke** command configures the SROS stateful inspection firewall to filter all OOB data to prevent network problems.

Usage Examples

The following example enables the firewall to filter all OOB data:

```
ProCurve(config)#ip firewall check winnuke
```
ip firewall fast-nat-fallover

Use the `ip firewall fast-nat-fallover` command to delete associations on default route policy-class changes.

**Note**

The SROS security features must be enabled (using the `ip firewall` command) for the stateful inspection firewall to be activated.

**Syntax Description**

No subcommands.

**Default Values**

All SROS security features are disabled by default until the `ip firewall` command is issued at the Global Configuration prompt. In addition, the fast NAT fallover is disabled until the `ip firewall fast-nat-fallover` command is issued.

**Usage Examples**

The following example enables `fast-nat-fallover`:

```
ProCurve(config)#ip firewall fast-nat-fallover
```
ip firewall fin-timeout <seconds>

Use the `ip firewall fin-timeout` command to specify the time period allowed for Transport Control Protocol (TCP) FIN. Use the `no` form of this command to return to the default setting.

**Syntax Description**

- `<seconds>`: Specifies the time period allowed for TCP FIN. Range is 0 to 4,294,967,295.

**Default Value**

By default, `ip firewall fin-timeout` is set to 4 seconds.

**Usage Examples**

The following example sets the TCP FIN time period to 120 seconds:

```
ProCurve(config)# ip firewall fin-timeout 120
```
**ip firewall policy-log threshold <value>**

Use the `ip firewall policy-log threshold` command to specify the number of connections required by an access control policy before the SROS will generate a log message. Use the `no` form of this command to return to the default threshold.

**Note**  
The SROS security features must be enabled (using the `ip firewall` command) for the stateful inspection firewall to be activated.

<table>
<thead>
<tr>
<th>Syntax Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;value&gt;</code></td>
</tr>
<tr>
<td>Specifies the number of access policy connections the SROS will identify before generating a log message (valid range: 0 to 4,294,967,295).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the `ip firewall policy-log threshold` is set to 100.

**Usage Examples**

The following example specifies a threshold of 15 connections before generating a log message:

ProCurve(config)#ip firewall policy-log threshold 15
**ip firewall rst-timeout <seconds>**

Use the `ip firewall rst-timeout` command to specify the time period allowed for Transport Control Protocol (TCP) reset. Use the `no` form of this command to return to the default setting.

**Syntax Description**

| `<seconds>` | Specifies the time period allowed for TCP reset. Range is 0 to 4,294,967,295. |

**Default Value**

By default, `ip firewall rst-timeout` is set to 20 settings.

**Usage Examples**

The following example sets the TCP reset time period to 120 seconds:

```
ProCurve(config)#ip firewall rst-timeout 120
```

**Note**

The SROS security features must be enabled (using the `ip firewall` command) for the stateful inspection firewall to be activated.
**ip firewall stealth**

Use the `ip firewall stealth` command to disable TCP reset for denied firewall associations. The stealth setting allows the route to be invisible as a route hop to associated devices.

**Note** The SROS security features must be enabled (using the `ip firewall` command) for the stateful inspection firewall to be activated.

**Syntax Description**

No subcommands.

**Default Values**

All SROS security features are disabled by default until the `ip firewall` command is issued at the Global Configuration prompt. In addition, the stealth option is disabled until the `ip firewall stealth` command is issued.

**Usage Examples**

The following example enables the `stealth` option:

```
ProCurve(config)#ip firewall stealth
```
ip forward-protocol udp <port number>

Use the `ip forward-protocol udp` command to specify the protocols and ports the SROS allows when forwarding broadcast packets. Use the `no` form of this command to disable a specified protocol or port from being forwarded.

**Note**  
The `ip helper` command must be used in conjunction with the `ip forward-protocol` command to configure the SROS to forward UDP broadcast packets.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;port number&gt;</th>
<th>Specifies the UDP traffic type (using source port)</th>
</tr>
</thead>
</table>

The following is the list of UDP port numbers that may be identified using the text name:

- biff (Port 512)
- bootps (Port 67)
- discard (Port 9)
- dnsix (Port 195)
- domain (Port 53)
- echo (Port 7)
- isakmp (Port 500)
- mobileip (Port 434)
- nameserver (Port 42)
- netbios-dgm (Port 138)
- netbios-ns (Port 137)
- netbios-ss (Port 139)
- ntp (Port 123)

Alternately, the `<port number>` may be specified using the following syntax:

- `<0-65535>`. Specifies the port number used by UDP to pass information to upper layers. All ports below 1024 are considered well-known ports and are controlled by the Internet Assigned Numbers Authority (IANA). All ports above 1024 are dynamically assigned ports that include registered ports for vendor-specific applications.

**Default Values**

By default, the SROS forwards broadcast packets for all protocols and ports.

**Functional Notes**

Use this command to configure the SROS to forward UDP packets across the WAN link to allow remote devices to connect to a UDP service on the other side of the WAN link.
Usage Examples

The following example forwards all Domain Name Server broadcast traffic to the DNS server with IP address 192.33.5.99:

ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip helper-address 192.33.5.99
ip ftp access-class <policyname> in

Use the ip ftp access-class in command to assign an access policy to all self-bound File Transfer Protocol (FTP) sessions.

Syntax Description

| <policyname> | Specifies the configured access policy (ACP) to apply to inbound FTP traffic |

Default Values

By default, all ftp access is allowed.

Usage Examples

The following example applies the configured ACP (labeled Inbound_FTP) to inbound FTP traffic:

ProCurve(config)#ip ftp access-class Inbound_FTP in
ip ftp agent

Use the **ip ftp agent** command to enable the file transfer protocol (FTP) agent.

**Syntax Description**

No subcommands.

**Default Values**

By default, the FTP agent is enabled.

**Usage Examples**

The following example enables the IP FTP agent:

```
ProCurve(config)#ip ftp agent
```
ip ftp source-interface <interface>

Use the `ip ftp source-interface` command to use the specified interface’s IP address as the source IP address for FTP traffic transmitted by the unit. Use the `no` form of this command if you do not wish to override the normal source IP address.

Syntax Description

| <interface> | Specifies the interface to be used as the source IP address for FTP traffic. Type `ip ftp source-interface?` for a complete list of valid interfaces. |

Default Values

No default value is necessary for this command.

Functional Notes

This command allows you to override the `sender` field in the IP packet. If you have multiple interfaces in your unit, changing the `sender` tells the receiver where to send replies. This functionality can also be used to allow packets to get through firewalls that would normally block the flow.

Usage Examples

configures the unit to use the `loopback 1` interface as the source IP for FTP traffic:

```
ProCurve(config)#ip ftp source-interface loopback 1
```
**ip host <name> <address>**

Use the `ip host` command to define an IP host name. This allows you to statically map host names and addresses in the host cache. Use the `no` form of this command to remove defined maps.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;name&gt;</code></td>
<td>Name of the host.</td>
</tr>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>IP address associated with this IP host.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the host table is empty.

**Functional Notes**

The name may be any combination of numbers and letters as long as it is not a valid IP address or does not exceed 256 characters.

**Usage Examples**

The following example defines two static mappings:

```
ProCurve(config)#ip host mac 10.2.0.2
ProCurve(config)#ip host dal 172.38.7.12
```
Use the `ip http` command to enable web access to the unit.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-class</td>
<td>Enables HTTP for all incoming connections associated with a specific access list.</td>
</tr>
<tr>
<td>&lt;listname&gt;</td>
<td>Specifies the access list name.</td>
</tr>
<tr>
<td>in</td>
<td>Applies to all incoming connections.</td>
</tr>
<tr>
<td>authentication</td>
<td>Assigns the specified AAA list to HTTP authentication.</td>
</tr>
<tr>
<td>secure-access-class</td>
<td>Applies to all self-bound HTTPS connections.</td>
</tr>
<tr>
<td>secure-server &lt;TCP port&gt;</td>
<td>Enables the SSL server. <code>&lt;TCP port&gt;</code> is optional and is used to specify an alternate TCP port for HTTPS traffic.</td>
</tr>
<tr>
<td>server &lt;TCP port&gt;</td>
<td>Enables the HTTP server connection. <code>&lt;TCP port&gt;</code> is optional and is used to specify an alternate TCP port for HTTP traffic.</td>
</tr>
<tr>
<td>session-limit &lt;limit&gt;</td>
<td>Sets the maximum number of sessions allowed. Valid range is 0 to 100 with 100 as the default.</td>
</tr>
<tr>
<td>session-timeout &lt;time&gt;</td>
<td>Sets the session timeout. Valid range is 10 to 86,400 seconds. The default is 600.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is disabled.

**Usage Examples**

The following example enables web access to the router:

```
ProCurve(config)#ip http server
```
**ip igmp join** `<group-address>`

Use the **ip igmp join** command to instruct the router stack to join a specific group. The stack may join multiple groups.

**Syntax Description**

| `<group-address>` | IP address of a multicast group. |

**Default Values**

No defaults necessary for this command.

**Functional Notes**

This command aids in debugging, allowing the router’s IP stack to connect to and respond on a multicast group. The local stack operates as an IGMP host on the attached segment. In multicast stub applications, the global helper address takes care of forwarding IGMP joins/responses on the upstream interface. The router may respond to ICMP echo requests for the joined groups.

**Usage Examples**

The following example configures the unit to join with the specified multicast group:

```
ProCurve(config)#ip igmp join 172.0.1.50
```
ip load-sharing [per-destination | per-packet]

Use the **ip load-sharing** command to configure whether parallel routes in the route table are used to load-share forwarded packets. If this command is disabled, the route table uses a single "best" route for a given subnet. If this command is enabled, the route table can use multiple "best" routes and alternate between them.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>per-destination</td>
<td>Specifies that the route used for forwarding a packet be based on a hash of the source and destination IP address in the packet.</td>
</tr>
<tr>
<td>per-packet</td>
<td>Specifies that each forwarding route lookup rotates through all the parallel &quot;best&quot; routes. (Parallel routes are defined as routes to the same subnet with the same metrics that only differ by their next hop address.)</td>
</tr>
</tbody>
</table>

**Default Values**

By default, ip load-sharing is disabled.

**Usage Examples**

The following example turns on load-sharing per destination:

```
ProCurve(config)#ip load-sharing per-destination
```

The following example disables load-sharing:

```
ProCurve(config)#no ip load-sharing
```
ip local policy route-map <map-name>

Use this command to specify a route-map for local policy routing on the device. This setting is applied to the local network interface. Use the no form of this command to return to the default route-map.

**Syntax Description**

<map-name> Specify the name of the route-map.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Before a route map can be specified, it must first be defined using the route-map command. See route-map <map-name> [ permit | deny ] <sequence number> on page 455 for more information.

**Usage Examples**

The following example specifies a route-map entitled myMap for local policy routing:

ProCurve(config)#ip local policy route-map myMap
ip mcast-stub helper-address <ip address>

Use the ip mcast-stub helper-address command to specify an IP address toward which IGMP host reports and leave messages are forwarded. This command is used in IP multicast stub applications in conjunction with the ip mcast-stub downstream and ip mcast-stub upstream commands. Use the no form of this command to return to default.

Syntax Description

| <ip address> | IGMP host reports and leave messages are forwarded toward this address. |

Default Values

By default, no helper-address is configured.

Functional Notes

Helper-address is configured globally and applies to all multicast-stub downstream interfaces. The address specified may be the next upstream hop or any upstream address on the distribution tree for the multicast source, up to and including the multicast source. The router selects, from the list of multicast-stub upstream interfaces, the interface on the shortest path to the specified address. The router then proxies, on the selected upstream interface (using an IGMP host function), any host joins/leaves received on the downstream interface(s). The router retransmits these reports with addresses set as if the report originated from the selected upstream interface.

For example, if the router receives multiple joins for a group, it will not send any extra joins out the upstream interface. Also, if it receives a leave, it will not send a leave until it is certain that there are no more subscribers on any downstream interface.

Usage Examples

The following example specifies 172.45.6.99 as the helper-address:

ProCurve(config)#ip mcast-stub helper-address 172.45.6.99
ip multicast-routing

Use the **ip multicast routing** command to enable the multicast router process. The command does not affect other multicast-related configuration. Use the **no** form of this command to disable. Disabling this command prevents multicast forwarding but does not remove other multicast commands and processes.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Usage Examples**

The following example enables multicast functionality:

ProCurve(config)#**ip multicast-routing**
**ip name-server <server-address1-6>**

Use the `ip name-server` command to designate one or more name servers to use for name-to-address resolution. Use the `no` form of this command to remove any addresses previously specified.

**Syntax Description**

- `<server-address1-6>` Specifies up to six name-server addresses.

**Default Values**

By default, no name servers are specified.

**Usage Examples**

The following example specifies host 172.21.111 as the primary name server and host 172.21.1.2 as the secondary server:

```
ProCurve(config)#ip name-server 172.21.1.111 172.21.1.2
```

This command will be reflected in the configuration file as follows:

```
ip name-server 172.21.1.111 172.21.1.2
```
**ip policy-class** <policyname>

Use the `ip policy-class` command to create an access control policy and enter the access control policy command set. Use the `no` form of this command to delete an access policy and all the entries contained in it. Variations of this command include:

`ip policy-class <policyname> <action>`.

**Note**  
Configured access policies will only be active if the `ip firewall` command has been entered at the Global Configuration mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.

**Caution**  
Before applying an access control policy to an interface, verify your Telnet connection will not be affected by the policy. If a policy is applied to the interface you are connecting through and it does not allow Telnet traffic, your connection will be lost.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;policyname&gt;</code></td>
<td>Identifies the configured access policy by alphanumeric descriptor (maximum of 255 characters). All access policy descriptors are case-sensitive.</td>
</tr>
<tr>
<td><code>&lt;action&gt;</code></td>
<td>Specifies the action for the ACP as <code>allow</code>, <code>discard</code>, or <code>nat</code>).</td>
</tr>
</tbody>
</table>
| `allow list`         | All packets permitted by the access control list (ACL) will be allowed to enter the interface to which the policy class is assigned and an association will be created in the firewall. All associations created by the `allow list` are subject to the built-in `firewall timers` (refer to `ip policy-timeout <protocol> <range> <port> <seconds>` on page 407). All packets denied by the ACL will be processed by the next policy class entry or implicitly discarded if no further policy class entries exist. Possible `allow list` actions performed by the access policy are as follows:  
  - `allow list <access control list name>`  
  - `allow list <access control list name> stateless`  
  - `allow list <access control list name> policy <access policy name>`  
  - `allow list <access control list name> policy <access policy name> stateless`  
  - `allow list <access control list name> self`  
  - `allow list <access control list name> self stateless`  
| `policy`             | When the `policy <access policy name>` is specified, the firewall attempts to match the specified access policy with the access policy that is applied to the packet's egress interface as determined by the routing table or policy-based routing configuration. If there is a match, the firewall will process the packet. If there is no match, the firewall will process the packet based on the next policy class entry or implicitly discard it if no further policy class entries exist. |
allow list (continued)  

**self**

When the `self` keyword is applied, packets permitted by the ACL destined for any local interface on the unit will be allowed. These packets are terminated by the unit and are not routed or forwarded to other destinations. Using the `self` keyword is helpful when opening up remote administrative access to the unit (Telnet, SSH, ICMP, Web GUI).

**stateless**

When the `stateless` keyword is applied, traffic is not subject to the built-in firewall timers. Stateless traffic bypasses the application-level gateways (ALGs). Stateless processing is helpful when passing traffic over VPN tunnels. Traffic sent over VPN tunnels is purposely selected and encrypted; there is no need to firewall the traffic as well. VPN configurations created using the VPN Wizard in the Web GUI use `stateless` processing by default.

allow reverse list

The `allow reverse list` command is identical in function to the `allow list` command with the exception of the `reverse` keyword. The `reverse` keyword instructs the firewall to use the source information as the destination information and visa versa in the specified ACL.

discard list

All packets permitted by the ACL will be explicitly discarded upon entering the interface that the policy class is assigned to. All packets denied by the ACL will be processed by the next policy class entry or implicitly discarded if no further policy class entries exist. Possible `discard list` actions performed by the access policy are as follows:

```
discard list <access control list name>
discard list <access control list name> policy <access policy name>
discard list <access control list name> self
```

**policy** `<access policy name>`

When the `policy` `<access policy name>` is specified, the firewall attempts to match the specified access policy with the access policy that is applied to the packet's egress interface as determined by the routing table or policy-based routing configuration. If there is a match, the firewall will process the packet. If there is no match, the firewall will process the packet based on the next policy class entry or implicitly discard it if no further policy class entries exist.

**self**

When the `self` keyword is applied, packets permitted by the access-control list destined for any local interface on the unit will be implicitly discarded.

**nat source list**

All packets permitted by the ACL entering the interface to which the policy class is assigned will translate the source IP address of the packet to the specified `address` or `interface` and an association will be created in the firewall. This function is commonly referred to as a “many-to-one NAT”. All associations created by the `nat source list` are subject to the built-in firewall timers (refer to `ip policy-timeout <protocol> <range> <port> <seconds>` on page 407). All packets denied by the extended access control list will be processed by the next
nat source list

(continued)

policy class entry or implicitly discarded if no further policy class entries exist. Possible nat source list actions performed by the access policy are as follows:

```
nat source list <access control list name> address <IP address> overload
nat source list <access control list name> address <IP address> policy
   <access policy name>
nat source list <access control list name> interface <interface> overload
nat source list <access control list name> interface <interface> policy
   <access policy name>
```

address <IP address>
The address keyword specifies the IP address from which the translated packets will be sourced.

interface <interface>
The primary IP address of an interface is used as the source IP for translated packets when the interface keyword is applied.

policy <access policy name>
When the policy <access policy name> is specified, the firewall attempts to match the specified access policy with the access policy that is applied to the packet's egress interface as determined by the routing table or policy-based routing configuration. If there is a match, the firewall will process the packet. If there is no match, the firewall will process the packet based on the next policy class entry or implicitly discard it if no further policy class entries exist.

overload
The overload command is not optional and must be used when using the nat source list command.

nat destination list

All packets permitted by the specified extended ACL entering the interface that the policy class is assigned to will translate the destination IP address of the packet to the specified address and an association will be created in the firewall. All associations created by the nat destination list are subject to the built-in firewall timers (refer to ip policy-timeout <protocol> <range> <port> <seconds> on page 407). All packets denied by the extended ACL will be processed by the next policy class entry or implicitly discarded if no further policy class entries exist. Possible nat destination list actions performed by the access policy are as follows:

```
nat destination list <extended access control list name> address <IP address>
   port <port number>
```

address <IP address>
The address keyword specifies the private IP host to which the translated packets are destined.

port <port number>
The port keyword is used to translate the original destination port to a user-specified port.
Default Values

By default, all SROS security features are disabled and there are no configured access lists.

Functional Notes

SROS access control policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded.

Caution: An implicit discard exists at the end of every policy class. Specifying a discard list is unnecessary in most applications and should be used with caution. A discard list can adversely affect certain functions of a unit (VPN, routing protocols, etc.). Specifying an empty ACL or a non-existent ACL in a policy class will result in an implicit permit.

Usage Examples

The following is an example of adding policy class entries (ACL self and ACL MATCHALL) to a policy class named Private:

ProCurve(config)#ip policy-class Private
ProCurve(config-policy-class)#allow list self self
ProCurve(config-policy-class)#nat destination list MATCHALL interface ppp 1 overload

The following is a sample output of the configuration after issuing these commands:

! ip access-list standard wizard-ics
   remark Internet Connection Sharing
   permit any
   !
   ip access-list extended self
   remark Traffic to Router
   permit ip any any log
   !
   ip policy-class Private
   allow list self self
   nat source list wizard-ics interface ppp 1 overload

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:
Step 1:
Enable the security features of the SROS using the ip firewall command.
Step 2:
Create an access control list to permit or deny specified traffic. Standard ACLs match based on the source of the packet. Extended ACLs match based on the source and destination of the packet. Sources can be expressed in one of four ways:

1. Using the keyword **any** to match any IP address.
2. Using **host** `<A.B.C.D>` to specify a single host address.
3. Using the `<A.B.C.D> <wildcard>` format to match all IP addresses in a range. Wildcard masks work in reverse logic from subnet masks. Specifying 255 in any octet of the wildcard mask equates to a "don't care".
4. Using the keyword **hostname** to match based on a DNS name. The unit must be configured with DNS servers for this function to work.

Step 3:
Create an access policy that uses a configured access list. SROS access policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (**access list**) and an action (**allow**, **discard**, **NAT**). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

```
allow list `<access control list name>`
allow list `<access control list name>` stateless
allow list `<access control list name>` policy `<access policy name>`
allow list `<access control list name>` policy `<access policy name>` stateless
allow list `<access control list name>` self
allow list `<access control list name>` self stateless
discard list `<access control list name>`
discard list `<access control list name>` policy `<access policy name>`
discard list `<access control list name>` self
nat destination list `<access control list name>` address `<IP address>` port `<port number>`
nat source list `<access control list name>` address `<IP address>` overload
nat source list `<access control list name>` address `<IP address>` policy `<access policy name>`
nat source list `<access control list name>` interface `<interface>` overload
nat source list `<access control list name>` interface `<interface>` policy `<access policy name>`
```

Step 4:
Apply the created access policy to an interface. To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** `<policy name>`. The following example assigns access policy **MatchAll** to the Ethernet 0/1 interface:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#access-policy MatchAll
```
Use the `ip policy-class max-sessions` and `ip policy-class max-host-sessions` commands to create or alter settings for an access control policy. For more details on IP policy class functionality in the SROS, refer to `ip policy-class <policyname>` on page 400. Use the `no` form of this command to delete an access policy and all the entries contained in it. Variations of this command include the following:

```plaintext
ip policy-class max-sessions <number>
ip policy-class <policyname> max-host-sessions <number>
ip policy-class <policyname> max-sessions <number>
```

### Syntax Description

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;policyname&gt;</code> - Identifies the configured access policy using an alphanumeric descriptor (maximum of 255 characters). All access policy descriptors are case-sensitive.</td>
<td>PRO-CURVE(config)#ip policy-class &lt;policyname&gt; max-host-sessions 100</td>
</tr>
<tr>
<td><code>max-sessions &lt;number&gt;</code> - Specifies the maximum number of allowed policy sessions. Identifying a policy name sets the session limit only for the named policy. Using this command without specifying a policy name sets the limit for the total number of allowed sessions for all policies on the device. This number must be within the appropriate range limits. The limits are 1 to 30,000. Setting this value to zero turns the feature off.</td>
<td>PRO-CURVE(config)#ip policy-class max-sessions 55700</td>
</tr>
<tr>
<td><code>max-host-sessions &lt;number&gt;</code> - Specifies the maximum number of allowed policy sessions which can be created from each unique source address. This command is used in conjunction with a named policy and only applies the limit to that particular policy. The number must be within the appropriate range limits. The limits are 1 to 30,000. Setting this value to 0 turns the feature off.</td>
<td>PRO-CURVE(config)#ip policy-class Private max-host-sessions 100</td>
</tr>
</tbody>
</table>

### Default Values

By default, all SROS security features are disabled and there are no configured access lists.

### Usage Examples

The following example allows no more than 100 policy sessions to be sourced from a single host IP address on the `Private` policy class:

```plaintext
PRO-CURVE(config)#ip policy-class Private max-host-sessions 100
```

The following example sets a total global limit of 55,700 policy sessions allowed on all policy classes:

```plaintext
PRO-CURVE(config)#ip policy-class max-sessions 55700
```

The following example allows no more than 100 policy sessions on the `Private` policy class:

```plaintext
PRO-CURVE(config)#ip policy-class Private max-sessions 100
```

The following example removes the policy sessions limit on the `Private` policy class:

```plaintext
PRO-CURVE(config)#no ip policy-class Private max-sessions 100
```
ip policy-class <policyname> rpf-check

Use the `ip policy-class rpf-check` command to verify that traffic has entered on the appropriate interface using a route lookup. Reverse Path Forwarding (RPF) is essentially a spoofing check. For more details on IP policy class functionality in SROS, refer to `ip policy-class <policyname>` on page 400. Use the `no` form of this command to disable this feature.

**Syntax Description**

<table>
<thead>
<tr>
<th><strong>&lt;policyname&gt;</strong></th>
<th>Identifies the configured access policy using an alphanumeric descriptor (maximum of 255 characters). All access policy descriptors are case-sensitive.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>rpf-check</strong></td>
<td>Enables RPF check (spoofing).</td>
</tr>
</tbody>
</table>

**Default Values**

This command is enabled by default.

**Functional Notes**

The `rpf-check` feature should be disabled if your application allows incoming traffic on policy classes that do not match the route table’s source destination specifications. This feature can be disabled on a per policy class basis by issuing the command in conjunction with the policy class name you do not want to be checked.

**Usage Examples**

The following example turns off the `rpf-check` feature for the `Private` policy class:

```
ProCurve(config)# no ip policy-class Private rpf-check
```
**ip policy-timeout** <protocol> <range> <port> <seconds>

Use multiple **ip policy-timeout** commands to customize timeout intervals for protocols (TCP UDP ICMP) or specific services (by listing the particular port number). Use the **no** form of this command to return to the default timeout values.

**Syntax Description**

| **<protocol>** | Specifies the data protocol such as ICMP, TCP, UDP, AHP, GRE, or ESP. |
| **<range>**   | Optional. Customizes timeout intervals for a range of TCP or UDP ports. |
| **<port>**    | Service port to apply the timeout value to; valid only for specifying TCP and UDP services (not allowed for ICMP). |

The following is the list of UDP port numbers that may be identified using the text name (in **bold**):

- all-ports
- biff (Port 512)
- bootpc (Port 68)
- bootps (Port 67)
- discard (Port 9)
- dnsix (Port 195)
- domain (Port 53)
- echo (Port 7)
- isakmp (Port 500)
- mobile-ip (Port 434)
- nameserver (Port 42)
- netbios-dgm (Port 138)
- netbios-ns (Port 137)
- netbios-ss (Port 139)

The following is the list of TCP port numbers that may be identified using the text name (in **bold**):

- all_ports
- bgp (Port 179)
- chargen (Port 19)
- cmd (Port 514)
- daytime (Port 13)
- discard (Port 9)
- domain (Port 53)
- echo (Port 7)
- exec (Port 512)
- ftp (Port 21)
- iso (Port 54)
- login (Port 513)
- lpd (Port 515)
- nntp (Port 119)
- nntp (Port 119)
- pim-auto-rp (Port 496)
- po (Port 516)
- pop2 (Port 109)
- pop3 (Port 110)
- smtp (Port 25)
- snmp (Port 161)
- snmptrap (Port 162)
- snmp (Port 161)
- snmptrap (Port 162)
- smtp (Port 25)
- smtp (Port 25)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)
- syslog (Port 514)

The following is the list of TCP port numbers that may be identified using the text name (in **bold**):

- all_ports
- bgp (Port 179)
- chargen (Port 19)
- cmd (Port 514)
- daytime (Port 13)
- discard (Port 9)
- domain (Port 53)
- echo (Port 7)
- exec (Port 512)
- ft
Syntax Description (Continued)

<table>
<thead>
<tr>
<th>&lt;port&gt; *Optional</th>
<th>finger (Port 79)</th>
<th>tacacs (Port 49)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftp (Port 21)</td>
<td>ftp-data (Port 20)</td>
<td>talk (Port 517)</td>
</tr>
<tr>
<td>gopher (Port 70)</td>
<td>hostname (Port 101)</td>
<td>telnet (Port 23)</td>
</tr>
<tr>
<td>ident (Port 113)</td>
<td>time (Port 37)</td>
<td>uucp (Port 540)</td>
</tr>
<tr>
<td>irc (Port 194)</td>
<td>whois (Port 43)</td>
<td>www (Port 80)</td>
</tr>
<tr>
<td>klogin (Port 543)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kshell (Port 544)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<seconds>

Wait interval (in seconds) before an active session is closed (valid range: 0 to 4294967295 seconds).

Default Values

| <seconds> | The following default policy timeout intervals apply: | tcp (600 seconds; 10 minutes) | udp (60 seconds; 1 minute) | icmp (60 seconds; 1 minute) |

Usage Examples

The following example creates customized policy timeouts for the following:

- Internet traffic (TCP Port 80) timeout 24 hours (86400 seconds)
- telnet (TCP Port 23) timeout 20 minutes (1200 seconds)
- FTP (21) timeout 5 minutes (300 seconds)
- All other TCP services timeout 8 minutes (480 seconds)

ProCurve(config)#ip policy-timeout tcp www 86400
ProCurve(config)#ip policy-timeout tcp telnet 1200
ProCurve(config)#ip policy-timeout tcp ftp 300
ProCurve(config)#ip policy-timeout tcp all_ports 480

The following example creates customized policy timeouts for UDP netbios ports 137-139 of 200 seconds and UDP ports 6000-7000 of 300 seconds:

ProCurve(config)#ip policy-timeout udp range netbios-ns netbios-ss 200
ProCurve(config)#ip policy-timeout udp 6000 7000 300
**ip prefix-list** `<listname>` **description** `<"text">`

Use the `ip prefix-list description` command to create and name prefix lists.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;listname&gt;</code></th>
<th>Specifies a particular prefix list.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>description <code>&lt;text&gt;</code></strong></td>
<td>Assigns text (set apart by quotation marks) used as a description for the prefix list. Maximum length is 80 characters.</td>
</tr>
</tbody>
</table>

**Default Values**

No default values are necessary for this command.

**Functional Notes**

This command adds a string of up to 80 characters as a description for a prefix list. It also creates the prefix list if a prefix list of that name does not already exist.

**Usage Examples**

The following example adds a description to the prefix-list `test`:

```
ProCurve(config)#ip prefix-list test description “An example prefix list”
```
ip prefix-list <listname> seq <sequence#> [permit | deny] <network/len> [le <le-value> | ge <ge-value>]

Use the ip prefix-list seq command to specify a prefix to be matched or a range of mask lengths.

Syntax Description

- `<listname>`: Specifies a particular prefix list.
- `<sequence#>`: Specifies the entry's unique sequence number which determines the processing order. Lower-numbered entries are processed first. Range: 1 to 4,294,967,294.
- `permit`: Permits access to matching entries.
- `deny`: Denies access to matching entries.
- `<network/len>`: Specifies the network number and network mask length.
- `le <le-value>`: Specifies the upper end of the range. Range: 0 to 32.
- `ge <ge-value>`: Specifies the lower end of the range. Range: 0 to 32.

Default Values

If no ge or le parameters are specified, an exact match is assumed. If only ge is specified, the range is assumed to be from ge-value to 32. If only le is specified, the range is assumed to be from len to le-value.

Functional Notes

This command specifies a prefix to be matched. Optionally, it may specify a range of mask lengths. The following rule must be followed: len < ge-value < le-value. A prefix list with no entries allows all routes. A route that does not match any entries in a prefix list is dropped. As soon as a route is permitted or denied, there is no further processing of the rule in the prefix list. A route that is denied at the beginning entry of a prefix list will not be allowed, even if it matches a permitting entry further down the list.

Usage Examples

The following example creates a prefix list entry in the prefix list `test` matching only the 10.0.0.0/8 network:

```
ProCurve(config)#ip prefix-list test seq 5 deny 10.0.0.0/8
```

The following example creates a prefix list entry in the prefix list `test` matching any network of length 24 or less:

```
ProCurve(config)#ip prefix-list test seq 10 permit 0.0.0.0/0 le 24
```
ip radius source-interface <interface>

Use the ip radius source-interface command to specify the NAS (network-attached storage) IP address attribute passed with the RADIUS authentication request packet.

Syntax Description

| <interface> | Specifies the source interface (in the format type slot/port). Type ip radius source-interface ? for a complete list of interfaces. |

Default Values

By default, no source interface is defined.

Functional Notes

If this value is not defined, the address of the source network interface is used.

Usage Examples

The following example configures the Ethernet 0/1 port to be the source interface:

ProCurve(config)#ip radius source-interface ethernet 0/1
**ip route** `<ip address> <subnet mask> [<interface or ip address> | null 0] <administrative distance>`

Use the `ip route` command to add a static route to the route table. This command can be used to add a default route by entering `ip route 0.0.0.0 0.0.0.0` and specifying the interface or IP address. Use the `no` form of this command to remove a configured static route.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Specifies the network address (in dotted decimal notation) to add to the route table.</td>
</tr>
<tr>
<td><code>&lt;subnet mask&gt;</code></td>
<td>Specifies the subnet mask (in dotted decimal notation) associated with the listed network IP address.</td>
</tr>
<tr>
<td><code>&lt;interface or ip address&gt;</code></td>
<td>Specifies the gateway peer IP address (in dotted decimal notation) or a configured interface in the unit. Use the <code>?</code> command to display a complete list of interfaces.</td>
</tr>
<tr>
<td><code>null 0</code></td>
<td>Routes traffic destined for the specified network to the null interface. The router drops all packets destined for the null interface. Use the null interface to allow the router to advertise a route but not forward traffic to the route.</td>
</tr>
<tr>
<td><code>&lt;administrative distance&gt;</code></td>
<td>Specifies an administrative distance associated with this router (1 to 255). The administrative distance provides a way for a router to determine the best route when multiple routes to the same destination exist. The smaller the administrative distance, the more reliable the route.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no configured routes in route table.

**Usage Examples**

The following example adds a static route to the `10.220.0.0/16` network through the next-hop router `192.22.45.254` and a default route to `175.44.2.10`:

```
ProCurve(config)#ip route 10.220.0.0 255.255.0.0 192.22.45.254
ProCurve(config)#ip route 0.0.0.0 0.0.0.0 175.44.2.10
```
ip routing

Use the **ip routing** command to enable the SROS IP routing functionality. Use the **no** form of this command to disable IP routing.

**Syntax Description**

No subcommands.

**Default Values**

By default, IP routing is enabled.

**Usage Examples**

The following example enables the SROS IP routing functionality:

```
ProCurve(config)#ip routing
```
**ip rtp firewall-traversal [policy-timeout <seconds>]**

Use the ip rtp firewall-traversal command to enable dynamic firewall traversal capability for RTP-based traffic, allowing deep packet inspection of SDP packets to occur so RTP will correctly traverse NAT in the firewall. This will open the proper ports dynamically for the RTP traffic.

**Syntax Description**

| **policy-timeout <seconds>** | Optional. Specifies timeout period allowed for inactive RTP sessions to remain in the firewall. Range is 1 to 4,294,967,295. |

**Default Values**

By default, the RTP dynamic firewall traversal is disabled and the default policy timeout period is 45 seconds.

**Functional Notes**

SIP uses the Session Description Protocol (SDP) to format the SIP message body in order to negotiate a Real-time Transport Protocol (RTP)/Real-time Transport Control Protocol (RTCP) connection between two or more User Agents (UAs). The ports used for this will always be selected in a pair, with the even port used for RTP and the odd port for RTCP.

The SIP ALG (enabled using the ip firewall alg sip) configures the firewall to examine the ALL SIP packets it identifies and maintain knowledge of SIP transmissions on the network. Since SIP packet headers include port information for the call setup, the ALG must intelligently read the packets and remember the information.

For a full SIP implementation, dynamic firewall traversal for RTP traffic must also be enabled using the ip rtp firewall-traversal command. This allows the firewall to open the proper ports for the RTP traffic between UAs. For more details on SIP functionality in the SROS, refer to the Functional Notes and Technology Review sections of the ip firewall alg [ftp | h323 | pptp | sip] on page 374.

**Usage Examples**

The following example enables dynamic firewall traversal and sets the policy timeout period at 60 seconds:

ProCurve(config)#ip rtp firewall-traversal policy-timeout 60
**ip scp server**

Use the `ip scp server` command to enable the secure copy server functionality within the SROS. Enabling the secure copy server allows the SROS to support the transfer of files using a secure connection. A secure connection helps provide protection against outside forces gaining access to configuration files. An external secure copy server (such as PuTTY) is required to facilitate the transfers from the terminal.

**Syntax Description**

No subcommands.

**Default Value**

By default, the secure copy server function is disabled.

**Usage Examples**

The following example enables the secure copy server function in the SROS:

```
ProCurve>enable
ProCurve#config terminal
ProCurve(config)#ip scp server
```
ip sip

Use the `ip sip` command to enable the SROS SIP stack. When the SIP stack is enabled, memory is allocated for SIP functionality. For more details on SIP operation, refer to the Technology Review section of the command `ip firewall alg [ftp | h323 | pptp | sip]` on page 374. Use the `no` form of this command to disable the SIP stack.

**Syntax Description**

No subcommands.

**Default Value**

By default, the SIP stack is disabled. Refer to the Functional Notes section for more details on the default state of SIP operation in the SROS.

**Functional Notes**

By default, the SROS SIP ALG is enabled. This ALG allows the firewall to examine the ALL SIP packets it identifies and maintain knowledge of SIP transmissions on the network based on the SIP header. The SIP ALG requires the use of the SIP stack and the SIP proxy server in order to properly route SIP calls and maintain the SIP information. When the SIP ALG is enabled, the SIP stack and SIP proxy server are automatically enabled. For proper SIP operation, the firewall must also be configured to allow for dynamic holes for the RTP/RTCP traffic associated with SIP calls between User Agents (UAs). This functionality must be manually enabled using the `ip rtp firewall-traversal` command.

To completely disable SIP operation in the SROS, the following commands should be entered: `no ip firewall alg sip`, `no ip sip`, `no ip sip proxy`, and `no ip rtp firewall-traversal`. The `no ip firewall alg sip` command disables the SIP ALG. The `no ip sip` command disables the SIP stack and frees all memory allocated to the stack. The `no ip sip proxy` command disables the SIP proxy server. This command is not necessary to disable SIP functionality (because the `no ip sip` effectively shuts the proxy server down by disabling the stack), but should be entered for a cleaner configuration.

**Usage Examples**

The following example enables the SIP stack in the SROS:

```
ProCurve>enable
ProCurve#config terminal
ProCurve(config)#ip sip
```
ip sip [database local | location] <username> <ip address>

Use the ip sip database local command to store the location database of SIP User Agents (UAs) across a power loss. Use the ip sip location command to manually add a SIP UA to the location database. For more details on SIP operation, refer to the Technology Review section of the command ip firewall alg [ftp | h323 | pptp | sip] on page 374. Use the no form of the ip sip location command to remove a specific database location entry.

Syntax Description

<table>
<thead>
<tr>
<th>database local</th>
<th>Stores the SIP location database using memory on the local router. This database is maintained across a power loss.</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Adds a SIP UA to the location database. Manually adding a UA to the database is generally not required unless your SIP network is running in non-registering mode.</td>
</tr>
<tr>
<td>&lt;username&gt;</td>
<td>Specifies the username for the UA being added to the location database.</td>
</tr>
<tr>
<td>&lt;ip address&gt;</td>
<td>Specifies the IP address for the UA being added to the location database.</td>
</tr>
</tbody>
</table>

Default Values

By default, this command is disabled.

Usage Examples

The following example specifies an IP SIP location of 192.33.5.99 for a user named 2001:

ProCurve(config)#ip sip location 2001 192.33.5.99
**ip sip proxy**

Use the **ip sip proxy** command to enable the SROS SIP proxy server functionality for SIP packets to or from User Agents (UAs) in the location database. The SIP proxy server is used to accept SIP packets on a network, read the packet, route the packet, and maintain a database of information pertaining to SIP UAs located on the network. To enable the SIP proxy server to intercept ALL SIP packets and route them (regardless of whether the UA is registered in the database location), use this command in conjunction with the **ip firewall alg sip** command. For more details on SIP operation, refer to the *Technology Review* section of the command **ip firewall alg [ftp | h323 | pptp | sip]** on page 374. Use the **no** form of this command to disable the SIP proxy server.

**Syntax Description**

No subcommands.

**Default Values**

By default, the SIP stack is disabled. Refer to the *Functional Notes* section for more details on the default state of SIP operation in the SROS.

**Functional Notes**

By default, the SROS SIP ALG is enabled. This ALG allows the firewall to examine the ALL SIP packets it identifies and maintain knowledge of SIP transmissions on the network based on the SIP header. The SIP ALG requires the use of the SIP stack and the SIP proxy server in order to properly route SIP calls and maintain the SIP information. When the SIP ALG is enabled, the SIP stack and SIP proxy server are automatically enabled. For proper SIP operation, the firewall must also be configured to allow for dynamic holes for the RTP/RTCP traffic associated with SIP calls between User Agents (UAs). This functionality must be manually enabled using the **ip rtp firewall-traversal** command.

To completely disable SIP operation in the SROS, the following commands should be entered: **no ip firewall alg sip**, **no ip sip**, **no ip sip proxy**, and **no ip rtp firewall-traversal**. The **no ip firewall alg sip** command disables the SIP ALG. The **no ip sip** command disables the SIP stack and frees all memory allocated to the stack. The **no ip sip proxy** command disables the SIP proxy server. This command is not necessary to disable SIP functionality (because the **no ip sip** effectively shuts the proxy server down by disabling the stack), but should be entered for a cleaner configuration.

**Usage Examples**

The following example enables the proxy server:

```
ProCurve(config)#ip sip proxy
```
ip sip registrar [authenticate | default-expires | max-expires | min-expires | realm] <timevalue>

Use the ip sip registrar command to enable and configure the SIP registrar server used for registering User Agents (UAs) into the location database. For more details on SIP operation, refer to the Technology Review section of the command ip firewall alg [ftp | h323 | pptp | sip] on page 374. Use the no form of the ip sip registrar command to disable the registrar server.

Syntax Description

- **authenticate**: Specifies that authentication is required for each UA during registration.
- **default-expires**: Specifies the default expiration period for the UA listing in the location database. UAs requesting registration without specifying an expiration period are given the default expiration period.
- **max-expires**: Specifies the maximum expiration period for the UA listing in the location database. All UAs registering with the SIP proxy server request an expiration period for the listing in the database. UAs requesting an expiration period between the max-expires and min-expires values are honored. Enter a time in seconds from 0 to 2,592,000.
- **min-expires**: Specifies the minimum expiration period for the UA listing in the location database. All UAs registering with the SIP proxy server request an expiration period for the listing in the database. UAs requesting an expiration period between the max-expires and min-expires values are honored. Enter a time in seconds from 0 to 2,592,000.
- **realm**: Specifies a realm (enter a ascii character string) for the UA listing in the location database.
- **<timevalue>**: Specifies time in seconds.

Default Values

By default, the SIP registrar server is disabled.

Usage Examples

The following example sets the default expiration to 5 seconds:
ProCurve(config)#ip sip registrar default-expires 5

The following example sets the realm string:
ProCurve(config)#ip sip registrar realm voice.procurve.com
**ip snmp agent**

Use the `ip snmp agent` command to enable the Simple Network Management Protocol (SNMP) agent.

**Syntax Description**

No subcommands.

**Default Values**

By default, the SNMP agent is disabled.

**Functional Notes**

Allows a MIB browser to access standard MIBs within the product. This also allows the product to send traps to a trap management station.

**Usage Examples**

The following example enables the IP SNMP agent:

```
ProCurve(config)#ip snmp agent
```
ip sntp source-interface <interface>

The `ip sntp source-interface` command to use the specified interface’s IP address as the source IP address for SNTP traffic transmitted by the unit. Use the `no` form of this command if you do not wish to override the normal source IP address.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;interface&gt;</th>
<th>Specifies the interface to be used as the source IP address for SNTP traffic. Type <code>ip sntp source-interface?</code> for a complete list of valid interfaces.</th>
</tr>
</thead>
</table>

**Default Values**

No default value is necessary for this command.

**Functional Notes**

This command allows you to override the `sender` field in the IP packet. If you have multiple interfaces in your unit, changing the `sender` tells the receiver where to send replies. This functionality can also be used to allow packets to get through firewalls that would normally block the flow.

**Usage Examples**

The following example configures the unit to use the `loopback 1` interface as the source IP for SNTP traffic:

ProCurve(config)#ip sntp source-interface loopback
ip [ssh-server <port> | telnet-server <port>]

Use the this command to specify alternate transmission control protocol (TCP) ports for secure shell (SSH) and Telnet servers. Use the no form of this command to return to default settings.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssh server &lt;port&gt;</td>
<td>Configures the SSH server to listen on an alternate TCP port.</td>
</tr>
<tr>
<td>telnet server &lt;port&gt;</td>
<td>Configures the Telnet server to listen on an alternate TCP port.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the SSH server listens on TCP port 22 and Telnet listens on TCP port 23.

**Functional Notes**

SSH is a newer version of Telnet which allows you to run command line and graphical applications (as well as transfer files) over an encrypted connection.

**Usage Examples**

The following example configures the Telnet server to listen on TCP port 2323 instead of the default port 23:

ProCurve(config)#ip telnet-server 2323

The following example configures the SSH server to listen on TCP port 2200 instead of the default port 22:

ProCurve(config)#ip ssh-server 2200

To return to the default settings, use the no version of the command. For example:

ProCurve(config)#no ip ssh-server 2200
ip subnet-zero

The `ip subnet-zero` command is the default operation and cannot be disabled. This command signifies the router’s ability to route to subnet-zero subnets.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is enabled.

**Usage Examples**

The following example `subnet-zero` is enabled:

```
ProCurve(config)#ip subnet-zero
```
**ip tacacs source-interface** `<interface>`

Use the **ip tacacs source-interface** command to specify a source IP address for TACACS+ traffic originated by the unit. Use the **no** form of this command to disable this feature.

**Syntax Description**

| `<interface>` | Specifies the interface to be used as the source IP address for TACACS+ traffic. Type **ip tacacs source-interface** ? for a complete list of valid interfaces. |

**Default Values**

No default value is necessary for this command.

**Functional Notes**

This command allows you to override the *sender* field in the IP packet. If you have multiple interfaces in your unit, changing the *sender* tells the receiver where to send replies. This functionality can also be used to allow packets to get through firewalls that would normally block the flow.

**Usage Examples**

The following example configures the unit to use the *loopback 1* interface as the source IP for TACACS+ traffic:

```
ProCurve(config)#ip tacacs source-interface loopback 1
```
ip tftp server [access-class <access-class> in]

Use the ip tftp server to enable the TFTP server. Use the no form of this command to disable the TFTP server.

Syntax Description

access-class <access-class> in  Specifies that all self-bound TFTP incoming connections are activated. Enter the name of the access class.

Default Values

By default, this command is disabled.

Usage Examples

The following example enables the TFTP server:

ProCurve(config)#ip tftp server

The following example enables the TFTP server as well as self-bound TFTP incoming connections. In this example, classname is the name of the access-class.

ProCurve(config)#ip tftp server access-class classname in
**ip tftp source-interface <interface>**

Use the `ip tftp source-interface` command to use the specified interface’s IP address as the source IP address for TFTP traffic transmitted by the unit. Use the `no` form of this command if you do not wish to override the normal source IP address.

**Syntax Description**

| `<interface>` | Specifies the interface to be used as the source IP address for TFTP traffic. |

**Default Values**

No default value is necessary for this command.

**Functional Notes**

This command allows you to override the `sender` field in the IP packet. If you have multiple interfaces in your unit, changing the `sender` tells the receiver where to send replies. This functionality can also be used to allow packets to get through firewalls that would normally block the flow.

**Usage Examples**

The following example configures the unit to use the `loopback 1` interface as the source IP for TFTP traffic:

```
ProCurve(config)#ip tftp source-interface loopback 1
```
isdn-group

Use the `isdn-group` command to enter the ISDN Group Configuration mode command set. Refer to the section *ISDN Group Config Command Set* on page 1112 for more information on the commands available for each group.

**Syntax Description**

No subcommands.

**Default Values**

No default values necessary for this command.

**Functional Notes**

An ISDN group allows the user to specify the maximum and minimum number of B-channels that can be used for a specific type of call. It is a logical group of B-channels from one or more ISDN interfaces. An ISDN interface can be a member of multiple ISDN groups which makes it possible to share its B-channels between different types of calls.

**Usage Examples**

The following example uses the `isdn-group` command to enter the ISDN Group Configuration mode:

```
ProCurve(config)#isdn-group
ProCurve(config-isdn-group 1)#
```
**isdn-number-template**

Use the `isdn-number-template` command to create an entry in the ISDN number type template that is used when encoding the Called Party and Calling Party information elements for inbound and outbound ISDN calls. Use the `no` form of this command to delete a configured entry.

Variations of this command include the following:

- `isdn-number-template <template id> prefix <prefix> abbreviated <pattern>`
- `isdn-number-template <template id> prefix <prefix> international <pattern>`
- `isdn-number-template <template id> prefix <prefix> national <pattern>`
- `isdn-number-template <template id> prefix <prefix> network-specific <pattern>`
- `isdn-number-template <template id> prefix <prefix> subscriber <pattern>`
- `isdn-number-template <template id> prefix <prefix> unknown <pattern>`

**Syntax Description**

- `<label>`
  - Specifies a numeric label identifier (valid range is 1 to 255) for the template entry.
- `prefix <number>`
  - Specifies the expected prefix for the call type. Prefixes can be left blank (using double quotation marks `"`) or consist of unlimited length strings of 0s and 1s. For example, for international calls made from within the United States, a prefix of `011` is expected.
- `abbreviated`
  - Specifies to use Abbreviated in the Type of Number octet (bits 110). Abbreviated is used mainly in private ISDN network applications and the implementation is network dependent.
- `international`
  - Specifies to use International in the Type of Number octet (bits 001). International is used for calls destined outside the national calling area. International calls have the international direct dialing prefix removed. For example, consider an international call of `011-N$, where the international direct dialing prefix is 011 and the N$ represents the digits necessary for routing the call at the destination. When the Called Party IE is created for this call, the prefix is stripped and the N$ digits are placed in the Number Digits field.
- `national`
  - Specifies to use National (bits 010) in the Type of Number octet. National is used for calls destined for inside the national calling area (i.e., does not cross into an international LATA). National calls have the direct dialing prefix removed. For example, consider a national call with a direct dialing prefix of 1 and NXX-NXX-XXXX to represent the ten-digit number necessary for routing the call. When the Called Party IE is created for this call, the prefix (1) is stripped and the NXX-NXX-XXXX digits are placed in the Number Digits field.
- `network-specific`
  - Specifies to use Network-Specific (bits 011) in the Type of Number octet. Network-Specific is used for calls that require special access to a private network which requires the use of a prefix that should be stripped once access to the network has been gained. Network-Specific calls have the dialing prefix removed. For example, a call to a private network with the 700 consists of 700-N$, where 700 is the dialing prefix and N$ represents the digits necessary for routing the call at the destination. When the Called Party IE is created for this call, the prefix is stripped and the N$ is placed in the Number Digits field.
- `subscriber`
  - Specifies to use Subscriber (bits 100) in the Type of Number octet. Subscriber is used for local calls (not long-distance). Subscriber calls, by default, have the area code removed. For example, a subscriber call to 916-555-1212 would have the...
prefix 916 stripped and 555-1212 in the Number Digits field. For areas with mandatory ten-digit dialing, a blank prefix should be entered to ensure that all ten digits are passed to the Number Digits field.

**unknown**

Specifies to use Unknown (bits 000) in the Type of Number octet. Unknown is used when the number type is not known. Unknown numbers are assumed to have no prefix, and the entire dialed number is presented in the Number Digits field.

**<pattern>**

Specifies a pattern for this template.

Valid Characters:

- 0-9 Match exact digit only.
- X Match any single digit 0-9.
- N Match any single digit 2-9.
- M Match any single digit 1-8.
- [ ] Match any digit in the list. For example: [1,4,6] matches 1, 4, and 6 only. [1-3, 5] matches 1, 2, 3, and 5.

**Default Values**

The following default number template entry exists for domestic emergency calls (911):

isdn-number-template 0 prefix " " subscriber 911

**Functional Notes**

The following is an example number type template:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Pattern</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot; &quot;</td>
<td>NXX-XXXX</td>
<td>Subscriber</td>
</tr>
<tr>
<td>&quot;4&quot;</td>
<td>NXX-NXX-XXXX</td>
<td>National</td>
</tr>
<tr>
<td>&quot;011&quot;</td>
<td>X$</td>
<td>International</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td>N11</td>
<td>Subscriber (i.e. 411, 911, etc.)</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example creates a number template entry (labeled 1) for calls from within one European country to Germany:

ProCurve(config)#isdn-number-template 1 prefix 49 international N$
**line [console | telnet | ssh] <line-number> <ending number>**

Use the `line` command to enter the line configuration for the specified console or telnet session. See the sections **Line (Console) Interface Config Command Set** on page 490, **Line (Telnet) Interface Config Command Set** on page 503, and **Line (SSH) Interface Config Command Set** on page 512 for information on the subcommands.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>console</strong></td>
<td>Specifies the DB-9 (female) CONSOLE port located on the rear panel of the unit. See the sections <strong>Line (Console) Interface Config Command Set</strong> on page 490 for information on the subcommands found in this.</td>
</tr>
<tr>
<td><strong>telnet</strong></td>
<td>Specifies a Telnet session(s) to configure for remote access. See the section <strong>Line (Telnet) Interface Config Command Set</strong> on page 503 for information on the subcommands found in this.</td>
</tr>
<tr>
<td><strong>ssh</strong></td>
<td>Enters the configuration mode for SSH. Refer to the section <strong>Line (SSH) Interface Config Command Set</strong> on page 512 for information on the subcommands found in that command set.</td>
</tr>
</tbody>
</table>

**<line-number>** Specifies the starting session to configure for remote access (valid range for console: 0; valid range for Telnet and SSH: 0 to 4).

If configuring a single Telnet or SSH session, enter the session number and leave the **<ending number>** field blank.

**<ending number>** Optional. Specifies the last Telnet or SSH session to configure for remote access (valid range: 0 to 4).

For example, to configure all available Telnet sessions, enter `line telnet 0 4`.

**Default Values**

By default, the SROS line console parameters are configured as follows:

- Data Rate: 9600
- Data bits: 8
- Stop bits: 1
- Parity Bits: 0
- No flow control

By default, there are no configured Telnet sessions.

**Usage Examples**

The following example begins the configuration for the CONSOLE port located on the rear of the unit:

```
ProCurve(config)#line console 0
ProCurve(config-con0)#
```
The following example begins the configuration for all available Telnet sessions:

ProCurve(config)#line telnet 0 4
ProCurve(config-telnet0-4)#

The following example begins the configuration for all available SSH sessions:

ProCurve(config)#line ssh 0 4
ProCurve(config-ssh0-4)#
lldp [minimum-transmit-interval l reinitialization-delay l transmit-interval l ttl-multiplier] <numeric value>

Use the lldp command to configure global settings that control the way LLDP functions.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum-transmit-interval</td>
<td>Defines the minimum amount of time between transmission of LLDP frames (in seconds).</td>
</tr>
<tr>
<td>reinitialization-delay</td>
<td>Minimum amount of time to delay after LLDP is disabled on a port before allowing transmission of additional LLDP frames on that port (in seconds).</td>
</tr>
<tr>
<td>transmit-interval</td>
<td>Defines the delay between LLDP frame transmission attempts during normal operation (in seconds).</td>
</tr>
<tr>
<td>ttl-multiplier</td>
<td>Defines the multiplier to be applied to the transmit interval to compute the time-to-live for data sent in an LLDP frame.</td>
</tr>
<tr>
<td>&lt;numeric value&gt;</td>
<td>Specifies the interval, delay, or multiplier.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, minimum-transmit-interval = 2 seconds (valid range: 1 through 8192); reinitialization-delay = 2 seconds (valid range 1 through 10); transmit-interval = 30 seconds (valid range 5 through 32,768); and ttl-multiplier = 4 (valid range 2 through 10).

**Functional Notes**

Once a device receives data from a neighboring device in an LLDP frame, it will retain that data for a limited amount of time. This amount of time is called time-to-live, and it is part of the data in the LLDP frame. The time-to-live transmitted in the LLDP frame is equal to the transmit-interval multiplied by the ttl-multiplier.

**Usage Examples**

The following example sets the LLDP minimum-transmit-interval to 10 seconds:

ProCurve(config)#lldp minimum-transmit-interval 10

The following example sets the LLDP reinitialization-delay to 5 seconds:

ProCurve(config)#lldp reinitialization-delay 5

The following example sets the LLDP transmit-interval to 15 seconds:

ProCurve(config)#lldp transmit-interval 15

The following example sets the LLDP ttl-multiplier to 2 and the time-to-live for all LLDP frames transmitted from this unit to 30 seconds:

ProCurve(config)#lldp transmit-interval 15

ProCurve(config)#lldp ttl-multiplier 2
logging console

Use the logging console command to enable the SROS to log events to all consoles. Use the no form of this command to disable console logging.

Syntax Description

No subcommands.

Default Values

By default, logging console is disabled.

Usage Examples

The following example enables the SROS to log events to all consoles:

ProCurve(config)#logging console
logging email address-list <email address> ; <email address>

Use the logging email command to specify one or more email addresses that will receive notification when an event matching the criteria configured using the logging email priority-level command is logged by the SROS. See logging email priority-level [error | fatal | info | notice | warning] on page 437 for more information. Use the no form of this command to remove a listed address.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;email address&gt;</th>
<th>Specifies the complete email address to use when sending logged messages (This field allows up to 256 characters.)</th>
</tr>
</thead>
</table>

Enter as many email addresses as desired, placing a semi-colon (;) between addresses.

Default Values

By default, there are no configured logging email addresses.

Usage Examples

The following example specifies three email addresses to use when sending logged messages:

ProCurve(config)#logging email address-list
admin@email.com;ntwk@email.com;support@email.com
logging email exception-report address-list <email address>; <email address>

Use the **logging email exception-report address-list** command to specify one or more email addresses to receive an exception report for use in troubleshooting. Use the **no** form of this command to remove a listed address.

**Syntax Description**

| **<email address>** | Specifies the complete email address to use when sending exception reports. (This field allows up to 256 characters.) Enter as many email addresses as desired, placing a semi-colon (;) between addresses. |

**Default Values**

By default, there are no configured logging email addresses.

**Functional Notes**

When the SROS experiences an exception it will generate a file with detailed information that Technical Support can use to diagnose the problem. This command allows the unit to email the exception report to a list of addresses upon rebooting after the exception. This command should be used in conjunction with the other logging email commands. Refer to **logging email address-list <email address> ; <email address>** on page 434, **logging email on** on page 436, **logging email priority-level [error | fatal | info | notice | warning]** on page 437, **logging email receiver-ip <ip address>** on page 438, **logging email sender** on page 439, and **logging email source-interface <interface>** on page 440 for more information.

**Usage Examples**

The following example will enable exception report forwarding to john.doe@company.com using the 1.1.1.1 SMTP email server:

ProCurve(config)#logging email on
ProCurve(config)#logging email receiver-ip 1.1.1.1
ProCurve(config)#logging email exception-report address-list john.doe@company.com
logging email on

Use the **logging email on** command to enable the SROS email event notification feature. Use the **logging email address-list** command to specify email address(es) that will receive notification when an event matching the criteria configured using the **logging email priority-level** command is logged by the SROS. See **logging email priority-level [error | fatal | info | notice | warning]** on page 437 and **logging email priority-level [error | fatal | info | notice | warning]** on page 437 for more information. Use the **no** form of this command to disable the email notification feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, email event notification is disabled.

**Functional Notes**

The domain name is appended to the sender name when sending event notifications. See the command **ip domain-name <name>** on page 367 for related information.

**Usage Examples**

The following example enables the SROS email event notification feature:

```
ProCurve(config)# logging email on
```
logging email priority-level [error | fatal | info | notice | warning]

Use the `logging email priority-level` command to set the threshold for events sent to the addresses specified using the `logging email address-list` command. All events with the specified priority or higher will be sent to all addresses in the list. The logging email on command must be enabled. See `logging email priority-level [error | fatal | info | notice | warning]` on page 437 and `logging email on` on page 436 for related information. Use the `no` form of this command to return to the default priority.

**Syntax Description**

Sets the minimum priority threshold for sending messages to email addresses specified using the `logging email address-list` command.

The following priorities are available (ranking from lowest to highest):

**Error**
When selected, events with `error` and `fatal` priorities are logged.

**Fatal**
When selected, only events with a `fatal` priority are logged.

**Info**
When selected, all events are logged.

**Notice**
When selected, events with `notice`, `warning`, `error`, and `fatal` priorities are logged.

**Warning**
When selected, events with `warning`, `error`, and `fatal` priorities are logged.

**Default Values**

`<priority>`  `warning`

**Usage Examples**

The following example sends all messages with `warning` level or greater to the email addresses listed using the `logging email address-list` command:

```
ProCurve(config)#logging email priority-level warning
```
logging email receiver-ip <ip address>

Use the logging email receiver-ip command to specify the IP address of the email server to use when sending notification that an event matched the criteria configured using the logging email priority-level command. See logging email priority-level [error | fatal | info | notice | warning] on page 437 for related information. Use the no form of this command to remove a configured address.

Syntax Description

| <ip address> | Specifies the IP address (in dotted decimal notation) of the mail server to use when sending logged messages. |

Default Values

By default, there are no configured email server addresses.

Usage Examples

The following example specifies an email server (with address 172.5.67.99) to use when sending logged messages:

ProCurve(config)#logging email receiver-ip 172.5.67.99
logging email sender

Use the `logging email sender` command to specify the sender in an outgoing email message. This name will appear in the `From` field of the receiver’s inbox. Use the `no` form of this command to disable this feature.

**Syntax Description**

No subcommands.

**Default Values**

No default value is necessary for this command.

**Usage Examples**

The following example sets a sender for outgoing messages:

```
ProCurve(config)#logging email sender myUnit@myNetwork.com
```
logging email source-interface <interface>

Use the logging email source-interface command to use the specified interface’s IP address as the source IP address for email messages transmitted by the unit. Use the no form of this command if you do not wish to override the normal source IP address.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt;</td>
<td>Specifies the interface to be used as the source IP address for email messages. Type logging email source-interface ? for a list of valid interface types.</td>
</tr>
</tbody>
</table>

Default Values

No default value is necessary for this command.

Functional Notes

This command allows you to override the sender field in the IP packet. If you have multiple interfaces in your unit, changing the sender tells the receiver where to send replies. This functionality can also be used to allow packets to get through firewalls that would normally block the flow.

Usage Examples

The following example configures the unit to use the loopback 1 interface as the source IP for email messages:

ProCurve(config)#logging email source-interface loopback 1
logging facility <facility type>

Use the logging facility command to specify a syslog facility type for the syslog server. Error messages meeting specified criteria are sent to the syslog server. For this service to be active, you must enable log forwarding. See logging forwarding on page 442 for related information. Facility types are described under Functional Notes below. Use the no form of this command to return it to its default setting.

Syntax Description

<facility type> Specifies the syslog facility type (see Functional Notes below).

Default Values

The default value is local7.

Functional Notes

The following is a list of all the valid facility types:

auth Authorization system
cron Cron facility
daemon System daemon
kern Kernel
local0 - local7 Reserved for locally defined messages
lpr Line printer system
mail Mail system
news USENET news
sys9 - sys14 System use
syslog System log
user User process
uucp UNIX-to-UNIX copy system

Usage Examples

The following example configures the syslog facility to the cron facility type:

ProCurve(config)#logging facility cron
logging forwarding on

Use the `logging forwarding on` command to enable the SROS syslog event feature. Use the `logging forwarding priority-level` command to specify the event matching the criteria used by the SROS to determine whether a message should be forwarded to the syslog server. See `logging forwarding priority-level [error | fatal | info | notice | warning]` on page 443 for related information. Use the `no` form of this command to disable the syslog event feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, syslog event notification is disabled.

**Usage Examples**

The following example enables the SROS syslog event feature:

```
ProCurve(config)#logging forwarding on
```
logging forwarding priority-level [error | fatal | info | notice | warning]

Use the `logging forwarding priority-level` command to set the threshold for events sent to the configured syslog server specified using the `logging forwarding receiver-ip` command. All events with the specified priority or higher will be sent to all configured syslog servers. See `logging email priority-level [error | fatal | info | notice | warning]` on page 437 for more information. Use the `no` form of this command to return to the default priority.

**Syntax Description**

Sets the minimum priority threshold for sending messages to the syslog server specified using the `logging forwarding receiver-ip` command.

The following priorities are available (ranking from lowest to highest):

- **Error** When selected, events with error and fatal priorities are logged.
- **Fatal** When selected, only events with a fatal priority are logged.
- **Info** When selected, all events are logged.
- **Notice** When selected, events with notice, warning, error, and fatal priorities are logged.
- **Warning** When selected, events with warning, error, and fatal priorities are logged.

**Default Values**

By default, the `logging forwarding priority-level` is set to `warning`.

**Usage Examples**

The following example sends all messages with `warning` level or greater to the syslog server listed using the `logging forwarding receiver-ip` command.

```
ProCurve(config)#logging forwarding priority-level warning
```
logging forwarding receiver-ip <ip address>

Use this logging forwarding receiver-ip command to specify the IP address of the syslog server to use when logging events that match the criteria configured using the logging forwarding priority-level command. Enter multiple logging forwarding receiver-ip commands to develop a list of syslog servers to use. See logging forwarding priority-level [error | fatal | info | notice | warning] on page 443 for related information. Use the no form of this command to remove a configured address.

Syntax Description

| <ip address> | Specifies the IP address (in dotted decimal notation) of the syslog server to use when logging messages. |

Default Values

By default, there are no configured syslog server addresses.

Usage Examples

The following example specifies a syslog server (with address 172.5.67.99) to use when logging messages:

ProCurve(config)#logging forwarding receiver-ip 172.5.67.99
logging forwarding source-interface

Use the **logging forwarding source-interface** command to configure the specified interface’s IP address as the source IP address for the syslog server to use when logging events. Use the **no** form of this command if you do not wish to override the normal source IP address.

**Syntax Description**

```plaintext
<interface>  
Specifies the interface to be used as the source IP address for event log traffic.  
Type **logging forwarding source-interface ?** for a list of valid interface types.
```

**Default Values**

No default value is necessary for this command.

**Functional Notes**

This command allows you to override the *sender* field in the IP packet. If you have multiple interfaces in your unit, changing the *sender* tells the receiver where to send replies. This functionality can also be used to allow packets to get through firewalls that would normally block the flow.

**Usage Examples**

configures the unit to use the **loopback 1** interface as the source IP for event log traffic:

```
ProCurve(config)#logging forwarding source-interface loopback 1
```
mac address-table aging-time <aging time>

Use the `mac address-table aging-time` command to set the length of time dynamic MAC addresses remain in the switch or bridge forwarding table. Use the `no` form of this command to reset this length to its default.

**Syntax Description**

| `<aging time>` | Specifies an aging time (in seconds) from 10 to 1,000,000. Set to 0 to disable the timeout. |

**Default Values**

By default, the aging time is 300 seconds.

**Usage Examples**

The following example sets the aging time to 10 minutes:

```
ProCurve(config)#mac address-table aging-time 600
```
Use the `mac address-table static` command to insert a static MAC address entry into the MAC address table. Use the `no` form of this command to remove an entry from the table.

**Syntax Description**

- `<mac address>`: Specifies a valid 48-bit MAC address.
- `bridge <bridge id>`: Specifies a valid bridge interface ID.
- `interface <interface>`: Specifies a valid slot/port interface ID. Type `mac address-table static bridge interface ?` for a complete list of valid interfaces.

**Default Values**

By default, there are no static entries configured.

**Usage Examples**

The following example adds a static MAC address to Ethernet 0/1 on bridge group 4:

```
ProCurve(config)#mac address-table static 00:12:79:00:00:01 bridge 4 interface ethernet 0/1
```
**modem countrycode [</countrycode>]**

Use the **modem countrycode** command to specify the modem configuration for the applicable country.

**Syntax Description**

| <countrycode> | Specifies the modem configuration for the applicable country. |

**Default Values**

By default, **modem countrycode** is set to **USA/Canada**.

**Functional Notes**

The following country codes are available for modem configuration:

<table>
<thead>
<tr>
<th>Country</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>- Algeria Modem configuration</td>
</tr>
<tr>
<td>Argentina</td>
<td>- Argentina Modem configuration</td>
</tr>
<tr>
<td>Australia</td>
<td>- Australia Modem configuration</td>
</tr>
<tr>
<td>Austria</td>
<td>- Austria Modem configuration</td>
</tr>
<tr>
<td>Bahrain</td>
<td>- Bahrain Modem configuration</td>
</tr>
<tr>
<td>Belgium</td>
<td>- Belgium Modem configuration</td>
</tr>
<tr>
<td>Bolivia</td>
<td>- Bolivia Modem configuration</td>
</tr>
<tr>
<td>Brazil</td>
<td>- Brazil Modem configuration</td>
</tr>
<tr>
<td>Chile</td>
<td>- Chile Modem configuration</td>
</tr>
<tr>
<td>China</td>
<td>- China Modem configuration</td>
</tr>
<tr>
<td>Colombia</td>
<td>- Colombia Modem configuration</td>
</tr>
<tr>
<td>Costa_Rica</td>
<td>- Costa_Rica Modem configuration</td>
</tr>
<tr>
<td>Cyprus</td>
<td>- Cyprus Modem configuration</td>
</tr>
<tr>
<td>Czechoslovakia</td>
<td>- Czechoslovakia Modem configuration</td>
</tr>
<tr>
<td>Denmark</td>
<td>- Denmark Modem configuration</td>
</tr>
<tr>
<td>Ecuador</td>
<td>- Ecuador Modem configuration</td>
</tr>
<tr>
<td>Egypt</td>
<td>- Egypt Modem configuration</td>
</tr>
<tr>
<td>Finland</td>
<td>- Finland Modem configuration</td>
</tr>
<tr>
<td>France</td>
<td>- France Modem configuration</td>
</tr>
<tr>
<td>Germany</td>
<td>- Germany Modem configuration</td>
</tr>
<tr>
<td>Greece</td>
<td>- Greece Modem configuration</td>
</tr>
<tr>
<td>Guatemala</td>
<td>- Guatemala Modem configuration</td>
</tr>
<tr>
<td>Hong_Kong</td>
<td>- Hong_Kong Modem configuration</td>
</tr>
<tr>
<td>Hungary</td>
<td>- Hungary Modem configuration</td>
</tr>
<tr>
<td>India</td>
<td>- India Modem configuration</td>
</tr>
<tr>
<td>Indonesia</td>
<td>- Indonesia Modem configuration</td>
</tr>
<tr>
<td>Ireland</td>
<td>- Ireland Modem configuration</td>
</tr>
<tr>
<td>Israel</td>
<td>- Israel Modem configuration</td>
</tr>
<tr>
<td>Italy</td>
<td>- Italy Modem configuration</td>
</tr>
<tr>
<td>Country</td>
<td>Modem Configuration</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Japan</td>
<td>Japan Modem configuration</td>
</tr>
<tr>
<td>Jordan</td>
<td>Jordan Modem configuration</td>
</tr>
<tr>
<td>Korea</td>
<td>Korea Modem configuration</td>
</tr>
<tr>
<td>Kuwait</td>
<td>Kuwait Modem configuration</td>
</tr>
<tr>
<td>Lebanon</td>
<td>Lebanon Modem configuration</td>
</tr>
<tr>
<td>Malaysia</td>
<td>Malaysia Modem configuration</td>
</tr>
<tr>
<td>Mexico</td>
<td>Mexico Modem configuration</td>
</tr>
<tr>
<td>Morocco</td>
<td>Morocco Modem configuration</td>
</tr>
<tr>
<td>Netherlands</td>
<td>Netherlands Modem configuration</td>
</tr>
<tr>
<td>New Zealand</td>
<td>New_Zealand Modem configuration</td>
</tr>
<tr>
<td>Norway</td>
<td>Norway Modem configuration</td>
</tr>
<tr>
<td>Oman</td>
<td>Oman Modem configuration</td>
</tr>
<tr>
<td>Panama</td>
<td>Panama Modem configuration</td>
</tr>
<tr>
<td>Peru</td>
<td>Peru Modem configuration</td>
</tr>
<tr>
<td>Philippines</td>
<td>Philippines Modem configuration</td>
</tr>
<tr>
<td>Poland</td>
<td>Poland Modem configuration</td>
</tr>
<tr>
<td>Portugal</td>
<td>Portugal Modem configuration</td>
</tr>
<tr>
<td>Puerto_Rico</td>
<td>Puerto_Rico Modem configuration</td>
</tr>
<tr>
<td>Qatar</td>
<td>Qatar Modem configuration</td>
</tr>
<tr>
<td>Russia</td>
<td>Russia Modem configuration</td>
</tr>
<tr>
<td>Saudi_Arabia</td>
<td>Saudi_Arabia Modem configuration</td>
</tr>
<tr>
<td>Singapore</td>
<td>Singapore Modem configuration</td>
</tr>
<tr>
<td>Slovakia</td>
<td>Slovakia Modem configuration</td>
</tr>
<tr>
<td>Slovenia</td>
<td>Slovenia Modem configuration</td>
</tr>
<tr>
<td>South_Africa</td>
<td>South_Africa Modem configuration</td>
</tr>
<tr>
<td>Spain</td>
<td>Spain Modem configuration</td>
</tr>
<tr>
<td>Sweden</td>
<td>Sweden Modem configuration</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Switzerland Modem configuration</td>
</tr>
<tr>
<td>Syria</td>
<td>Syria Modem configuration</td>
</tr>
<tr>
<td>Taiwan</td>
<td>Taiwan Modem configuration</td>
</tr>
<tr>
<td>Thailand</td>
<td>Thailand Modem configuration</td>
</tr>
<tr>
<td>Trinidad</td>
<td>Trinidad Modem configuration</td>
</tr>
<tr>
<td>Tunisia</td>
<td>Tunisia Modem configuration</td>
</tr>
<tr>
<td>Turkey</td>
<td>Turkey Modem configuration</td>
</tr>
<tr>
<td>UAE</td>
<td>UAE Modem configuration</td>
</tr>
<tr>
<td>UK</td>
<td>UK Modem configuration</td>
</tr>
<tr>
<td>USA/Canada</td>
<td>USA/Canada Modem configuration</td>
</tr>
<tr>
<td>Uruguay</td>
<td>Uruguay Modem configuration</td>
</tr>
<tr>
<td>Venezuela</td>
<td>Venezuela Modem configuration</td>
</tr>
<tr>
<td>Yemen</td>
<td>Yemen Modem configuration</td>
</tr>
</tbody>
</table>
Usage Example

The following example specifies to use the **USA/Canada** modem configuration.

```
ProCurve(config)#modem countrycode USA/Canada
```
**qos map <mapname><sequence number>**

Use the `qos map` command to activate the QoS Map Command Set (which allows you to create and/or edit a QoS map). For details on specific commands, refer to the section *Quality of Service (QoS) Map Commands* on page 1265. Use the `no` form of this command to delete a map entry.

**Syntax Description**

- `<mapname>` Specifies the QoS map name.
- `<sequence number>` Specifies a number (valid range: 0 to 65,535) to differentiate this QoS map and to assign match order.

**Default Values**

No default value is necessary for this command.

**Functional Notes**

A QoS policy is defined using a QoS map. The QoS map is a named list with sequenced entries. An entry contains a single match reference and one or more actions (priority, set, or both). Multiple map entries for the same QoS map are differentiated by a sequence number. The sequence number is used to assign match order.

Once created, a QoS map must be applied to an interface (using the `qos-policy out <map-name>` command) in order to actively process traffic. Any traffic for the interface that is not sent to the priority queue is sent using the default queuing method for the interface (such as weighted fair queuing). `qos-policy out <mapname>` on page 811 for more information.

**Usage Examples**

The following example demonstrates basic settings for a QoS map and assigns a map to the frame-relay interface:

```
ProCurve>enable
ProCurve#config terminal
ProCurve(config)#qos map VOICEMAP 10
ProCurve(config-qos-map)#match precedence 5
ProCurve(config-qos-map)#priority 512
ProCurve(config-qos-map)#exit
ProCurve(config)#interface fr 1
ProCurve(config-fr 1)#qos-policy out VOICEMAP
```
radius-server

Use the `radius-server` command to configure several global RADIUS parameters. Most of these global defaults can be overridden on a per-server basis.

Variations of this command include the following:

- `radius-server challenge-noecho`
- `radius-server deadtime <minutes>`
- `radius-server enable-username <name>`
- `radius-server key <key>`
- `radius-server retry <attempts>`
- `radius-server timeout <seconds>`

Syntax Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>challenge-noecho</td>
<td>Turns off echoing of user challenge-entry. When echo is turned on, users see the text of the challenge as they type responses. Enabling this option hides the text as it is being entered.</td>
</tr>
<tr>
<td>deadtime &lt;minutes&gt;</td>
<td>Specifies how long a RADIUS server is considered dead once a timeout occurs. The server will not be tried again until after the deadtime expires.</td>
</tr>
<tr>
<td>enable-username &lt;name&gt;</td>
<td>Specifies a username to be used for enable authentication.</td>
</tr>
<tr>
<td>key &lt;key&gt;</td>
<td>Specifies the shared key to use with a RADIUS server.</td>
</tr>
<tr>
<td>retry &lt;attempts&gt;</td>
<td>Specifies how many attempts to make on a RADIUS server before marking it dead.</td>
</tr>
<tr>
<td>timeout &lt;seconds&gt;</td>
<td>Specifies how long to wait for a RADIUS server to respond to a request.</td>
</tr>
</tbody>
</table>

Default Values

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>challenge-noecho</td>
<td>By default, echo is turned on.</td>
</tr>
<tr>
<td>deadtime</td>
<td>1 minute</td>
</tr>
<tr>
<td>key</td>
<td>No default</td>
</tr>
<tr>
<td>retry</td>
<td>3 attempts</td>
</tr>
<tr>
<td>timeout</td>
<td>5 seconds</td>
</tr>
<tr>
<td>enable-username</td>
<td>$enab15$</td>
</tr>
</tbody>
</table>
Functional Notes

RADIUS servers (as defined with the `radius-server` command) may have many optional parameters. However, they are uniquely identified by their addresses and ports. Port values default to 1812 and 1813 for authorization and accounting, respectively. If a server is added to a named group but is not defined by a `radius-server` command, the server is simply ignored when accessed. Empty server lists are not allowed. When the last server is removed from a list, the list is automatically deleted.

Usage Examples

The following example shows a typical configuration of these parameters:

```
ProCurve(config)#radius-server challenge-noecho
ProCurve(config)#radius-server deadtime 10
ProCurve(config)#radius-server timeout 2
ProCurve(config)#radius-server retry 4
ProCurve(config)#radius-server key my secret key
```
**radius-server host**

Use the `radius-server host` to specify the parameters for a remote RADIUS server. At a minimum, the address (IP or DNS name) of the server must be given. The other parameters are also allowed and (if not specified) will take default values or fall back on the global RADIUS server’s default settings.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>acct-port</strong></td>
<td>Sends accounting requests to this remote port.</td>
</tr>
<tr>
<td><strong>auth-port</strong></td>
<td>Sends authentication requests to this remote port.</td>
</tr>
<tr>
<td><strong>retransmit</strong></td>
<td>Retries server after timeout this number of times (uses RADIUS global setting if not given).</td>
</tr>
<tr>
<td><strong>timeout</strong></td>
<td>Waits for a response this number of seconds (uses RADIUS global setting if not given).</td>
</tr>
<tr>
<td><strong>key</strong></td>
<td>Defines the shared key with the RADIUS server (uses RADIUS global setting if not given). Note that the key must appear last on the input line since it reads the rest of the line beyond the <code>key</code> keyword.</td>
</tr>
<tr>
<td><strong>key encrypted</strong></td>
<td>Defines an encrypted shared key with the RADIUS server (uses RADIUS global setting if not given). Note that the key must appear last on the input line since it reads the rest of the line beyond the <code>key</code> keyword.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>acct-port</td>
<td>1813</td>
</tr>
<tr>
<td>auth-port</td>
<td>1812</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example shows a typical configuration of these parameters:

```
ProCurve(config)#radius-server host 1.2.3.4
ProCurve(config)#radius-server host 3.3.1.2 acct-port 1646 key my key
```
route-map `<map-name>` [ permit | deny ] `<sequence number>`

Use the route-map command to create a route map and enter the Route Map Configuration command set. A route map is a type of filter that matches various attributes and then performs actions on the way the route is redistributed. Use the no form of this command to return to default settings.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;map-name&gt;</code></th>
<th>Specifies a name for the route map.</th>
</tr>
</thead>
<tbody>
<tr>
<td>permit</td>
<td>Redistributes routes matching the route map attributes.</td>
</tr>
<tr>
<td>deny</td>
<td>Specifies not to redistribute routes matching the route map attributes.</td>
</tr>
<tr>
<td><code>&lt;sequence number&gt;</code></td>
<td>Specifies a sequence number of this route entry. Range is 1 to 4,294,967,295.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no route maps are defined.

**Functional Notes**

Route maps can be assigned to a neighbor using the route-map command in the BGP Neighbor command set. See route-map `<map-name>` [in | out] on page 1228 for more information.

**Usage Examples**

The following example creates the route map, specifies that routes matching its criteria will be denied, and assigns a sequence number of 100:

ProCurve(config)#route-map MyMap deny 100
ProCurve(config-route-map)#

You can then define the attributes of the route map from the Route Map Configuration Command set. Enter a ? at the (config-route-map)# prompt to explore the available options.
**router bgp <AS number>**

Use the `router bgp` command to enter the BGP Configuration mode. Refer to the sections *BGP Configuration Command Set* on page 1205 and *BGP Neighbor Configuration Command Set* on page 1218 for more information.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;AS number&gt;</code></td>
<td>Specifies the AS number for this BGP interface.</td>
</tr>
</tbody>
</table>

**Default Values**

No default values necessary for this command.

**Usage Examples**

The following example uses the `router bgp` command to enter the BGP Configuration mode:

```
ProCurve(config)#router bgp 65000
ProCurve>(config-bgp)#
```

**Technology Review**

The following SROS BGP-related guidelines may help guide decisions made during basic BGP implementation.

- Ignore route if next hop is unreachable.
- Prefer route with largest weight (only used in the local router, set by applying route maps to set this value on desired inbound updates).
- Prefer route with largest local preference.
- Prefer route injected by this router via network command.
- Prefer route with shortest AS_PATH.
- Prefer route with lowest origin type. Routes originally injected by the network command or aggregation (IGP) have a lower origin type than those originally injected by redistribution into BGP.
- Prefer routes with lowest MED value.

Before the route is installed into the route table (forwarding table), a check is made of other sources that may have information about the same subnet (static routes, IGPs, etc.) The route with the lowest administrative distance is installed.
router ospf

Use the `router ospf` command to activate OSPF in the router and to enter the OSPF Configuration mode. See the section `Router (OSPF) Configuration Command Set` on page 1235 for more information. Use the `no` form of this command to disable OSPF routing.

**Syntax Description**

No subcommands.

**Default Values**

By default, OSPF is disabled.

**Functional Notes**

The SROS can be configured to use OSPF with the firewall enabled (using the `ip firewall` command). To do this, configure the OSPF networks as usual, specifying which networks the system will listen for and broadcast OSPF packets to. See `ip firewall` on page 369 for more information.

To apply stateful inspection to packets coming into the system, create a policy-class that describes the type of action desired and then associate that policy-class to the particular interface (see `ip policy-class <policyname>` on page 400). The firewall is intelligent and will only allow OSPF packets that were received on an OSPF configured interface. No modification to the policy-class is required to allow OSPF packets into the system.

**Usage Examples**

The following example uses the `router ospf` command to enter the OSPF Configuration mode:

```
ProCurve(config)#router ospf
ProCurve(config-ospf)#
```
router pim-sparse

Use the `router pim-sparse` command to globally enable protocol-independent multicast (PIM) on the unit and to enter the PIM Sparse Configuration mode. Refer to the section Router (PIM Sparse) Configuration Command Set on page 1249 for more information on the subcommands for PIM Sparse Configuration mode.

**Syntax Description**

No subcommands.

**Default Values**

No default values necessary for this command.

**Functional Notes**

Additional commands for PIM are found in the related interface configuration modes. See the `ip pim-sparse` commands in the various interface configuration sections for more information.

**Usage Examples**

The following example uses the `router pim-sparse` command to enter the PIM Sparse Configuration mode:

```
ProCurve(config)#router pim-sparse
ProCurve(config-pim-sparse)#
```
**router rip**

Use the `router rip` command to enter the RIP Configuration mode. See the section *Router (RIP) Configuration Command Set* on page 1253 for more information.

**Syntax Description**

No subcommands.

**Default Values**

No default values necessary for this command.

**Usage Examples**

The following example uses the `router rip` command to enter the RIP Configuration mode:

```
ProCurve(config)#router rip
ProCurve(config-rip)#
```

**Technology Review**

The RIP protocol is based on the Bellham-Ford (distance-vector) algorithm. This algorithm provides that a network will converge to the correct set of shortest routes in a finite amount of time, provided that:

- Gateways continuously update their estimates of routes.
- Updates are not overly delayed and are made on a regular basis.
- The radius of the network is not excessive.
- No further topology changes take place.

RIP is described in RFC 1058 (Version 1) and updated in RFCs 1721, 1722, and 1723 for Version 2. Version 2 includes components that ease compatibility in networks operating with RIP V1.

All advertisements occur on regular intervals (every 30 seconds). Normally, a route that is not updated for 180 seconds is considered dead. If no other update occurs in the next 60 seconds for a new and better route, the route is flushed after 240 seconds. Consider a connected route (one on a local interface). If the interface fails, an update is immediately triggered for that route only (advertised with a metric of 16).

Now consider a route that was learned and does not receive an update for 180 seconds. The route is marked for deletion, and even if it was learned on an interface, a poisoned (metric =16) route should be sent by itself immediately and during the next two update cycles with the remaining normal split horizon update routes. Following actual deletion, the poison reverse update ceases. If an update for a learned route is not received for 180 seconds, the route is marked for deletion. At that point, a 120-second garbage collection (GC) timer is started. During the GC timer, expiration updates are sent with the metric for the timed out route set to 16.
If an attached interface goes down, the associated route is immediately (within the same random five-second interval) triggered. The next regular update excludes the failed interface. This is the so-called first hand knowledge rule. If a gateway has first hand knowledge of a route failure (connected interfaces) or reestablishment, the same action is taken. A triggered update occurs, advertising the route as failed (metric = 16) or up (normal metric) followed by the normal scheduled update.

The assumption here is that if a gateway missed the triggered update, it will eventually learn from another gateway in the standard convergence process. This conserves bandwidth.

RIP-Related Definitions:
Route - A description of the path and its cost to a network.
Gateway - A device that implements all or part of RIP - a router.
Hop - Metric that provides the integer distance (number of intervening gateways) to a destination network gateway.
Advertisement - A broadcast or multicast packet to port 520 that indicates the route for a given destination network.
Update - An advertisement sent on a regular 30-second interval including all routes exclusive of those learned on an interface.
safe-mode <reload timer> <threshold time>

Use the safe-mode command to enable SafeMode operation. SafeMode operation allows the user to perform configuration changes without the possibility of being disconnected during a Telnet or SSH session. Use the no form of this command to disable SafeMode operation.

Syntax Description

<reload timer> Specifies the amount of time (in seconds) allowed for the user to write the configuration to memory. With SafeMode enabled, the router will begin the reboot process after the reload timer has expired. See the Functional Notes section for more details. Valid range is 30 to 3600 seconds.

<threshold time> Specifies the amount of time (in seconds) the router waits before reloading when the user is prompted to restart the SafeMode timer. To restart the SafeMode reload timer, press <Ctrl + r>. Valid range is 30 to 3600 seconds.

Default Values

reload timer The default for the reload timer is 300 seconds.

threshold time The default for the threshold time is 60 seconds.

Functional Notes

Users that manage a router remotely via SSH or Telnet can encounter a situation where the configuration change they perform interrupts network connectivity. For example, if the user is connected to the router via a Telnet session and configures an access list (ACL) that blocks all Telnet access and applies it to an interface, the user loses connectivity. In this example, the only way to recover connectivity is to directly connect to the Console port and remove the ACL.

Once enabled, SafeMode requires the user to periodically reset a reload timer within a specified amount of time. A message will appear on the display:


To reset the SafeMode reload timer, press <Ctrl + r>. If no reset request is received, the router displays the following message to the screen:


The router saves the current disruptive running-configuration to internal flash (labeling it problem-config) and reboots. Once the router has rebooted, it will display a reboot cause message and load the current saved startup-configuration file. The startup-configuration should allow the user to regain access to the router. The user should then review the saved disruptive configuration file (problem-config) and correct the disruption.

Usage Examples

The following example enables SafeMode operation and specifies a reload timer of 240 seconds and a threshold timer of 120 seconds:

ProCurve(config)#safe-mode 240 120
service password-encryption

Use the service password-encryption command to turn on global password protection. Use the no form of this command to return to default settings.

If you need to go back to a previous revision of the code (e.g., SROS J.03.01), this command must be disabled first. Once the service is disabled, all necessary passwords must be re-entered so that they are in the clear text form. If this is not done properly, you will not be able to log back in to the unit after you revert to a previous revision that does not support password encryption.

Syntax Description

No subcommands.

Default Values

By default, global password protection is disabled.

Functional Notes

When enabled, all currently configured passwords are encrypted. Also, any new passwords are encrypted after they are entered. Password encryption is applied to all passwords, including passwords for username, enable, Telnet/console, PPP, BGP, and authentication keys. When passwords are encrypted, unauthorized persons cannot view them in configuration files since the encrypted form of the password is displayed in the running-config. While this provides some level of security, the encryption method used with password encryption is not a strong form of encryption so you should take additional network security measures.

You cannot recover a lost encrypted password. You must erase the startup-config and set a new password.

Usage Examples

The following example enables password encryption for all passwords on the unit:

ProCurve(config)#service password-encryption
**snmp-server chassis-id <id string>**

Use the `snmp-server chassis-id` command to specify an identifier for the Simple Network Management Protocol (SNMP) server. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;id string&gt;</th>
<th>Identifies the product by alphanumeric string (up to 32 characters in length).</th>
</tr>
</thead>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>&lt;id string&gt;</th>
<th>Chassis ID</th>
</tr>
</thead>
</table>

**Usage Examples**

The following example configures a chassis ID of **A432692**:

```
ProCurve(config)# snmp-server chassis-id A432692
```
snmp-server community <community> view <viewname> [ro | rw] <listname>

Use the `snmp-server community` command to specify a community string to control access to Simple Network Management Protocol (SNMP) information. Use the `no` form of this command to remove a specified community.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;community&gt;</code></td>
<td>Specifies the community string (a password to grant SNMP access).</td>
</tr>
<tr>
<td><code>view &lt;viewname&gt;</code></td>
<td>Optional. Specifies a previously defined view. Views define objects available to the community. For information on creating a new view, see `snmp-server view &lt;viewname&gt; &lt;oidtree&gt; [excluded</td>
</tr>
<tr>
<td><code>ro</code></td>
<td>Optional. Keyword to grant read-only access, allowing retrieval of MIB objects.</td>
</tr>
<tr>
<td><code>rw</code></td>
<td>Optional. Keyword to grant read-write access, allowing retrieval and modification of MIB objects.</td>
</tr>
<tr>
<td><code>&lt;listname&gt;</code></td>
<td>Optional. Specifies an access-control list name used to limit access. Refer to <code>ip access-list extended &lt;listname&gt;</code> on page 350 and <code>ip access-list standard &lt;listname&gt;</code> on page 354 for more information on creating access-control lists.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no configured SNMP communities.

**Usage Examples**

The following example specifies a community named `MyCommunity`, specifies a previously defined view named `blockinterfaces`, and assigns read-write access:

```
ProCurve(config)#snmp-server community MyCommunity view blockinterfaces rw
```
**snmp-server contact [email | pager | phone] <number>**

Use the `snmp-server contact` command to specify the email address, pager number, or phone number. Use the `no` form of this command to remove a configured contact.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>email</strong></td>
<td>Specifies email address for the SNMP server contact.</td>
</tr>
<tr>
<td><strong>pager</strong></td>
<td>Specifies pager number for the SNMP server contact.</td>
</tr>
<tr>
<td><strong>phone</strong></td>
<td>Specifies phone number for the SNMP server contact.</td>
</tr>
<tr>
<td><strong>&lt;number&gt;</strong></td>
<td>Identifies the contact (up to 32 characters in length).</td>
</tr>
</tbody>
</table>

**Default Values**

No default values necessary for this command.

**Usage Examples**

The following example specifies 5551212 for the pager number:

```
ProCurve(config)#snmp-server contact pager 5551212
```
snmp-server enable traps <trap type> [snmp]

Use the `snmp-server enable traps` command to enable all Simple Network Management Protocol (SNMP) traps available on your system or specified using the `<trap type>` option. Use multiple `snmp-server enable traps` to enable multiple trap types. Use the `no` form of this command to disable traps (or the specified traps).

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;trap type&gt;</th>
<th>Optional. Specifies the type of notification trap to enable. Leaving this option blank enables ALL system traps.</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp</td>
<td>Optional. Enables a subset of traps specified in RFC 1157. The following traps are supported: coldStart, warmStart, linkUp, linkDown, authenticationFailure</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no enabled traps.

**Usage Examples**

The following example enables the SNMP traps:

ProCurve(config)#snmp-server enable traps snmp
snmp-server host <address> traps <community> <trap type> [snmp]

Use the snmp-server host traps command to specify traps sent to an identified host. Use multiple snmp-server host traps commands to specify all desired hosts. Use the no form of this command to return to the default value.

Syntax Description

- **<address>** Specifies the IP address of the SNMP host that receives the traps.
- **<community>** Specifies the community string (used as a password) for authorized agents to obtain access to SNMP information.
- **<trap type>** Optional. Specifies the type of notification trap to enable. Leaving this option blank enables ALL system traps.
- **snmp** Optional. Enables a subset of traps specified in RFC 1157.

The following traps are supported:
- coldStart
- warmStart
- linkUp
- linkDown
- authenticationFailure

Default Values

By default, there are no hosts or traps enabled.

Usage Examples

The following example sends all SNMP traps to the host at address 190.34.69 and community string My Community:

ProCurve(config)#snmp-server host 190.34.69 traps My Community snmp
**snmp-server host <address> traps version <version> <community> <trap type> [snmp]**

Use the `snmp-server host traps version` command to specify traps sent to an identified host. Use multiple `snmp-server host traps version` commands to specify all desired hosts. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Specifies the IP address of the SNMP host that receives the traps.</td>
</tr>
<tr>
<td><code>&lt;version&gt;</code></td>
<td>Specifies the SNMP version as one of the following:</td>
</tr>
<tr>
<td></td>
<td>1 - SNMPv1</td>
</tr>
<tr>
<td></td>
<td>2C - SNMPv2C</td>
</tr>
<tr>
<td><code>&lt;community&gt;</code></td>
<td>Specifies the community string (used as a password) for authorized agents to</td>
</tr>
<tr>
<td></td>
<td>obtain access to SNMP information.</td>
</tr>
<tr>
<td><code>&lt;trap type&gt;</code></td>
<td>Optional. Specifies the type of notification trap to enable. Leaving this</td>
</tr>
<tr>
<td></td>
<td>option blank enables ALL system traps.</td>
</tr>
<tr>
<td><code>snmp</code></td>
<td>Optional. Enables a subset of traps specified in RFC 1157.</td>
</tr>
<tr>
<td></td>
<td>The following traps are supported:</td>
</tr>
<tr>
<td></td>
<td>coldStart</td>
</tr>
<tr>
<td></td>
<td>warmStart</td>
</tr>
<tr>
<td></td>
<td>linkUp</td>
</tr>
<tr>
<td></td>
<td>linkDown</td>
</tr>
<tr>
<td></td>
<td>authenticationFailure</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no hosts or traps enabled.

**Usage Examples**

The following example sends all SNMP traps to the host at address **190.3.44.69** and community string **My Community** using SNMPv2C:

```
ProCurve(config)#snmp-server host 190.3.44.69 traps version 2c My Community snmp
```
**snmp-server location** `<"string">`

Use the **snmp-server location** command to specify the Simple Network Management Protocol (SNMP) system location string. Use the **no** form of this command to return to the default value.

**Syntax Description**

```
"<string>"  Alphanumeric string encased in quotation marks (up to 32 characters in length) used to populate the system location string.
```

**Default Values**

```
<string>  ProCurve
```

**Usage Examples**

The following example specifies a location of **5th Floor Network Room**:

```
ProCurve(config)#snmp-server location "5th Floor Network Room"
```
**snmp-server management-url** `<URL>`

Use the `snmp-server management-url` command to specify the URL for the device’s management software. Use the no form of this command to remove the management URL.

**Syntax Description**

| `<URL>` | Specifies the URL for the management software. |

**Default Values**

No default is necessary for this command.

**Usage Examples**

The following example specifies the URL `http://www.mywatch.com` as the device’s management software:

```
ProCurve(config)#snmp-server management-url http://www.mywatch.com
```
snmp-server management-url-label <label>

Use the `snmp-server management-url-label` command to specify a label for the URL of the device’s management software. Use the `no` form of this command to remove the label.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;label&gt;</code></th>
<th>Specifies a label for the URL of the management software (maximum length 255 characters).</th>
</tr>
</thead>
</table>

**Default Values**

No default is necessary for this command.

**Usage Examples**

The following example specifies the label `watch` for the management software:

```
ProCurve(config)#snmp-server management-url-label watch
```
snmp-server source-interface <interface>

Use the `snmp-server source-interface` command to tell the SROS where to expect Simple Network Management Protocol (SNMP) traps to originate from (interface type). All SNMP originated packets (including traps and get/set requests) will use the designated interface’s IP address. Use the `no` form of this command to remove specified interfaces.

**Syntax Description**

| <interface> | Specifies the physical interface that should originate SNMP traps. Enter `snmp-server source-interface ?` for a complete list of valid interfaces. |

**Default Values**

By default, there are no trap-source interfaces defined.

**Usage Examples**

The following example specifies that the Ethernet interface (`ethernet 0/1`) should be the source for all SNMP traps and get/set requests:

```
ProCurve(config)#snmp-server source-interface ethernet 0/1
```
**snmp-server view** `<viewname>` `<oidtree>` [excluded | included]

Use the `snmp-server view` command to create or modify a Simple Network Management Protocol (SNMP) view entry. Use the `no` form of this command to remove an entry.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;viewname&gt;</code></td>
<td>Specifies label for the view record being created. The name is a record reference.</td>
</tr>
<tr>
<td><code>&lt;oidtree&gt;</code></td>
<td>Specifies the object identifier (oid) to include or exclude from the view. To identify the subtree, specify a string using numbers, such as 1.4.2.6.8. Replace a single subidentifier with the asterisk (*) to specify a subtree family.</td>
</tr>
<tr>
<td><code>excluded</code></td>
<td>Specifies an excluded view.</td>
</tr>
<tr>
<td><code>included</code></td>
<td>Specifies an included view.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The `snmp-server view` command can include or exclude a group of OIDs. The following example shows how to create a view (named `blockInterfaces`) to exclude the OID subtree family 1.3.3.1.2.1:

```
ProCurve(config)#snmp-server view blockInterfaces 1.3.6.1.2.1.2.* excluded
```

The following example shows how to create a view (named `block`) to include a specific OID:

```
ProCurve(config)#snmp-server view block 1.3.6.1.2.1.2. included
```
sntp retry-timeout <time>

Use the sntp retry-timeout command to set the amount of time to wait for a response before allowing a new request.

Syntax Description

| <time> | Specifies time (in seconds) to wait for a response before retrying. The range is from 3 to 4,294,967,294. |

Default Values

By default, the retry timeout is set to 5 seconds.

Usage Examples

The following example sets the SNTP retry timeout to 10 seconds:

ProCurve(config)#sntp retry-time 10
sntp server <address or hostname> version <1-3>

Use the sntp server command to set the hostname of the SNTP server as well as the version of SNTP to use. The Simple Network Time Protocol (SNTP) is an abbreviated version of the Network Time Protocol (NTP). SNTP is used to set the time of the SROS product over a network. The SNTP server usually serves the time to many devices within a network.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address or hostname&gt;</td>
<td>Specifies the IP address or hostname of the SNTP server.</td>
</tr>
<tr>
<td>version &lt;1-3&gt;</td>
<td>Specifies which NTP version is used (1-3).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, version is set to 1.

**Usage Examples**

The following example sets the SNTP server to `time.nist.gov` using SNTP version 1 (the default version):

ProCurve(config)#sntp server time.nist.gov

The following example sets the SNTP server as `time.nist.gov`. All requests for time use version 2 of the SNTP:

ProCurve(config)#sntp server time.nist.gov version 2
**sntp wait-time <time>**

Use the **sntp wait-time** command to set the time between updates from the time server.

**Syntax Description**

| <time> | Specifies time (in seconds) between updates. The range is from 10 to 4,294,967,294. |

**Default Values**

By default, the wait time is set to 86400 seconds (1 day).

**Usage Examples**

The following example sets the SNTP wait time to two days:

ProCurve(config)#sntp wait-time 172800
spanning-tree edgeport bpdufilter default

Use the spanning-tree edgeport bpdufilter default command to enable the BPDU filter on all ports by default. Use the no form of this command to disable the setting.

Syntax Description

No subcommands.

Default Values

Disabled by default.

Functional Notes

The BPDU filter blocks any BPDUs from being transmitted and received on an interface. This can be overridden on an individual port.

Usage Examples

The following example enables the bpdufilter on all ports by default:

ProCurve(config)#spanning-tree edgeport bpdufilter default

To disable the BPDU filter on a specific interface, issue the appropriate commands for the given interface using the following commands as an example:

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#spanning-tree bpdufilter disable
**spanning-tree edgeport bpduguard default**

Use the `spanning-tree edgeport bpduguard default` command to enable the BPDU guard on all ports by default. Use the `no` form of this command to disable the setting.

**Syntax Description**

No subcommands.

**Default Values**

Disabled by default.

**Functional Notes**

The `bpduguard` blocks any BPDUs from being received on an interface. This can be overridden on an individual port.

**Usage Examples**

The following example enables the BPDU guard on all ports by default.

```
ProCurve(config)#spanning-tree bpduguard default
```

To disable the BPDU guard on a specific interface, issue the appropriate commands for the given interface using the following commands as an example:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#spanning-tree bpduguard disable
```
spanning-tree edgeport default

Use the `spanning-tree edgeport default` command to configure all ports to be edgeports by default. Use the `no` form of this command to disable the setting.

**Syntax Description**

No subcommands.

**Default Values**

Disabled by default.

**Usage Examples**

The following example configures all interfaces running spanning tree to be edgeports by default:

```
ProCurve(config)#spanning-tree edgeport default
```

An individual interface can be configured to not be considered an edgeport. For example:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#no spanning-tree edgeport
```
spanning-tree forward-time <seconds>

Use the `spanning-tree forward-time` command to specify the delay interval (in seconds) when forwarding spanning-tree packets. Use the `no` form of this command to return to the default interval.

**Syntax Description**

| <seconds> | Forward delay interval in seconds (Range: 4 to 30). |

**Default Values**

| <seconds> | 15 seconds |

**Usage Examples**

The following example sets the forwarding time to 15 seconds:

```
ProCurve(config)#spanning-tree forward-time 15
```
spanning-tree hello-time <seconds>

Use the spanning-tree hello-time command to specify the delay interval (in seconds) between hello bridge protocol data units (BPDUs). To return to the default interval, use the no form of this command.

**Syntax Description**

| <seconds> | Delay interval (in seconds) between hello BPDUs. Range: 0 to 1,000,000. |

**Default Values**

| <seconds> | 2 seconds |

**Usage Examples**

The following example configures a spanning-tree hello-time interval of 10000 seconds:

```plaintext
ProCurve(config)#spanning-tree hello-time 10000
```
spanning-tree max-age <seconds>

Use the spanning-tree max-age command to specify the interval (in seconds) the spanning-tree will wait to receive Bridge Protocol Data Units (BPDUs) from the root bridge before assuming the network has changed (thus re-evaluating the spanning-tree topology). Use the no form of this command to return to the default interval.

Syntax Description

| <seconds> | Wait interval (in seconds) between received BPDUs (from the root bridge). Range: 6 to 40. |

Default Values

| <seconds> | 20 seconds |

Usage Examples

The following example configures a max-age interval of 45 seconds:

ProCurve(config)#spanning-tree max-age 45
spanning-tree mode [rstp | stp]

Use the spanning-tree mode command to choose a spanning-tree mode of operation.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rstp</td>
<td>Enables rapid spanning-tree protocol.</td>
</tr>
<tr>
<td>stp</td>
<td>Enables spanning-tree protocol.</td>
</tr>
</tbody>
</table>

Default Values

By default, this is set to rstp.

Usage Examples

The following example sets the spanning-tree mode to rapid spanning-tree protocol:

ProCurve(config)#spanning-tree mode rstp
spanning-tree pathcost method [short | long]

Use the spanning-tree pathcost command to select a short or long pathcost method used by the spanning-tree protocol.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>Selects a short pathcost method.</td>
</tr>
<tr>
<td>long</td>
<td>Selects a long pathcost method.</td>
</tr>
</tbody>
</table>

Default Values

By default, this is set to short.

Usage Examples

The following example designates the spanning-tree protocol to use a long pathcost method:

ProCurve(config)#spanning-tree pathcost method long
spanning-tree priority <value>

Use the `spanning-tree priority` command to set the priority for spanning-tree interfaces. The lower the priority value, the higher the likelihood the configured spanning-tree interface will be the root for the bridge group. To return to the default bridge priority value, use the `no` version of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;value&gt;</th>
<th>Priority value for the bridge interface. Configuring this value to a low number increases the interface’s chance of being the root. Therefore, the maximum priority level would be 0. Range: 0 to 65,535.</th>
</tr>
</thead>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>&lt;value&gt;</th>
<th>32768</th>
</tr>
</thead>
</table>

**Usage Examples**

The following example sets `spanning-tree priority` to the maximum level:

```
ProCurve(config)#spanning-tree priority 0
```
tacacs-server

Use the `tacacs-server` command to customize setting for communication with TACACS servers. Use the `no` form of this command to return to default settings.

Variations of this command include the following:

- `tacacs-server host <hostname or IP address>`
- `tacacs-server host <hostname or IP address> key <key>`
- `tacacs-server host <hostname or IP address> port <TCP port>`
- `tacacs-server host <hostname or IP address> timeout <seconds>`
- `tacacs-server key <key>`
- `tacacs-server packet maxsize <maximum packet size>`
- `tacacs-server timeout <seconds>`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>host &lt;name/IP&gt;</code></td>
<td>Specifies the IP host by name or IP address.</td>
</tr>
<tr>
<td><code>key &lt;key&gt;</code></td>
<td>Sets an encryption string to be used for encrypting and decrypting the traffic between the Network Access Server (NAS) and the TACACS+ daemon. Setting a key for a particular server (using the <code>tacacs-server host &lt;name/IP&gt; key &lt;key&gt;</code> command) supersedes keys set globally using the <code>tacacs-server key &lt;key&gt;</code> command.</td>
</tr>
<tr>
<td><code>port &lt;tcp port&gt;</code></td>
<td>Specifies the TCP port number to be used when connecting to the TACACS+ daemon.</td>
</tr>
<tr>
<td><code>timeout &lt;seconds&gt;</code></td>
<td>Specifies a timeout limit (in seconds) that the unit will wait for a response from the daemon before declaring an error. Range is 1 to 1000 seconds. Setting a timeout for a particular server (using the <code>tacacs-server host &lt;name/IP&gt; timeout &lt;seconds&gt;</code> command) supersedes time limits set globally using the <code>tacacs-server timeout &lt;seconds&gt;</code> command.</td>
</tr>
<tr>
<td><code>packet maxsize &lt;size&gt;</code></td>
<td>Specifies a maximum packet size for this server. Range is 10,240 to 65,535.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the key is set to `key` and the default TCP port number is 49.

**Usage Examples**

The following example sets a timeout limit of 60 seconds for the specified server:

```
ProCurve(config)#tacacs-server host 10.5.6.7 timeout 60
```
**thresholds [BES | CSS | DM | ES | LCV | LES | PCV | SEFS | SES | UAS] [15Min | 24Hr] <threshold count>**

Use the `thresholds` command to specify DS1 performance counter thresholds. Use the `no` form of this command to return to default settings.

**Note** *Threshold settings are applied to ALL DSIs.*

**Syntax Description**

- **BES** Specifies the bursty errored seconds threshold.
- **CSS** Specifies the controlled slip seconds threshold.
- **DM** Specifies the degraded minutes threshold.
- **ES** Specifies the errored seconds threshold.
- **LCV** Specifies the line code violations threshold.
- **LES** Specifies the line errored seconds threshold.
- **PCV** Specifies the path coding violations threshold.
- **SEFS** Specifies the severely errored framing seconds threshold.
- **SES** Specifies the severely errored seconds threshold.
- **UAS** Specifies the unavailable seconds threshold.
- **15Min** Specifies that the threshold you are setting is for the counter’s 15 minute statistics.
- **24Hr** Specifies that the threshold you are setting is for the counter’s 24 hour statistics.
- `<threshold>` Specifies the maximum occurrences allowed for this error type. Once a threshold is exceeded, an event is sent to the console specifying the appropriate counter. Additionally, if SNMP traps are enabled, the unit will send a trap with the same information as the console event.

**Default Values**

The default values for this command are as follows:

```
thresholds BES 15Min 10
thresholds BES 24Hr 100
thresholds CSS 15Min 1
thresholds CSS 24Hr 4
thresholds DM 15Min 1
thresholds DM 24Hr 4
thresholds ES 15Min 65
thresholds ES 24Hr 648
thresholds LCV 15Min 13340
thresholds LCV 24Hr 133400
thresholds LES 15Min 65
thresholds LES 24Hr 648
thresholds PCV 15Min 72
thresholds PCV 24Hr 691
```
thresholds SES 15Min 10
thresholds SES 24Hr 100
thresholds SEFS 15Min 2
thresholds SEFS 24Hr 17
thresholds UAS 15Min 10
thresholds UAS 24Hr 10

Usage Examples

The following example sets the threshold for the 15 minute and 24 hour bursty errored seconds counter to 25 and 200, respectively:

ProCurve(config)#thresholds BES 15Min 25
ProCurve(config)#thresholds BES 24Hr 200
**username <username> password <password>**

Use this command to configure the username and password to use for all protocols requiring a username-based authentication system including FTP server authentication, line (login local-user list), and HTTP access.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;username&gt;</td>
<td>Alphanumerical string up to 30 characters in length (the username is case-sensitive)</td>
</tr>
<tr>
<td>&lt;password&gt;</td>
<td>Alphanumerical string up to 30 characters in length (the username is case-sensitive)</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there is no established username and password.

**Functional Notes**

All users defined using the `username/password` command are valid for access to the unit using the `login local-userlist` command.

**Usage Examples**

The following example creates a username of `procurve` with password `procurve`:

```
ProCurve(config)#username procurve password procurve
```
LINE (CONSOLE) INTERFACE CONFIG COMMAND SET

To activate the Line (Console) Interface Configuration mode, enter the line console 0 command at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#line console 0
ProCurve(config-con 0)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- do on page 1308
- end on page 1309
- exit on page 1310
- ping <address> on page 1311

All other commands for this command set are described in this section in alphabetical order.

- accounting commands [<level> l <name> l default] on page 491
- authorization commands [<level> l <name> l default] on page 492
- databits [7 | 8] on page 493
- flowcontrol [none | software in] on page 494
- line-timeout <minutes> on page 495
- login on page 496
- login authentication <aaa login list> on page 497
- login local-userlist on page 498
- parity [even | mark | none | odd | space] on page 499
- password [md5] <password> on page 500
- speed <rate> on page 501
- stopbits [1 | 2] on page 502
accounting commands [<level> | <name> | default]

Use the `accounting commands` command to assign AAA accounting methods to lines. You must first turn AAA on for this command to become available. Use the `no` form of this command to disable this feature.

**Syntax Description**

- `<level>` Specifies a command level (1 or 15).
- `<name>` Applies a named accounting method to this line.
- `default` Applies the default accounting method to a line.

**Default Values**

The default for this command is `off`.

**Usage Examples**

The following example applies the default accounting method to all Console sessions:

```
ProCurve(config)#aaa on
ProCurve(config)#line console 0
ProCurve(config-con 0)#accounting commands 1 default
```
authorization commands [<level> | <name> | default]

Use the authorization commands command to assign AAA authorization methods to lines. You must first turn AAA on for this command to become available. Use the no form of this command to disable this feature.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;level&gt;</td>
<td>Specifies a command level (1 or 15).</td>
</tr>
<tr>
<td>&lt;name&gt;</td>
<td>Applies a named authorization method to this line.</td>
</tr>
<tr>
<td>default</td>
<td>Applies the default authorization method to a line.</td>
</tr>
</tbody>
</table>

Default Values

The default for this command is off.

Usage Examples

The following example applies the default authorization method to all Console sessions:

```
ProCurve(config)#aaa on
ProCurve(config)#line console 0
ProCurve(config-con 0)#authorization commands 1 default
```
**databits [7 | 8]**

The `databits` command is used to set the number of databits per character for a terminal session. This value must match the configuration of your VT100 terminal or terminal emulator software. The default is 8 databits per character. Use the `no` form of this command to return to the default value.

### Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Specifies 7 data bits per character.</td>
</tr>
<tr>
<td>8</td>
<td>Specifies 8 data bits per character.</td>
</tr>
</tbody>
</table>

### Default Values

By default, console terminal sessions use 8 data bits.

### Usage Examples

The following example configures 7 databits per character for the console terminal session:

```
ProCurve(config)#line console 0
ProCurve(config-console 0)#databits 7
```
flowcontrol [none | software in]

Use the `flowcontrol` command to set flow control for the line console. Use the `no` form of this command to return to the default setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Specifies no flow control.</td>
</tr>
<tr>
<td>software in</td>
<td>Configures the SROS to derive flow control from the attached device.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, flow control is set to none.

**Usage Examples**

The following example configures no flow control for the line console:

```
ProCurve(config)#line console 0
ProCurve(config-con 0)#flowcontrol none
```
**line-timeout <minutes>**

Use the `line-timeout` command to specify the number of minutes a line session may remain inactive before the SROS terminates the session. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;minutes&gt;</th>
<th>Specifies the number of minutes a line session may remain inactive before the SROS terminates the session.</th>
</tr>
</thead>
</table>

Entering a `line-timeout` value of 0 disables the feature.

**Default Values**

By default the `line-timeout` is set to 15 minutes (Console and Telnet).

**Usage Examples**

The following example specifies a timeout of 2 minutes:

```
ProCurve(config)#line console 0
ProCurve(config-con 0)#line-timeout 2
```
**login**

Use the `login` command to enable security login on the line session requiring the password configured using the `password` command. Use the `no` form of this command to disable the login feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, there is no login password set for access to the unit.

**Usage Examples**

The following example enables the security login feature and specifies a password on the available console session:

```
ProCurve(config)#line console 0
ProCurve(config-console 0)#login
ProCurve(config-console 0)#password mypassword
```
login authentication <aaa login list>

Use the login authentication command to specify the named AAA login list to use for authenticating users connecting on this line. Use the no form of this command to return to the default setting.

Syntax Description

<aaa login list> Specifies the AAA login list to use for authentication.

Default Values

The default value is the default AAA list.

Functional Notes

If the AAA subsystem is activated but no login authentication list is given, the default list is used. If the default list is used but the default list is not configured, the behavior for consoles is to be granted access. This prevents a lockout configuration.

Usage Examples

The following example specifies that myList will be used for authenticating users connecting on this line:

ProCurve(config)#line console 0
ProCurve(config-con 0)#login authentication myList
login local-userlist

Use the login local-userlist command to enable security login for the terminal session requiring the usernames and passwords configured using the username/password Global Configuration command. Use the no form of this command to disable the login local-userlist feature.

**Note**

All user properties assigned using the username/password command are valid when using the *login local-userlist* command.

**Syntax Description**

No subcommands.

**Default Values**

By default, there is no login password set for access to the unit.

**Usage Examples**

The following example displays creating a local userlist and enabling the security login feature on the CONSOLE port:

```plaintext
ProCurve(config)#username my_user password my_password
ProCurve(config)#line console 0
ProCurve(config-con 0)#login local-userlist
```

When connecting to the unit, the following prompts are displayed:

User Access Login
Username: Procurve
Password:
Router#
parity [even | mark | none | odd | space]

Use the `parity` command to specify the type of parity used as error correction. This value must match the configuration of your VT100 terminal or terminal emulator software. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>even</td>
<td>Set the parity bit to 0 if the number of 1 bits in the data sequence is odd, or set to 1 if the number of 1 bits is even.</td>
</tr>
<tr>
<td>mark</td>
<td>Always set the parity bit to 1.</td>
</tr>
<tr>
<td>none</td>
<td>No parity bit used.</td>
</tr>
<tr>
<td>odd</td>
<td>Set the parity bit to 1 if the number of 1 bits in the data sequence is even, or set to 1 if the number is odd.</td>
</tr>
<tr>
<td>space</td>
<td>Always set the parity bit to 0.</td>
</tr>
</tbody>
</table>

**Default Values**

```
<option> none
```

**Functional Notes**

Parity is the process used to detect whether characters have been altered during the data transmission process. Parity bits are appended to data frames to ensure that parity (whether it be odd or even) is maintained.

**Usage Examples**

The following example specifies mark parity for the console terminal session:

```
ProCurve(config)#line console 0
ProCurve(config-con 0)#parity mark
```
password [md5] <password>

Use the password command to configure the password (with optional encryption) required on the line session when security login is enabled (using the login command). Use the no form of this command to remove a configured password.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>md5</td>
<td>Optional. Specifies Message Digest 5 (md5) as the encryption protocol to use when displaying the enable password during show commands. If the md5 keyword is not used, encryption is not used when displaying the enable password during show commands.</td>
</tr>
<tr>
<td>&lt;password&gt;</td>
<td>Alphanumeric character string (up to 16 characters) used to specify the password for the line session.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there is no login password set for access to the unit.

**Usage Examples**

The following example enables the security login feature and specifies a password on the CONSOLE port:

```
ProCurve(config)#line console 0
ProCurve(config-con 0)#login
ProCurve(config-con 0)#password mypassword
```

To provide extra security, the SROS can encrypt the enable password when displaying the current configuration. For example, the following is a show configuration printout (password portion) with an unencrypted enable password (procurve):

```
! enable password procurve
! Alternately, the following is a show configuration printout (password portion) with an enable password of procurve using md5 encryption:

! enable password md5 encrypted 5aa5fbae7d01a90e79fb57705ce74676
!```
speed <rate>

Use the `speed` command to specify the data rate for the Console port. This setting must match your VT100 terminal emulator or emulator software. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;rate&gt;</th>
<th>Specifies rate of data transfer on the interface (2400; 4800; 9600; 19,200; 38,400; 57,600; or 115,200 bps).</th>
</tr>
</thead>
</table>

**Default Values**

By default, the speed is set to 9600 bps.

**Usage Examples**

The following example configures the Console port for 19200 bps:

```
ProCurve(config)#line console 0
ProCurve(config-con 0)#speed 19200
```
stopbits [1 | 2]

Use the stopbits command to set the number of stopbits per character for a terminal session. This value must match the configuration of your VT100 terminal or terminal emulator software. The default is 1 stopbit per character. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specifies 1 stopbit per character.</td>
</tr>
<tr>
<td>2</td>
<td>Specifies 2 stopbits per character.</td>
</tr>
</tbody>
</table>

Default Values

By default, stopbits is set to 1.

Usage Examples

The following example configures 2 stopbits per character for the console terminal session:

```
ProCurve(config)#line console 0
ProCurve(config-con 0)#stopbits 2
```
To activate the Line (Telnet) Interface Configuration mode, enter the `line telnet` command specifying a Telnet session(s) at the Global Configuration mode prompt. For example:

ProCurve>`enable
ProCurve#`configure terminal
ProCurve(config)#`line telnet 0 4
ProCurve(config-telnet0-4)#

You can select a single line by entering the `line telnet` command followed by the line number (0-4). For example:

ProCurve>`enable
ProCurve#`configure terminal
ProCurve(config)#`line telnet 2
ProCurve(config-telnet2)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `access-class <listname> in` on page 504
- `accounting commands [<level> l <name> l default]` on page 505
- `authorization commands [<level> l <name> l default]` on page 506
- `line-timeout <minutes>` on page 507
- `login` on page 508
- `login authentication <aaa login list>` on page 509
- `login local-userlist` on page 510
- `password [md5] <password>` on page 511
**access-class** `<listname>` **in**

Use the `access-class in` command to restrict Telnet access using a configured access list. Received packets passed by the access list will be allowed. Use the access list configuration to deny hosts or entire networks or to permit specified IP addresses.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;listname&gt;</code></th>
<th>Alphanumeric descriptor for identifying the configured access list (all access list descriptors are case-sensitive).</th>
</tr>
</thead>
</table>

**Default Values**

By default, there are no configured access lists associated with Telnet sessions.

**Functional Notes**

When using the `access-class in` command to associate an access list with a Telnet session, remember to duplicate the `access-class in` command for all configured Telnet sessions 0 through 4. Telnet access to the unit using a particular Telnet session is not possible. Users will be assigned the first available Telnet session.

**Usage Examples**

The following example associates the access list `Trusted` (to allow Telnet sessions from the 192.68.56.0/24 network) with all Telnet sessions (0 through 4):

Create the access list:
```
ProCurve(config)#ip access-list standard Trusted
ProCurve(config)#permit 192.168.56.0 0.0.0.255
```

Enter the line (telnet):
```
ProCurve(config)#line telnet 0 4
```

Associate the access list with the Telnet session:
```
ProCurve(config-telnet0-4)#access-class Trusted in
```
accounting commands [<level> | <name> | default]

Use the `accounting commands` command to assign AAA accounting methods to lines. You must first turn AAA on for this command to become available.

Syntax Description

- `<level>` Specifies a command level (1 or 15).
- `<name>` Applies a named accounting method to this line.
- `default` Applies the default accounting method to a line.

Default Values

The default for this command is `off`.

Usage Examples

The following example applies the default accounting method to Telnet session 1:

```
ProCurve(config)#aaa on
ProCurve(config)#line telnet 1
ProCurve(config-telnet1)#accounting commands 1 default
```
**authorization commands [<level> I <name> I default]**

Use the *authorization commands* command to assign AAA authorization methods to lines. You must first turn AAA on for this command to become available.

**Syntax Description**

| **<level>** | Specifies a command level (1 or 15). |
| **<name>** | Applies a named authorization method to this line. |
| **default** | Applies the default authorization method to a line. |

**Default Values**

The default for this command is *off*.

**Usage Examples**

The following example applies the default authorization method to line 1:

```
ProCurve(config)#aaa on
ProCurve(config)#line telnet 1
ProCurve(config-telnet1)#authorization commands 1 default
```
line-timeout <minutes>

Use the line-timeout command to specify the number of minutes a line session may remain inactive before the SROS terminates the session. Use the no form of this command to return to the default value.

Syntax Description

<minutes> Specifies the number of minutes a line session may remain inactive before the SROS terminates the session.

Entering a line-timeout value of 0 disables the feature.

Default Values

<minutes> 15 minutes (Console and Telnet)

Usage Examples

The following example specifies a timeout of 2 minutes:

ProCurve(config)#line telnet 0
ProCurve(config-telnet0)#line-timeout 2
login

Use the login command to enable security login on the line session requiring the password configured using the password command. Use the no form of this command to disable the login feature.

Syntax Description

No subcommands.

Default Values

By default, there is no login password set for access to the unit.

Usage Examples

The following example enables the security login feature and specifies a password on all the available Telnet sessions (0 through 4):

ProCurve(config)#line telnet 0 4
ProCurve(config-telnet0-4)#login
ProCurve(config-telnet0-4)#password mypassword
**login authentication** `<aaa login list>`

Use the `login authentication` command to specify the named AAA login list to use for authenticating users connecting on this line.

**Syntax Description**

| `<aaa login list>` | Specifies the AAA login list to use for authentication. |

**Default Values**

The default value is the default AAA list.

**Functional Notes**

If the AAA subsystem is activated but no login authentication list is given, the default list is used. If the default list is used but the default list is not configured, the behavior for telnets is to use the local user database.

**Usage Examples**

The following example specifies that `myList` will be used for authenticating users connecting on this line:

```
ProCurve(config)#line telnet 2
ProCurve(config-telnet2)#login authentication myList
```
login local-userlist

Use the `login local-userlist` command to enable security login for the terminal session requiring the usernames and passwords configured using the `username/password` Global Configuration command. Use the `no` form of this command to disable the login local-userlist feature.

**Note**
All user properties assigned using the `username/password` command are valid when using the `login local-userlist` command.

**Syntax Description**
No subcommands.

**Default Values**
By default, there is no login password set for access to the unit.

**Usage Examples**
The following example displays creating a local userlist and enabling the security login feature:

```plaintext
ProCurve(config)#username my_user password my_password
ProCurve(config)#line telnet 0
ProCurve(config-telnet0)#login local-userlist
```

When connecting to the unit, the following prompts are displayed:
User Access Login
Username: **my_user**
Password:
Router#
password [md5] <password>

Use the password command to configure the password (with optional encryption) required on the line session when security login is enabled (using the login command). Use the no form of this command to remove a configured password.

**Syntax Description**

<table>
<thead>
<tr>
<th>md5</th>
<th>Optional. Specifies Message Digest 5 (md5) as the encryption protocol to use when displaying the enable password during show commands. If the md5 keyword is not used, encryption is not used when displaying the enable password during show commands.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;password&gt;</td>
<td>Alphanumeric character string (up to 16 characters) used to specify the password for the line session.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there is no login password set for access to the unit.

**Usage Examples**

The following example enables the security login feature and specifies a password for the Telnet session 0:

```
ProCurve(config)#line telnet 0
ProCurve(config-telnet0)#login
ProCurve(config-telnet0)#password mypassword
```

To provide extra security, the SROS can encrypt the enable password when displaying the current configuration. For example, the following is a show configuration printout (password portion) with an unencrypted enable password (procurve):

```
! enable password procurve
```

Alternately, the following is a show configuration printout (password portion) with an enable password of procurve using md5 encryption:

```
! enable password md5 encrypted 5aa5fbae7d01a90e79fb57705ce74676
```
To activate the Line Secure Shell (SSH) Interface Configuration mode, enter the `line ssh` command specifying a SSH session(s) at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# line ssh 0 4
ProCurve(config-ssh0-4)#
```

You can select a single line by entering the `line ssh` command followed by the line number (0-4). For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# line ssh 2
ProCurve(config-ssh2)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `access-class <listname>` in on page 513
- `accounting commands [ <level> l <name> l default]` on page 514
- `authorization commands [ <level> l <name> l default]` on page 515
- `line-timeout <minutes>` on page 516
- `login authentication <aaa login list>` on page 517
- `login local-userlist` on page 518
**access-class** `<listname>` **in**

Use the **access-class in** command to restrict Secure Shell (SSH) access using a configured access list. Received packets passed by the access list will be allowed. Use the access list configuration to deny hosts or entire networks or to permit specified IP addresses. See **ip access-list standard `<listname>`** on page 354 and **ip access-list extended `<listname>`** on page 350 for more information about configuring access lists.

**Syntax Description**

| `<listname>` | Identifies the configured access list using an alphanumeric descriptor (all access list descriptors are case-sensitive). |

**Default Values**

By default, there are no configured access lists associated with SSH sessions.

**Functional Notes**

When using the **access-class in** command to associate an access list with an SSH session, remember to duplicate the **access-class in** command for all configured SSH sessions 0 through 4. SSH access to the unit using a particular SSH session is not possible. Users will be assigned the first available SSH session.

**Usage Examples**

The following example associates the access list **Trusted** (to allow SSH sessions from the 192.168.56.0/24 network) with all SSH sessions (0 through 4):

Create the access list:

ProCurve(config)#**ip access-list standard Trusted**  
ProCurve(config)#**permit 192.168.56.0 0.0.0.255**

Enter the line (ssh):

ProCurve(config)#**line ssh 0 4**

Associate the access list with the SSH session:

ProCurve(config-ssh0-4)#**access-class Trusted in**
accounting commands [<level> | <name> | default]

Use the accounting commands command to assign AAA accounting methods to lines. You must first turn AAA on for this command to become available.

Syntax Description

- `<level>` Specifies a command level (1 or 15).
- `<name>` Applies a named accounting method to this line.
- `default` Applies the default accounting method to a line.

Default Values

By default, AAA accounting methods are not applied to SSH lines.

Usage Examples

The following example applies the default accounting method to line 1:

```
ProCurve(config)#aaa on
ProCurve(config)#line ssh 1
ProCurve(config-ssh1)#accounting commands 1 default
```
authorization commands [<level> | <name> | default]

Use the authorization commands command to assign AAA authorization methods to lines. You must first turn AAA on for this command to become available.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;level&gt;</td>
<td>Specifies a command level (1 or 15).</td>
</tr>
<tr>
<td>&lt;name&gt;</td>
<td>Applies a named authorization method to this line.</td>
</tr>
<tr>
<td>default</td>
<td>Applies the default authorization method to a line.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, AAA authorization methods are not applied to SSH lines.

**Usage Examples**

The following example applies the default authorization method to line 1:

ProCurve(config)#aaa on
ProCurve(config)#line ssh 1
ProCurve(config-ssh1)#authorization commands 1 default
line-timeout <minutes>

Use the line-timeout command to specify the number of minutes a line session may remain inactive before the SROS terminates the session. Use the no form of this command to return to the default value.

Syntax Description

<minutes> Specifies the number of minutes a line session may remain inactive before the SROS terminates the session. Valid range: 0 to 35,791.

Entering a line-timeout value of 0 disables the feature.

Default Values

By default the line-timeout is set to 15 minutes.

Usage Examples

The following example specifies a timeout of 2 minutes for all SSH sessions:

ProCurve(config)#line ssh 0 4
ProCurve(config-ssh0-4)#line-timeout 2
login authentication <aaa login list>

Use the login authentication command to assign the named AAA login list to use for authenticating users connecting on this line. Use the no form of the command to remove the AAA authentication list.

Syntax Description

| <aaa login list> | Specifies the name of the AAA login list to use for authentication. |

Default Values

The default value is the default AAA list.

Functional Notes

If the AAA subsystem is activated but no login authentication list is given, the default list is used. If the default list is used but the default list is not configured, SSH uses the local user database.

Usage Examples

The following example specifies that myList will be used for authenticating users connecting on this line:

ProCurve(config)#line ssh 2
ProCurve(config-ssh2)#login authentication myList
login local-userlist

Use the `login local-userlist` command to check the local list of usernames and passwords configured using the `username/password` Global Configuration command (see `username <username> password <password>` on page 489). Use the `no` form of this command to disable the login local-userlist feature.

**Note**  
All user properties assigned using the `username/password` command are valid when using the `login local-userlist` command.

**Syntax Description**
No subcommands.

**Default Values**
By default, there is no login password set for access to the unit.

**Usage Examples**
The following example creates a local userlist and enables the security login feature:

```
ProCurve(config)#username my_user password my_password
ProCurve(config)#line ssh 0
ProCurve(config-ssh0)#login local-userlist
```

When connecting to the unit, the following prompts are displayed:

```
User Access Login
Username: my_user
Password: #
```
To activate the ADSL Interface Configuration mode, enter the `interface adsl` command at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# interface adsl 1/1
ProCurve(config-adsl 1/1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `retrain` on page 520
- `snr-margin <margin>` on page 521
- `snr-margin [showtime-monitor | training-monitor]` on page 522
- `training-mode [ADSL2 | ADSL2+ | G.DMT | G.LITE | Multi-Mode | READSL2 | T1.413]` on page 523
retrain

Use the `retrain` command to force the modem to retrain.

**Syntax Description**

No subcommands.

**Default Values**

No default is necessary for this command.

**Usage Examples**

The following example forces a modem retrain:

```plaintext
ProCurve(config)#interface adsl1/1
ProCurve(config-adsl 1/1)#retrain
```
**snr-margin** `<margin>`

Use the `snr-margin` command to set the minimum Signal to Noise Ratio margin (in dB).

**Syntax Description**

`<margin>` Sets the minimum SNR margin in dB. The range is from 1 to 15.

**Default Values**

By default, SNR margin is 0 dB.

**Usage Examples**

The following example sets the SNR margin to a minimum level of 3 dB:

```
ProCurve(config)#interface adsl 1/1
ProCurve(config-adsl 1/1)#snr-margin 3
```
snr-margin [showtime-monitor | training-monitor]

Use the snr-margin monitor command to enable monitoring during training and showtime. Use the no form of this command to disable monitoring.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>showtime-monitor</td>
<td>Enables margin monitoring to retrain the ADSL interface if the specified minimum margin is violated during showtime.</td>
</tr>
<tr>
<td>training-monitor</td>
<td>Enables margin monitoring to retrain the ADSL interface if the specified minimum margin is violated during training.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, SNR margin monitoring is disabled.

**Usage Examples**

The following example enables SNR margin monitoring during showtime:

```plaintext
ProCurve(config)#interface adsl 1/1
ProCurve(config-adsl 1/1)#snr-margin showtime-monitor
```
training-mode [ADSL2 | ADSL2+ | G.DMT | G.LITE | Multi-Mode | READSL2 | T1.413]

Use the training mode command to configure the ADSL training mode.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSL2</td>
<td>Specifies ITU G.992.3 Annex A mode.</td>
</tr>
<tr>
<td>ADSL2+</td>
<td>Specifies ITU G.992.5 ADSL2+ mode.</td>
</tr>
<tr>
<td>G.DMT</td>
<td>Specifies ANSI full-rate mode.</td>
</tr>
<tr>
<td>G.LITE</td>
<td>Specifies ANSI splitterless mode.</td>
</tr>
<tr>
<td>Multi-Mode</td>
<td>Specifies auto detect mode. When set to multi-mode, the ADSL interface attempts to train to the DSLAM using each of the supported training modes until a match is found.</td>
</tr>
<tr>
<td>READLS2</td>
<td>Specifies ITU G.992.3 Annex L mode.</td>
</tr>
<tr>
<td>T1.413</td>
<td>Specifies ANSI T1.413 mode.</td>
</tr>
</tbody>
</table>

Default Values

By default, the training mode is set to Multi-Mode.

Functional Notes

Some of the listed training modes (G.LITE, T1.413, ADSL2, ADSL2+, READSL2) are currently supported for ADSL over POTS (Annex A) and are not valid for ADSL over ISDN (Annex B) modules.

Usage Examples

The following example sets the training mode to T1.413:

ProCurve(config)#interface adsl 1/1
ProCurve(config-adsl 1/1)#training-mode T1.413
**BRI INTERFACE CONFIGURATION COMMAND SET**

To activate the BRI Interface Configuration mode, enter the `interface bri` command at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# interface bri 1/2
ProCurve(config-bri 1/2)#
```

**Note**: The BRI interface number in the example above is shown as **bri 1/2**. This number is based on the interface’s location (slot/port) and could vary depending on the unit’s configuration. Use the `do show interfaces` command to determine the appropriate interface number.

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `bonding commands` begin on page 525
- `caller-id override [always <number> | if-no-cid <number>]` on page 531
- `isdn [ldn 1 | ldn 2] <ldn>` on page 532
- `isdn [spid 1 | spid 2] <spid string> <LDN>` on page 533
- `isdn switch-type [basic-5ess | basic-dms | basic-net3 | basic-ni]` on page 534
- `loopback local [all | b1 | b2 | both]` on page 535
- `loopback network [b1 | b2 | both]` on page 536
- `maintenance [reset | restart-d]` on page 537
- `resource pool-member <pool-name> [ <priority> ]` on page 538
- `test-call [dial <number> | speed 56 | 64] | answer | hangup | hangup channels <1 | 2>` on page 539
bonding txadd-timer <seconds>

Use the bonding txadd-timer command to specify the value (in seconds) for the aggregate call connect timeout. Use the no form of this command to return to the default value.

**Syntax Description**

| <seconds> | Specifies the number of seconds the endpoint will wait for additional channels (to add to the bonded aggregate) before considering the bonding negotiation a failure. |

**Default Values**

By default, the bonding txadd-timer value is set to 50 seconds.

**Functional Notes**

Specifies the length of time both endpoints wait for additional calls to be connected at the end of negotiation before deciding that the bonding call has failed. The factory default setting is sufficient for most calls to connect, although when dialing overseas it may be necessary to lengthen this timer to allow for slower call routing.

**Usage Examples**

The following example defines a bonding txadd-timer value of 95 seconds:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#bonding txadd-timer 95
```
**bonding txcid-timer <seconds>**

Use the `bonding txcid-timer` command to specify the value (in seconds) for the bearer channel (B-channel) negotiation timeout. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;seconds&gt;</code></th>
<th>Specifies the number of seconds the endpoint may negotiate data rates and channel capacities before considering the bonding negotiation a failure.</th>
</tr>
</thead>
</table>

**Default Values**

By default, the `bonding txcid-timer` value is set to 5 seconds.

**Functional Notes**

Specifies the length of time both endpoints attempt to negotiate an agreeable value for bearer channels and channel capacities before deciding the bonding call has failed.

**Usage Examples**

The following example defines a `bonding txcid-timer` value of 8 seconds:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#bonding txcid-timer 8
```
**bonding txdeq-timer <seconds>**

Use the **bonding txdeq-timer** command to specify the value (in seconds) for the network delay equalization timeout. Use the **no** form of this command to return to the default value.

**Syntax Description**

| <seconds> | Specifies the number of seconds the endpoint allots for attempting to equalize the network delay between bearer channels before considering the bonding negotiation a failure. |

**Default Values**

By default, the **bonding txdeq-timer** value is set to 50 seconds.

**Usage Examples**

The following example defines a **bonding txdeq-timer** value of 80 seconds:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#bonding txdeq-timer 80
```
bonding txfa-timer <seconds>

Use the bonding txfa-timer command to specify the value (in seconds) for the frame pattern detection timeout. Use the no form of this command to return to the default value.

Syntax Description

| <seconds> | Specifies the number of seconds the endpoint allots for attempting to detect the bonding frame pattern (when a call is connected) before considering the bonding negotiation a failure. |

Default Values

By default, the bonding txfa-timer value is set to 10 seconds.

Functional Notes

Specifies the length of time the endpoint attempts to detect the bonding frame pattern when a call is connected before deciding the bonding call has failed. When operating with other manufacturers' bonding equipment, it may be necessary to change this time so that it matches TXADD01.

Usage Examples

The following example defines a bonding txfa-timer value of 15 seconds:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#bonding txfa-timer 15
```
bonding txinit-timer <seconds>

Use the bonding txinit-timer command to specify the value (in seconds) for the originating endpoint negotiation timeout. Use the no form of this command to return to the default value.

Syntax Description

- `<seconds>`: Specifies the number of seconds the endpoint waits to detect the bonding negotiation frame pattern from the remote endpoint (when a call is connected) before considering the bonding negotiation a failure.

Default Values

By default, the bonding txinit-timer value is set to 10 seconds.

Functional Notes

Specifies the length of time the originating endpoint attempts to detect the bonding negotiation pattern from the answering endpoint before deciding the bonding call has failed.

Usage Examples

The following example defines a bonding txinit-timer value of 15 seconds:

ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#bonding txinit-timer 15
**bonding txnull-timer** `<seconds>`

Use the `bonding txnull-timer` command to specify the value (in seconds) for the answering endpoint negotiation timeout. Use the `no` form of this command to return to the default value.

**Syntax Description**

| `<seconds>` | Specifies the number of seconds the endpoint waits to detect the bonding negotiation frame pattern from the originating endpoint (after answering a call) before considering the bonding negotiation a failure. |

**Default Values**

By default, the `bonding txnull-timer` value is set to 10 seconds.

**Functional Notes**

Specifies the length of time the answering endpoint attempts to detect the bonding negotiation pattern from the originating endpoint before deciding the bonding call has failed. It may be necessary to shorten this timer if the DTE equipment using the bonding module also has timer constraints for completing non-bonding parameter negotiation.

**Usage Examples**

The following example defines a `bonding txnull-timer` value of 8 seconds:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#bonding txnull-timer 8
```
**caller-id override [always <number> | if-no-cid <number>]**

Use the `caller-id override` command to configure the unit to replace caller ID information with a user-specified number. Use the `no` form of this command to disable any caller ID overrides.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>always &lt;number&gt;</code></td>
<td>Always forces replacement of the incoming caller ID number with the number given.</td>
</tr>
<tr>
<td><code>if-no-cid &lt;number&gt;</code></td>
<td>Replaces the incoming caller ID number with the number given only if there is no caller ID information available for the incoming call.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is disabled.

**Functional Notes**

Forces a replacement of the incoming caller ID number with the number given. The received caller ID, if any, is discarded, and the given override number is used to connect the incoming call to a circuit of the same number.

**Usage Examples**

The following example configures the unit to always provide the given number as the caller ID number:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#caller-id override always 5551000
```
**isdn [ldn 1 | ldn 2] <ldn>**

Use the `isdn ldn` command to specify the Local Directory Numbers (LDNs) for the Basic Rate ISDN (BRI) interface. This information should be supplied by your service provider. Use the `no` form of this command to remove a configured LDN.

**Note**  
The BRI module requires all incoming calls to be directed to the Local Directory Number (LDN) associated with the SPID programmed using the `isdn spid1` command. All calls to the LDN associated with SPID 2 will be rejected (unless part of a bonding call).

### Syntax Description

- **ldn 1**
  Specifies the LDN associated with the SPID entered as SPID 1.

- **ldn 2**
  Specifies the LDN associated with the SPID entered as SPID 2.

- **<ldn>**
  Specifies the LDN assigned to the circuit by the service provider. The LDN is the number used by remote callers to dial into the ISDN circuit.

### Default Values

By default, there are no configured LDNs.

### Functional Notes

Inbound calls are not accepted on interfaces without programmed LDNs. LDNs can also be entered using the `isdn spid` command. The `isdn spid` and `isdn ldn` commands overwrite the existing programmed LDN; therefore the last LDN programming entered takes precedence.

### Usage Examples

The following example defines an LDN of 555-1111:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#isdn ldn1 5551111
```
Use the `isdn` command to specify the Service Profile Identifiers (SPIDs) and the Local Directory Numbers (LDNs) for the Basic Rate ISDN (BRI) interface. This information should be supplied by your service provider. Use the `no` form of this command to remove a configured SPID.

**Syntax Description**

- `<spid string>` Specifies the 8- to 14-digit number identifying your BRI line in the Central Office switch. A SPID is generally created using the area code and phone number associated with the line and a four-digit suffix. For example, the following SPIDs may be provided on a BRI line with phone numbers 555-1111 and 555-1112:
  - SPID 1: 701 555 1111 0101
  - SPID 2: 701 555 1112 0101

- `<LDN>` Optional. Specifies the LDN assigned to the circuit by the service provider. An LDN programmed using the `isdn spid 1` command is automatically associated with SPID 1. An LDN programmed using the `isdn spid 2` command is automatically associated with SPID 2. The LDN is the number used by remote callers to dial into the ISDN circuit. Inbound calls are not accepted on interfaces without programmed LDNs. LDNs can also be entered using the `isdn ldn` command. The `isdn spid` and `isdn ldn` commands overwrite the existing programmed LDN; therefore the last LDN programming entered takes precedence.

**Default Values**

By default, there are no configured SPIDs or LDNs.

**Functional Notes**

The SROS does not support “SPID-less” 5ESS signaling. SPIDs are required for all configured BRI endpoints using 5ESS signaling.

For Euro applications, a SPID is not necessary. Use the `isdn ldn` command to configure the LDN for Euro applications.

**Usage Examples**

The following example defines a SPID of 704 555 1111 0101 and an LDN of 555 1111:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#isdn spid 1 70455511110101 5551111
```
**isdn switch-type [basic-5ess | basic-dms | basic-net3 | basic-ni]**

Use the `isdn switch-type` command to specify the ISDN signaling type configured on the Basic Rate ISDN (BRI) interface. The type of ISDN signaling implemented on the BRI interface does not always match the manufacturer of the Central Office switch. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic-5ess</td>
<td>Specifies Lucent/AT&amp;T 5ESS signaling.</td>
</tr>
<tr>
<td>basic-dms</td>
<td>Specifies Nortel DMS-100 custom signaling. The <code>basic-dms</code> signaling type is not compatible with proprietary SL-1 DMS signaling.</td>
</tr>
<tr>
<td>basic-net3</td>
<td>Specifies Net3 Euro-ISDN signaling.</td>
</tr>
<tr>
<td>basic-ni</td>
<td>Specifies National ISDN-1 signaling.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the ISDN signaling type is set to National ISDN-1.

**Functional Notes**

The `isdn switch-type` command specifies the type of ISDN signaling implemented on the BRI interface, not the manufacturer of the Central Office switch. It is quite possible to have a Lucent Central Office switch providing National ISDN signaling on the BRI interface.

**Usage Examples**

The following example configures a BRI interface for a circuit with Lucent 5ESS (custom) signaling:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#isdn switch-type basic-5ess
```
**loopback local [all | b1 | b2 | both]**

Use the `loopback local` command to enable a local loopback of the interface (towards the router). Use the `no` form of this command to disable the loopback.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>all</strong></td>
<td>Loops the entire interface back towards the router (including the D channel). With an active <code>loopback active all</code>, the established D channel between the ISDN module and the Central Office switch drops.</td>
</tr>
<tr>
<td><strong>b1</strong></td>
<td>Loops the data on B1 back towards the router. A B1 loopback does not disrupt D channel signaling.</td>
</tr>
<tr>
<td><strong>b2</strong></td>
<td>Loops the data on B2 back towards the router. A B2 loopback does not disrupt D channel signaling.</td>
</tr>
<tr>
<td><strong>both</strong></td>
<td>Loops the data on B1 and B2 back towards the router, but does not disrupt D channel signaling.</td>
</tr>
</tbody>
</table>

**Default Values**

No default necessary for this command.

**Usage Examples**

The following example enables a b2 loopback of the bri 1/2 interface and disables the loopback:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#loopback local b2
ProCurve(config-bri 1/2)#no loopback local b2
```
**loopback network [b1 | b2 | both]**

Use the `loopback network` command to enable a loopback of the interface (towards the network). Use the `no` form of this command to disable the loopback.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>b1</strong></td>
<td>Loops the data on B1 back towards the network. A B1 loopback does not disrupt D channel signaling.</td>
</tr>
<tr>
<td><strong>b2</strong></td>
<td>Loops the data on B2 back towards the network. A B2 loopback does not disrupt D channel signaling.</td>
</tr>
<tr>
<td><strong>both</strong></td>
<td>Loops the data on B1 and B2 back towards the network, but does not disrupt D channel signaling.</td>
</tr>
</tbody>
</table>

**Default Values**

No default necessary for this command.

**Usage Examples**

The following example enables a b2 loopback of the bri 1/2 interface and disables the loopback:

```
ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#loopback network b2
ProCurve(config-bri 1/2)#no loopback network b2
```
maintenance [reset | restart-d]

Use the maintenance command to force a reset of the interface (initiating the SABME/UA process) or to reset the D channel (by sending a RESTART message).

Caution  The maintenance command disrupts data flow on the ISDN interface. All active calls will drop when the reset or restart process begins.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reset</td>
<td>Forces a complete reset of the interface by initiating the SABME/UA process.</td>
</tr>
<tr>
<td>restart-d</td>
<td>Resets the D channel by sending a Q.931 RESTART message to the Central Office Switch.</td>
</tr>
</tbody>
</table>

Default Values

No default necessary for this command.

Usage Examples

The following example resets the bri 1/2 interface:

ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#maintenance reset
resource pool-member <pool-name> [<priority>]

Use the resource pool-member command to assign the interface to a resource pool, making it a demand routing resource. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;pool-name&gt;</th>
<th>Specifies the name of the resource pool to which this interface is assigned.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;priority&gt;</td>
<td>Optional. Specifies the priority of using this interface versus other interfaces contained in the specified resource pool using a number 1 to 255. Lower numbers indicate higher priority. Interfaces with the same priority are selected in alphabetical order by interface name.</td>
</tr>
</tbody>
</table>

Default Values

By default, BRI interfaces are not assigned to a resource pool.

Usage Examples

The following example configures a BRI interface as a member of resource pool MyPool:

ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#resource pool-member MyPool
test-call [dial <number> | speed <56 | 64> | answer | hangup | hangup channels <1 | 2>]

Use the test-call command to dial, answer, or disconnect a test call used for verifying the BRI interface connection. No data traffic is permitted on an ISDN interface in test mode.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dial &lt;number&gt;</td>
<td>Places the interface in test mode and makes an outbound call to the entered number. Enter the number to dial using numerical digits without spaces or special characters (such as hyphens or parentheses). Test calls made using the test-call dial command will perform channel negotiation, but no data passes through from end to end on the call. Remove the interface from test mode using the no test-call dial command.</td>
</tr>
<tr>
<td>speed &lt;56</td>
<td>64&gt;</td>
</tr>
<tr>
<td>answer</td>
<td>Places the interface in test answer mode and configures it to accept inbound calls. Using the test-call answer command supersedes any other interface configuration that may exist. Test calls answered by the interface while in test mode will perform channel negotiation, but no data passes through from end to end on the call. Remove the interface from test answer mode using the no test-call answer command.</td>
</tr>
<tr>
<td>hangup</td>
<td>Disconnects all active test calls.</td>
</tr>
<tr>
<td>hangup channels &lt;1</td>
<td>2&gt;</td>
</tr>
</tbody>
</table>

**Default Values**

By default, test call functionality is disabled.

**Usage Examples**

The following example places the bri 1/1 interface in test answer mode and then removes the interface from test answer mode:

ProCurve(config)#interface bri 1/2
ProCurve(config-bri 1/2)#test-call answer
ProCurve(config-bri 1/2)#no test-call answer
DSX-1 INTERFACE CONFIGURATION COMMAND SET

To activate the DSX-1 Interface Configuration mode, enter the `interface t1` command (and specify the DSX-1 port) at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `coding [ami | b8zs]` on page 541
- `framing [d4 | esf]` on page 542
- `line-length <value>` on page 543
- `loopback network [line | payload]` on page 544
- `loopback remote line inband` on page 545
- `remote-loopback` on page 546
- `signaling-mode [message-oriented | none | robbed-bit]` on page 547
- `snmp trap link-status` on page 548
- `test-pattern [ones | zeros]` on page 549
coding [ami | b8zs]

Use the `coding` command to configure the line coding for a DSX-1 physical interface. This setting must match the line coding supplied on the circuit by the external device.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ami</code></td>
<td>Configures the line coding for alternate mark inversion (AMI).</td>
</tr>
<tr>
<td><code>b8zs</code></td>
<td>Configures the line coding for bipolar eight zero substitution (B8ZS).</td>
</tr>
</tbody>
</table>

### Default Values

By default, all DSX-1 interfaces are configured with B8ZS line coding.

### Functional Notes

The line coding configured in the unit must match the line coding of the DSX-1 circuit. A mismatch will result in line errors (e.g., BPVs).

### Usage Examples

The following example configures the DSX-1 interface for AMI line coding:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#coding ami
```
framing [d4 | esf]

Use the `framing` command to configure the framing format for the DSX-1 interface. This parameter should match the framing format set on the external device. Use the `no` form of this command to return to the default value.

**Syntax Description**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>d4</td>
<td>Specifies D4 superframe (SF) format.</td>
</tr>
<tr>
<td>esf</td>
<td>Specifies extended superframe (ESF) format.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the framing format is set to `esf`.

**Functional Notes**

A frame is comprised of a single byte from each of the T1’s timeslots; there are 24 timeslots on a single T1 circuit. Framing bits are used to separate the frames and indicate the order of information arriving at the receiving equipment. D4 and ESF are two methods of collecting and organizing frames over the circuit.

**Usage Examples**

The following example configures the DSX-1 interface for D4 framing:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#framing d4
```
**line-length <value>**

Use the `line-length` command to set the line build out (in feet or dB) for the DSX-1 interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

<value>     Configures the line build out for the DSX-1 interface.

Valid options include: -7.5 dB or <0 to 655> feet.

**Default Values**

By default, the line build out is set to 0 feet.

**Functional Notes**

The `line-length` value represents the physical distance between DSX equipment (measured in cable length). Based on this setting, the SROS device increases signal strength to compensate for the distance the signal must travel. Valid distance ranges are listed below:

- 0 to 133 feet
- 134 to 265 feet
- 266 to 399 feet
- 400 to 533 feet
- 534 to 655 feet

Use the `-7.5 dB` setting for maximum signal attenuation.

**Usage Examples**

The following example configures the DSX-1 interface `line-length` for 300 feet:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#line-length 300
```
**loopback network [line | payload]**

Use the **loopback network** command to initiate a loopback on the interface toward the network. Use the **no** form of this command to deactivate the loopback.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>line</td>
<td>Initiates a metallic loopback of the physical DSX-1 network interface.</td>
</tr>
<tr>
<td>payload</td>
<td>Initiates a loopback of the T1 framer (CSU portion) of the DSX-1 network</td>
</tr>
<tr>
<td>interface</td>
<td>interface</td>
</tr>
</tbody>
</table>

**Default Values**

No default necessary for this command.

**Functional Notes**

The following diagram depicts the difference between a line and payload loopback.

![Diagram showing the difference between line and payload loopback]

**Usage Examples**

The following example initiates a payload loopback of the DSX-1 interface:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#loopback network payload
```
**loopback remote line inband**

Use the `loopback remote line inband` command to send a loopback code to the remote unit to initiate a line loopback. Use the `no` form of this command to send a loopdown code to the remote unit to deactivate the loopback.

**Syntax Description**

| inband       | Uses the inband channel to initiate a full 1.544 Mbps physical (metallic) loopback of the signal received by the remote unit from the network. |

**Default Values**

No defaults necessary for this command.

**Functional Notes**

A remote loopback can only be issued if a bind does not exist on the interface and if the signaling mode is set to `none`. The following diagram depicts the difference between a line and payload loopback.

---

**Usage Examples**

The following example initiates a remote line loopback using the inband channel:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#loopback remote line inband
```
remote-loopback

Use the remote-loopback command to configure the interface to respond to loopbacks initiated by a remote unit (or the service provider). Use the no form of this command to disable this feature.

Syntax Description
No subcommands.

Default Values
By default, all interfaces respond to remote loopbacks.

Usage Examples
The following example enables remote loopbacks on the DSX-1 interface:

ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#remote-loopback
signaling-mode [message-oriented | none | robbed-bit]

Use the `signaling-mode` command to configure the signaling type (robbed bit for voice or clear channel for data) for the DS0s mapped to the DSX-1 port.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-oriented</td>
<td>Specifies clear channel signaling on Channel 24 only. Use this signaling type with QSIG installations.</td>
</tr>
<tr>
<td>none</td>
<td>Specifies clear channel signaling on all 24 DS0s. Use this signaling type with data-only or PRI DSX-1 installations.</td>
</tr>
<tr>
<td>robbed-bit</td>
<td>Specifies robbed bit signaling on all DS0s. Use this signaling type for voice-only DSX-1 applications.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the signaling mode is set to robbed-bit.

**Usage Examples**

The following example configures the DSX-1 port for PRI compatibility:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#signaling-mode none
```
**snmp trap link-status**

Use the **snmp trap link-status** to control the Simple Network Management Protocol (SNMP) variable *ifLinkUpDownTrapEnable* (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the *ifLinkUpDownTrapEnable* OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The **snmp trap link-status** command is used to control the RFC2863 *ifLinkUpDownTrapEnable* OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the DSX-1 interface:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#no snmp trap link-status
```
test-pattern [ones | zeros]

Use the test-pattern command to activate the built-in pattern generator and begin sending the specified test pattern. This pattern generation can be used to verify a data path when used in conjunction with an active loopback. Use the no form of this command to cease pattern generation.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ones</td>
<td>Generates a test pattern of continous ones.</td>
</tr>
<tr>
<td>zeros</td>
<td>Generates a test pattern of continous zeros.</td>
</tr>
</tbody>
</table>

Default Values

No defaults necessary for this command.

Usage Examples

The following example activates the pattern generator for a stream of continuous ones:

```
ProCurve(config)#interface t1 1/2
ProCurve(config-t1 1/2)#test-pattern ones
```
E1 INTERFACE CONFIGURATION COMMAND SET

To activate the E1 Interface Configuration mode, enter the `interface e1` command (and specify the E1 port) at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `clock source [internal | line | through]` on page 551
- `coding [ami | hdb3]` on page 552
- `framing [crc4]` on page 553
- `loop-alarm-detect` on page 554
- `loopback network [line | payload]` on page 555
- `loopback remote v54` on page 556
- `remote-alarm [rai | ais]` on page 557
- `remote-loopback` on page 558
- `sa4tx-bit [0 | 1]` on page 559
- `snmp trap line-status` on page 560
- `snmp trap link-status` on page 561
- `tdm-group <group number> timeslots <1-31> speed [56 | 64]` on page 562
- `test-pattern [clear | errors | insert | ones | p215 | p220 | p511 | qrss | zeros]` on page 563
- `ts16` on page 564
**clock source [internal | line | through]**

Use the `clock source` command to configure the source timing used for the interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>internal</code></td>
<td>Configures the unit to provide clocking using the internal oscillator.</td>
</tr>
<tr>
<td><code>line</code></td>
<td>Configures the unit to recover clocking from the E1 circuit.</td>
</tr>
<tr>
<td><code>through</code></td>
<td>Configures the unit to recover clocking from the circuit connected to the G.703 interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the unit is configured to recover clocking from the primary circuit.

**Functional Notes**

When operating on a circuit that is providing timing, setting the `clock source` to `line` can avoid errors such as Clock Slip Seconds (CSS).

**Usage Examples**

The following example configures the unit to recover clocking from the primary circuit:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#clock source line
```
coding [ami | hdb3]

Use the `coding` command to configure the line coding for the E1 physical interface. This setting must match the line coding supplied on the circuit by the service provider.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ami</td>
<td>Configures the line coding for alternate mark inversion (AMI).</td>
</tr>
<tr>
<td>hdb3</td>
<td>Configures the line coding for high-density bipolar 3 (HDB3).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all E1 interfaces are configured with HDB3 line coding.

**Functional Notes**

The line coding configured in the unit must match the line coding of the E1 circuit. A mismatch will result in line errors (e.g., BPVs).

**Usage Examples**

The following example configures the E1 interface for AMI line coding:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#coding ami
```
framing [crc4]

Use the `framing` command to configure the framing format for the E1 interface. This parameter should match the framing format set on the external device. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>syntax</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crc4</td>
<td>Enables CRC-4 bits to be transmitted in the outgoing data stream. Also, the received signal is checked for CRC-4 errors.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, CRC-4 framing is disabled.

**Functional Notes**

The framing value must match the configuration of the E1 circuit. A mismatch will result in a loss of frame alarm.

**Usage Examples**

The following example configures the E1 interface for CRC-4 framing:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#framing crc4
```
loop-alarm-detect

The `loop-alarm-detect` command enables detection of a loop alarm on the E1 interface. Use the `no` form of this command to disable this feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is enabled.

**Functional Notes**

This command enables the detection of a loopback alarm. This alarm works in conjunction with the `sa4tx-bit` command setting. The loopback condition is detected by comparing the transmitted `sa4tx-bit` value to the received Sa4 bit value. If the bits match, a loopback is assumed. This detection method only works with a network in which the far end is transmitting the opposite value for Sa4.

**Usage Examples**

The following example enables detection of a loop alarm on the E1 interface:

```
ProCurve(config)#config e1 1/1
ProCurve(config-e1 1/1)#loop-alarm-detect
```
loopback network [line | payload]

Use the loopback network command to initiate a loopback on the interface toward the network. Use the no form of this command to deactivate the loopback.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>line</td>
<td>Initiates a metallic loopback of the physical E1 network interface.</td>
</tr>
<tr>
<td>payload</td>
<td>Initiates a loopback of the E1 framer (CSU portion) of the E1 network interface.</td>
</tr>
</tbody>
</table>

Default Values

No default necessary for this command.

Functional Notes

The following diagram depicts a line loopback.

![Diagram of E1 Network Interface]

Usage Examples

The following example initiates a line loopback of the E1 interface:

ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#loopback network line
loopback remote v54

The `loopback remote v54` command initiates an E1 remote loopback test (with a V.54 loopback pattern). Use the `no` form of this command to deactivate the loopback.

**Syntax Description**
No subcommands.

**Default Values**
No default value is necessary for this command.

**Functional Notes**
This command causes a V.54 inband loop code to be sent in the payload towards the far end.

**Usage Examples**
The following example sends a V.54 inband loop code to the far end:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#loopback remote v54
```
remote-alarm [rai | ais]

The `remote-alarm` command selects the alarm signaling type to be sent when a loss of frame is detected on the E1 receive signal. Use the `no` form of this command to disable all transmitted alarms.

**Syntax Description**

<table>
<thead>
<tr>
<th>rai</th>
<th>Specifies sending a remote alarm indication (RAI) in response to a loss of frame. Also prevents received RAI from causing a change in interface operational status.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ais</td>
<td>Specifies sending an alarm indication signal (AIS) as an unframed all-ones signal.</td>
</tr>
</tbody>
</table>

**Default Values**

The default for this command is `rai`.

**Functional Notes**

An E1 will respond to a loss of frame on the receive signal by transmitting a remote alarm to the far end to indicate the error condition. TS0 of an E1 contains the Frame Alignment Signal (FAS) in the even-numbered frames. The odd-numbered frames are not used for frame alignment, and some of those bits are labeled as spare bits (SA bits) in bit positions 4 through 8.

**Usage Examples**

The following example enables transmission of AIS in response to a loss of frame:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#remote alarm ais
```
remote-loopback

Use the remote-loopback command to configure the interface to accept loopback requests initiated by a remote unit (or the service provider). Use the no form of this command to disable this feature.

Syntax Description

No subcommands.

Default Values

By default, all interfaces respond to remote loopbacks.

Functional Notes

This controls the acceptance of any remote loopback requests. When enabled, remote loopbacks are detected and cause a loopback to be applied. When disabled, remote loopbacks are ignored.

Usage Examples

The following example enables remote loopbacks on the E1 interface:

ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#remote-loopback
**sa4tx-bit [0 | 1]**

The `sa4tx-bit` command selects the Tx value of Sa4 in this E1 interface. Use the `no` form of this command to return to the default value of 1.

**Syntax Description**

No subcommands.

**Default Values**

The default value for this command is 1.

**Functional Notes**

This command assigns a value to the Tx spare bit in position 4. The odd-numbered frames of TS0 are not used for frame alignment. Bits in position 4 through 8 are called spare bits. Values of 0 or 1 are accepted.

**TS0 odd frame**

<table>
<thead>
<tr>
<th>Bit position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit use</td>
<td>0</td>
<td>1</td>
<td>RAI = 1</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example sets the Tx value of Sa4 to 0:

ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#sa4tx-bit 0
**snmp trap line-status**

Use the `snmp trap line-status` command to control the Simple Network Management Protocol (SNMP) variable `dsx1LineStatusChangeTrapEnable` (RFC2495) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the `no` form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the `dsx1LineStatusChangeTrapEnable` OID is set to enabled for all interfaces except virtual Frame Relay Interfaces.

**Functional Notes**

The `snmp trap line-status` command is used to control the RFC2495 `dsx1LineStatusChangeTrapEnable` OID (OID number 1.3.6.1.2.1.10.18.6.1.17.0).

**Usage Examples**

The following example disables the line-status trap on the T1 interface:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-t1 1/1)#no snmp trap line-status
```
snmp trap link-status

Use the **snmp trap link-status** to control the Simple Network Management Protocol (SNMP) variable `ifLinkUpDownTrapEnable` (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the `ifLinkUpDownTrapEnable` OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The **snmp trap link-status** command is used to control the RFC2863 `ifLinkUpDownTrapEnable` OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the E1 interface:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#no snmp trap link-status
```
tdm-group <group number> timeslots <1-31> speed [56 | 64]

Use the tdm-group command to create a group of contiguous channels on this interface to be used during the bind process. See bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port> on page 1304 for related information.

**Caution** Changing tdm-group settings could result in service interruption.

**Syntax Description**

- `<group number>` Identifies the created TDM group by numerical label (Range: 1 to 255).
- `timeslots <1-31>` Specifies the channels to be used in the TDM group. This can be entered as a single number representing one of the 31 E1 channel timeslots or as a contiguous group of channels. (For example, 1-10 specifies the first 10 channels of the E1.)
- `speed [56 | 64]` Optional. Specifies the individual channel rate on the E1 interface to be 56 or 64 kbps. The default speed is 64 kbps. 56 kbps operation is not available on all E1 interfaces. Refer to the Quick Start Guide provided with your E1 module to determine whether 56 kbps is valid.

**Default Values**

By default, there are no configured TDM groups.

**Usage Examples**

The following example creates a TDM group (labeled 5) of 10 channels at 64 kbps each:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#tdm-group 5 timeslots 1-10 speed 64
```
test-pattern [clear | errors | insert | ones | p215 | p220 | p511 | qrss | zeros]

Use the test-pattern command to activate the built-in pattern generator and begin sending the specified test pattern. This pattern generation can be used to verify a data path when used in conjunction with an active loopback. Use the no form of this command to cease pattern generation.

Syntax Description

clear  Clears the test pattern error count. Display the error count using the errors keyword.
errors  Displays the test pattern error count.
insert  Inserts an error into the currently active test pattern. Display the error count using the errors keyword.
ones   Generates a test pattern of continuous ones.
p215   Generates a pseudorandom test pattern based on a 15-bit shift register.
p220   Generates a pseudorandom test pattern based on a 20-bit shift register.
p511   Generates a test pattern of repeating ones and zeros.
qrss   Generates a test pattern of random ones and zeros.
zeros  Generates a test pattern of continuous zeros.

Default Values

No defaults necessary for this command.

Usage Examples

The following example activates the pattern generator for a stream of continuous ones:

ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#test-pattern ones
**ts16**

Use the `ts16` command to enable timeslot 16 multiframe to be checked on the receive signal. Use the `no` form of this command to disable timeslot 16.

**Syntax Description**

No subcommands.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

If timeslot 16 is used on the incoming E1, do not map timeslot 16 using the `tdm-group` command. By default, all timeslots not physically mapped using the `tdm-group` command are passed through to the G.703 interface. Leaving timeslot 16 unmapped makes it available for multiframe signaling by the connected E1 device.

**Usage Examples**

The following example enables timeslot 16 multiframing:

```
ProCurve(config)#interface e1 1/1
ProCurve(config-e1 1/1)#ts16
```
ETHERNET INTERFACE CONFIGURATION COMMAND SET

There are several types of Ethernet interfaces associated with the SROS:

- Basic Ethernet interfaces (e.g., eth 0/1)
- Ethernet sub-interfaces associated with a VLAN (e.g., eth 0/1.1)
- Ethernet interface range (e.g., eth 0/1, 0/2)

To activate the basic Ethernet Interface Configuration mode, enter the `interface ethernet` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#
```

To activate the Ethernet Sub-Interface Configuration mode, enter the `interface ethernet` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface ethernet 0/1.1
ProCurve(config-eth 0/1.1)#
```

To activate the Ethernet Configuration mode for a range of interfaces, enter the `interface range` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface range ethernet 0/1, 0/2
ProCurve(config-eth 0/1, 0/2)#
```
The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- **alias <text>** on page 1303
- **bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>** on page 1304
- **description <text>** on page 1307
- **do** on page 1308
- **end** on page 1309
- **exit** on page 1310
- **ping <address>** on page 1311
- **shutdown** on page 1315

All other commands for this command set are described in this section in alphabetical order.

- **access-policy <policyname>** on page 568
- **arp arpa** on page 569
- **bandwidth <value>** on page 570
- **bridge-group <group#>** on page 571
- **crypto map <mapname>** on page 573
- **dynamic-dns** on page 575
- **encapsulation 802.1q** on page 577
- **full-duplex** on page 578
- **half-duplex** on page 579
ip commands begin on page 580
lldp receive on page 608
lldp send [management-address | port-description | system-capabilities | system-description | system-name] on page 609
lldp send-and-receive on page 610
mac-address <address> on page 611
max-reserved-bandwidth <percent> on page 612
mtu <size> on page 613
port-auth supplicant [username <username> | password <password>] on page 614
qos-policy out <mapname> on page 615
snmp trap on page 616
snmp trap link-status on page 617
spanning-tree commands begin on page 618
speed [10 | 100 | auto] on page 623
traffic-shape rate <rate> <burstrate> on page 624
vlan-id <vlan id> [native] on page 625
access-policy <policyname>

Use the access-policy command to assign a specified access policy for the inbound traffic to an interface. Use the no form of this command to remove an access policy association.

**Note** Configured access policies will only be active if the ip firewall command has been entered at the Global Configuration Mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.

**Syntax Description**

| <policyname> | Identifies the configured access policy by alphanumeric descriptor (all access policy descriptors are case-sensitive). |

**Default Values**

By default, there are no configured access policies associated with an interface.

**Functional Notes**

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter access policy <policy name>. For more details on creating and using access policies, refer to ip policy-class <policynname> on page 400.

**Usage Examples**

The following example associates the access policy UnTrusted (to allow inbound traffic to the Web server) to the Ethernet 0/1 interface:

Enable the SROS security features:
ProCurve(config)#ip firewall

Create the access list (this is the packet selector):
ProCurve(config)#ip access-list extended InWeb
ProCurve(config-ext-nacl)#permit tcp any host 10.12.5.253 eq 80

Create the access policy that contains the access list InWeb:
ProCurve(config)#ip policy-class UnTrusted
ProCurve(config-poly-class)#allow list InWeb

Associate the access policy with the Ethernet 0/1 interface:
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#access-policy UnTrusted
arp arpa

Use the **arp arpa** command to enable address resolution protocol (ARP) on the Ethernet interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arpa</td>
<td>Sets standard address resolution protocol for this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

The default for this command is arpa.

**Usage Examples**

The following example enables standard ARP for the Ethernet interface:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#arp arpa
```
bandwidth <value>

Use the bandwidth command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the no form of this command to restore the default value.

**Syntax Description**

<value> Specifies bandwidth in kbps.

**Default Values**

To view default values, use the show interfaces command.

**Functional Notes**

The bandwidth command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the Ethernet 0/1 interface to 10 Mbps:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)# bandwidth 10000
```
**bridge-group <group#>**

Use the **bridge-group** command to assign an interface to the specified bridge group. Use the **no** form of this command to remove the interface from the bridge group.

**Syntax Description**

| <group#> | Specifies the bridge group number (1 to 255). |

**Default Values**

By default, there are no configured bridge groups.

**Functional Notes**

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (e.g., Ethernet to T1 bridge, Ethernet to Frame Relay sub-interface).

**Usage Examples**

The following example assigns the Ethernet interface to bridge-group 17:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#bridge-group 17
```
**bridge-group <group#> vlan-transparent**

Use the **bridge-group vlan-transparent** command to prevent an interface from removing the VLAN tag. Use the **no** form of this command to allow the interface to remove the VLAN tag from the packet.

**Note** The **bridge-group vlan-transparent** command is not a global command. The command must be applied on all interfaces of the bridge group.

### Syntax Description

<group#> Specifies the bridge group number. Valid range is 1 to 255.

### Default Values

By default, VLAN tags are removed from the data.

### Usage Examples

The following example removes the VLAN tags from the packets on the Ethernet interface 0/1:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#bridge-group 1 vlan-transparent
```
crypto map <mapname>

Use the crypto map command to associate crypto maps with the interface.

**Syntax Description**

<mapname> Specifies the crypto map name that you wish to assign to the interface.

**Default Values**

By default, no crypto maps are assigned to an interface.

**Functional Notes**

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the unencrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the diagram above, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access-group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only unencrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, unencrypted source of the data. The destination information will be the far-end, unencrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy-class is the far-end. The destination information is the local-side.

Usage Examples

The following example applies all crypto maps with the name MyMap to the Ethernet interface:

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#crypto map MyMap
dynamic-dns

Use the `dynamic-dns` command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the `no` versions of these commands to disable these features. Variations of this command include:

- `dynamic-dns custom <hostname> <minutes>`
- `dynamic-dns dyndns <hostname> <username> <password>`
- `dynamic-dns dyndns-custom <hostname> <username> <password>`
- `dynamic-dns dyndns-static <hostname> <username> <password>`

**Syntax Description**

- `<hostname>` Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).
- `<minutes>` Specifies the intervals in minutes to update the server with information (updates also occur when the interface's IP address changes regardless of the update intervals).
  
Referring to Functional Notes below for additional argument descriptions.

**Default Values**

No default is necessary for this command.

**Functional Notes**

- `custom` - Constanttime.com’s Custom Dynamic DNS service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows to manage IP address mappings (A records), domain aliases (CNAME records) and mail servers (MX records).

- `dyndns` - The Dynamic DNS service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- `dyndns-custom` - DynDNS.org’s Custom DNS service provides a full DNS solution, giving you complete control over an entire domain name. A web-based interface provides two levels of control over your domain, catering to average or power users. Five globally redundant DNS servers ensure that your domain will always resolve.

A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background, and provides layout and functionality similar to a BIND zone file allowing for the creation of nearly any record type.

Custom DNS can be used with both static and dynamic IPs, and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
dyndns-static - The Static DNS service is similar to Dynamic DNS service, in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate though the DNS system. This service is provided for up to five hostnames.

If your IP address doesn't change often or at all, but you still want an easy name to remember it by (without having to purchase your own domain name) Static DNS service is ideal for you.

If you would like to use your own domain name (such as yourname.com) you need Custom DNS service which also provides full dynamic and static IP address support.

Usage Examples

The following example sets the dynamic-dns to dyndns-custom with hostname host, username user, and password pass:

ProCurve(config-eth 0/1)#dynamic-dns dyndns-custom host user pass
encapsulation 802.1q

Use the encapsulation 802.1q command to put the interface into 802.1q (VLAN) mode.

Syntax Description

No subcommands.

Default Values

No default value is necessary for this command.

Usage Examples

The following example puts interface eth 0/1 in 802.1q mode and configures a sub-interface for vlan usage:

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#encapsulation 802.1q
ProCurve(config-eth 0/1)#interface ethernet 0/1.1
ProCurve(config-eth 0/1.1)vlan-id 3
full-duplex

Use the `full-duplex` command to configure the Ethernet interface for full-duplex operation. This allows the interface to send and receive simultaneously. Use the `no` form of this command to return to the default `half-duplex` operation.

Syntax Description

No subcommands.

Default Values

By default, all Ethernet interfaces are configured for half-duplex operation.

Functional Notes

Full-duplex Ethernet is a variety of Ethernet technology currently being standardized by the IEEE. Because there is no official standard, vendors are free to implement their independent versions of full-duplex operation. Therefore, it is not safe to assume that one vendor's equipment will work with another.

Devices at each end of a full-duplex link have the ability to send and receive data simultaneously over the link. Theoretically, this simultaneous action can provide twice the bandwidth of normal (half-duplex) Ethernet. To deploy full-duplex Ethernet, each end of the link must only connect to a single device (a workstation or a switched hub port). With only two devices on a full-duplex link, there is no need to use the medium access control mechanism (to share the signal channel with multiple stations) and listen for other transmissions or collisions before sending data.

`Note`

Some Ethernet equipment (though rare) is unable to negotiate duplex if speed is manually determined. To avoid incompatibilities, manually set the duplex if the speed is manually set. See `speed [10 | 100 | auto]` on page 623 for more information.

The 10BaseT, 100BaseTX, and 100BaseFX signalling systems support full-duplex operation (because they have transmit and receive signal paths that can be simultaneously active).

Usage Examples

The following example configures the Ethernet interface for `full-duplex` operation:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#full-duplex
```
**half-duplex**

Use the `half-duplex` command to configure the Ethernet interface for half-duplex operation. This setting allows the Ethernet interface to either send or receive at any given moment, but not simultaneously. Use the `no` form of this command to disable half-duplex operation.

**Syntax Description**

No subcommands.

**Default Values**

By default, all Ethernet interfaces are configured for half-duplex operation.

**Functional Notes**

Half-duplex Ethernet is the traditional form of Ethernet that employs the Carrier Sense Multiple Access/Collision Detect (CSMA/CD) protocol to allow two or more hosts to share a common transmission medium while providing mechanisms to avoid collisions. A host on a half-duplex link must "listen" on the link and only transmit when there is an idle period. Packets transmitted on the link are broadcast (so it will be "heard" by all hosts on the network). In the event of a collision (two hosts transmitting at once), a message is sent to inform all hosts of the collision and a backoff algorithm is implemented. The backoff algorithm requires the station to remain silent for a random period of time before attempting another transmission. This sequence is repeated until a successful data transmission occurs.

**Note**  
Some Ethernet equipment (though rare) is unable to negotiate duplex if speed is manually determined. To avoid incompatibilities, manually set the duplex if the speed is manually set. See `speed [10 | 100 | auto]` on page 623 for more information.

**Usage Examples**

The following example configures the Ethernet interface for `half-duplex` operation:

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#half-duplex
**ip access-group <listname> [in | out]**

Use the `ip access-group` command to create an access list to be used for packets transmitted on or received from the specified interface. Use the `no` form of this command to disable this type of control.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;listname&gt;</code></td>
<td>Specifies the assigned IP access list name.</td>
</tr>
<tr>
<td>in</td>
<td>Enables access control on packets received on the specified interface.</td>
</tr>
<tr>
<td>out</td>
<td>Enables access control on packets transmitted on the specified interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, these commands are disabled.

**Functional Notes**

When this command is enabled, the IP destination address of each packet must be validated before being passed through. If the packet is not acceptable per these settings, it is dropped.

**Usage Examples**

The following example sets up the router to only allow Telnet traffic (as configured in the user-configured `TelnetOnly` IP access list) into the Ethernet interface:

```
ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#interface eth 0/1
ProCurve(config-eth 0/1)#ip access-group TelnetOnly in
```
ip address dhcp

Use the `ip address dhcp` command to use Dynamic Host Configuration Protocol (DHCP) to obtain an address on the Ethernet interface. Use the `no` form of this command to remove a configured IP address (using DHCP) and disable DHCP operation on the interface.

```
ip address dhcp [client-id [<interface> | <identifier>] hostname <"string"> ]
```

**Syntax Description**

- **client-id**
  - Optional. Specifies the client identifier used when obtaining an IP address from a DHCP server.
  - `<interface>`
    - Specifying an interface defines the client identifier as the hexadecimal MAC address of the specified interface (including a hexadecimal number added to the front of the MAC address to identify the media type).
    - For example, specifying the `client-id ethernet 0/1` (where the Ethernet interface has a MAC address of 0012.7991.1150) defines the client identifier as `01:00:12:79:91:11:50` where 01 defines the media type as Ethernet. Refer to `hardware-address <hardware-address> <type>` on page 1287 for a detailed listing of media types.

- `<identifier>`
  - Specifies a custom client-identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters).
  - For example, a custom client identifier of `0f:ff:ff:ff:ff:51:04:99:a1` may be entered using the `<identifier>` option.

- **hostname**
  - Specifies a text string (to override the global router name) to use as the name in the DHCP option 12 field.
  - `<string>`
    - String (encased in quotation marks) of up to 35 characters to use as the name of the host for DHCP operation.

- **no-default-route**
  - Keyword used to specify that the SROS not install the default-route obtained via DHCP.

- **no-domain-name**
  - Keyword used to specify that the SROS not install the domain-name obtained via DHCP.

- **no-nameservers**
  - Keyword used to specify that the SROS not install the DNS servers obtained via DHCP.

**Default Values**

- **client-id**
  - Optional. By default, the client identifier is populated using the following formula:
    ```
    TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS
    ```
  - Where `TYPE` specifies the media type in the form of one hexadecimal byte (refer to `hardware-address <hardware-address> <type>` on page 1287 for a detailed listing of media types) and the `MAC ADDRESS` is the Media Access Control (MAC) address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to Ethernet 0/1 is used in this field).

- **hostname <"string">**
  - By default, the hostname is the name configured using the Global Configuration
**hostname** command.

**Functional Notes**
Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the **client-id** and **hostname** fields.

**Usage Examples**
The following example enables DHCP operation on Ethernet interface 0/1:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip address dhcp
```
ip address <address> <mask> [secondary]

Use the ip address command to define an IP address on the specified interface (only one primary address is allowed). Use the optional secondary keyword to define a secondary IP address. Use the no form of this command to remove a configured IP address.

Syntax Description

- `<address>` Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).
- `<mask>` Specifies the subnet mask that corresponds to the listed IP address.
- `secondary` Optional. Configures secondary IP addresses for the specified interface. Multiple secondary IP addresses may be assigned (no limit).

Default Values

By default, there are no assigned IP addresses.

Functional Notes

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

Usage Examples

The following example configures a secondary IP address of 192.168.72.101 /30:

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip address 192.168.72.101 255.255.255.252 secondary

To view configured secondary IP address(es), use the show running-config command. Secondary IP address information is not available through other show commands.
ip dhcp release

Use the **ip dhcp release** command to transmit a message to the DHCP server requesting termination of the IP address lease on that interface.

---

**Caution**

If you are currently connected to the unit using a Telnet session through the Ethernet interface, using the **ip dhcp release** command will terminate your Telnet session and render your Telnet capability inoperable until a new IP address is assigned by the DHCP server.

---

**Syntax Description**

No subcommands.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically-assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain.

**Usage Examples**

The following example releases the IP address assigned (by DHCP) on the Ethernet interface (eth 0/1):

```
ProCurve(config)#int eth 0/1
ProCurve(config-eth 0/1)#ip dhcp release
```
ip dhcp renew

Use the `ip dhcp renew` command to transmit a message to the DHCP server requesting renewal of the IP address lease on that interface.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain.

**Usage Examples**

The following example renews the IP address assigned (by DHCP) on the Ethernet interface (eth 0/1):

```
ProCurve(config)#int eth 0/1
ProCurve(config-eth 0/1)#ip dhcp renew
```
**ip helper-address <address>**

Use the `ip helper-address` command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the `no` form of this command to disable forwarding packets.

**Syntax Description**

`<address>` Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

**Default Values**

By default, broadcast UDP packets are not forwarded.

**Functional Notes**

When used in conjunction with the `ip forward-protocol` command, the `ip helper-address` feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign helper address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the `ip forward-protocol` command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the `ip helper-address` feature:

1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the `ip forward-protocol` command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

**Usage Examples**

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.33.5.99:

```plaintext
ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip helper-address 192.168.5.99
```
**ip igmp**

Use the `ip igmp` command to configure multicasting-related functions for the interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>immediate-leave</td>
<td>Specifies that if only one host (or IGMP snooping switch) is connected to the interface, when a leave is received, multicast of that group is immediately terminated as opposed to sending a group query and timing out the group if no device responds. Works in conjunction with <code>ip igmp last-member-query-interval</code>. Applies to all groups when configured.</td>
</tr>
<tr>
<td>last-member-query-interval</td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group's multicast packets to the interface. Range: 100 to 65535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td>querier-timeout</td>
<td>Specifies the number of seconds that the router waits after the current querier’s last query before it takes over as querier (IGMP V2). Range: 60-300 seconds. Default: 2x the <code>query-interval</code> value.</td>
</tr>
<tr>
<td>query-interval</td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td>query-max-response-time</td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td>static-group</td>
<td>Configures the router’s interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>version [1</td>
<td>2]</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example sets the query message interval on the interface to 200 milliseconds:

```
ProCurve(config-eth 0/1)#ip igmp last-member-query-interval 200
```
**ip mcast-stub downstream**

Use the `ip mcast-stub downstream` command to enable multicast forwarding and IGMP (router mode) on an interface, and to place it in multicast stub downstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub upstream` commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. See `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub upstream` on page 591 for more information.

**Usage Examples**

The following example enables multicast forwarding and IGMP on the interface:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip mcast-stub downstream
```
**ip mcast-stub fixed**

Use the `ip mcast-stub fixed` command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the `no` form of this command to disable this mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Multicast routing must be enabled prior to setting `ip mcast-stub fixed` on the selected interface. Also, use the `ip igmp static-group <A.B.C.D>` command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

**Usage Examples**

The following example enables multicast traffic forwarding and IGMP on the interface:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip mcast-stub fixed
```
**ip mcast-stub helper-address** *<ip address>*

Use the `ip mcast-stub helper-address` command to specify an IP address toward which IGMP host reports and leave messages are forwarded. This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub downstream` and `ip mcast-stub upstream` commands. Use the `no` form of this command to return to default.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Specifies the address to which the IGMP host reports and leave messages are forwarded.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no helper-address is configured.

**Functional Notes**

The helper address is configured globally and applies to all multicast-stub downstream interfaces. The address specified may be the next upstream hop or any upstream address on the distribution tree for the multicast source, up to and including the multicast source. The router selects, from the list of multicast-stub upstream interfaces, the interface on the shortest path to the specified address. The router then proxies, on the selected upstream interface (using an IGMP host function), any host joins/leaves received on the downstream interface(s). The router retransmits these reports with addresses set as if the report originated from the selected upstream interface.

For example, if the router receives multiple joins for a group, it will not send any extra joins out the upstream interface. Also, if it receives a leave, it will not send a leave until it is certain that there are no more subscribers on any downstream interface.

**Usage Examples**

The following example specifies 172.45.6.99 as the helper address:

```
ProCurve(config)#ip mcast-stub helper-address 172.45.6.99
```
**ip mcast-stub upstream**

Use the `ip mcast-stub upstream` command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub downstream` commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router’s unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface. Refer to `ip mcast-stub downstream` on page 588 for more information.

**Usage Examples**

The following example enables multicast forwarding on the interface:

```
ProCurve(config-eth 0/1)#ip mcast-stub upstream
```
ip ospf

Use the `ip ospf` command to customize OSPF settings (if needed).

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication-key</td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td>cost <code>&lt;value&gt;</code></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32,767.</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td>message-digest-key <code>&lt;keyid&gt;</code> md5 <code>&lt;key&gt;</code></td>
<td>Configures OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td>priority <code>&lt;value&gt;</code></td>
<td>Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32,767.</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32,767.</td>
</tr>
</tbody>
</table>

**Default Values**

- retransmit-interval `<seconds>`: 5 seconds
- transmit-delay `<seconds>`: 1 second
- hello-interval `<seconds>`: 10 seconds
- dead-interval `<seconds>`: 40 seconds

**Usage Example**

The following example sets the maximum number of seconds allowed between hello packets to 25,000:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip ospf dead-interval 25000
```
ip ospf authentication [message-digest | null]

Use the `ip ospf authentication` command to authenticate an interface that is performing OSPF authentication.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-digest</td>
<td>Optional. Selects message-digest authentication type.</td>
</tr>
<tr>
<td>null</td>
<td>Optional. Specifies that no authentication be used.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this is set to null (meaning no authentication is used).

**Usage Examples**

The following example specifies that no authentication will be used on the Ethernet interface:

```
ProCurve(config-eth 0/1)#ip ospf authentication null
```
ip ospf network [broadcast | point-to-point]

Use the **ip ospf network** command to specify the type of network on this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcast</td>
<td>Sets the network type for broadcast.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Sets the network type for point-to-point.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

**Functional Notes**

A point-to-point network will not elect designated routers.

**Usage Examples**

The following example designates a broadcast network type:

```
ProCurve(config-eth 0/1)#ip ospf network broadcast
```
**ip pim sparse-mode**

Use the `ip pim sparse-mode` command to enable PIM Sparse Mode on the interface. Use the `no` form of this command to disable PIM Sparse Mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, PIM Sparse Mode is disabled for all interfaces.

**Functional Notes**

PIM Sparse Mode is a multicast routing protocol that makes use of the unicast forwarding table. PIM-systems builds unidirectional shared trees rooted at a Rendezvous Point (RP) for a multicast group, or a shortest-path tree rooted at a specific source for a multicast group.

**Usage Examples**

The following example enables PIM Sparse Mode on the eth 0/1:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip pim sparse-mode
```
ip pim-sparse dr-priority <value>

Use the `ip pim-sparse dr-priority` command to specify the priority of this PIM interface for use when selecting the designated router (DR). Use the `no` form of this command to return to the default value.

**Syntax Description**

| <value> | Specifies the priority of this interface (to be used when determining the DR). Valid range is 1 to 4,294,967,295. |

**Default Values**

By default, the priority of all PIM interfaces is 1.

**Functional Notes**

Interfaces advertise their configured priority values in the hello messages transmitted on the interface. Routers use the priority values to determine the appropriate DR. The router on the network segment with the highest priority is selected as the DR. If a hello message is received on the interface from a router on the network segment and it does not contain a priority, the entire network segment defaults to DR selection based on IP addresses instead of priority. In this instance, the DR is selected as the router on the network segment that has the highest IP address. ProCurve Secure Routers will always include a priority in all transmitted hello messages. If no priority is specifically designated by the user, the priority is set as the default of 1.

**Usage Examples**

The following example specifies a priority of 100 on the eth 0/1 interface:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip pim-sparse dr-priority 100
```
**ip pim-sparse hello-timer <time>**

Use the `ip pim-sparse hello-timer` command to set the time interval at which periodic hello messages are transmitted out the interface. Each PIM interface has an independent hello-timer. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the interval (in seconds) at which periodic hellos are sent out the interface. Valid range is 10 to 3600 seconds. |

**Default Values**

By default, hellos are transmitted on PIM interfaces every 60 seconds.

**Functional Notes**

Hello messages are used to inform neighbors of a router's presence. Hello messages normally generate a small amount of traffic on an interface. Setting the `hello-timer` to a small interval increases the number of hellos sent (thus increasing the amount of traffic). Set the `hello-timer` to a reasonable value, taking into consideration the bandwidth available on the interface.

**Usage Examples**

The following example specifies hellos to be sent on the eth 0/1 interface every 3600 seconds:

ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip pim-sparse hello-timer 3600
**ip pim-sparse nbr-timeout <time>**

Use the *ip pim-sparse nbr-timeout* command to specify the interval the PIM interface waits before declaring that the neighbor is not present. Use the *no* form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the time interval (in seconds) the PIM interface waits before a neighbor is considered not present. Valid range is 30 to 10,800 seconds. |

**Default Values**

By default, the *nbr-timeout* is set to 105 seconds.

**Usage Examples**

The following example specifies a wait interval of 360 seconds on the eth 0/1 interface:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip pim-sparse nbr-timeout 360
```
**ip pim-sparse override-interval** `<time>`

Use the `ip pim-sparse override-interval` command to specify the delay interval after a join/prune in which another router on the LAN may override the join/prune. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;time&gt;</code></th>
<th>Specifies the delay interval (in milliseconds) after a join/prune in which another router on the LAN may override the join/prune. Valid range is 0 to 65,535 milliseconds.</th>
</tr>
</thead>
</table>

**Default Values**

By default, the `override-interval` is set to 2500 milliseconds.

**Usage Examples**

The following example sets the delay interval for join/prune overrides to 3000 milliseconds on the eth 0/1 interface:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip pim-sparse override-interval 3000
```
Use the `ip pim-sparse propagation-delay` command to specify the expected propagation delay for join/prune messages. Set the propagation delay (in milliseconds) to estimate the amount of delay found in the local link. Use the `no` form of this command to return to the default value.

**Syntax Description**

| `<time>` | Specifies the expected propagation delay in the local link in milliseconds. Valid range is 0 to 32,767 milliseconds. |

**Default Values**

By default, the `propagation-delay` is set to 500 milliseconds.

**Usage Examples**

The following example sets the expected propagation delay to 300 milliseconds on the `eth 0/1` interface:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip pim-sparse propagation-delay 300
```
**ip policy route-map <mapname>**

Use the `ip policy route-map` command to assign a policy route map to this interface. Use the `no` form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

**Syntax Description**

- `<mapname>` Specifies the route map to associate with this interface.

**Default Values**

By default, policy-based routing is disabled for all interfaces.

**Usage Examples**

The following example associates the route map named `MyMap` with the eth 0/1 interface:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip policy route-map MyMap
```
**ip proxy-arp** `<ip address> <subnet mask>`

Use the `ip proxy-arp` command to enable proxy Address Resolution Protocol (ARP) on the interface. Use the `no` form of this command to disable this feature.

**Syntax Description**

| `<ip address>` | Defines the proxy ARP IP address in dotted decimal notation (for example: 192.168.73.101). |
| `<subnet mask>` | Specifies the subnet mask that corresponds to the listed IP address. |

**Default Values**

By default, `ip proxy-arp` is enabled.

**Functional Notes**

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all `arp` requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

**Usage Examples**

The following enables proxy ARP on the Ethernet interface:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip proxy-arp
```
ip rip receive version [1 | 2]

Use the `ip rip receive version` command to configure the RIP version the unit accepts in all RIP packets received on the interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accepts RIP version 1 packets received on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Accepts RIP version 2 packets received on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1.

**Functional Notes**

Use the `ip rip receive version` command to specify a RIP version that overrides the `version` (in the Router RIP) configuration.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the Ethernet interface to accept only RIP version 2 packets:

```text
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip rip receive version 2
```
ip rip send version [1 | 2]

Use the ip rip send version command to configure the RIP version the unit sends in all RIP packets transmitted on the interface.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transmits RIP version 1 packets received on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Transmits RIP version 2 packets received on the interface.</td>
</tr>
</tbody>
</table>

Default Values

By default, all interfaces transmit RIP version 1 (the default value for the version command).

Functional Notes

Use the ip rip send version command to specify a RIP version that overrides the version (in the Router RIP) configuration.

The SROS only transmits one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the Ethernet interface to transmit only RIP version 2 packets:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip rip send version 2
```
ip rip summary-address <network address> <network mask>

Use the ip rip summary-address command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the no form of this command to disable this mode.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;network address&gt;</th>
<th>Specifies the IP address of the network to be summarized.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network mask&gt;</td>
<td>Specifies the network mask to be applied to the specific network to be summarized.</td>
</tr>
</tbody>
</table>

Default Values

By default, no manual summarization is applied by RIP.

Functional Notes

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the ip rip summary-address command. This command is only effective if RIP version 2 is configured.

Usage Examples

The following example enables manual summarization on the specified IP address:

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip rip summary-address 10.10.123.0 255.255.255.0
ip route-cache

Use the `ip route-cache` command to enable route caching on the interface. Use the `no` form of this command to disable route caching and return to process switching mode.

**Note**

> Using Network Address Translation (NAT) or the SROS firewall capabilities on an interface requires process switching mode (using the `no ip route-cache` command).

**Syntax Description**

No subcommands.

**Default Values**

By default, route caching is enabled on all interfaces.

**Functional Notes**

Route caching allows an IP interface to provide optimum performance when processing IP traffic.

**Usage Examples**

The following example enables route caching on the Ethernet interface:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip route-cache
```
**ip unnumbered <interface>**

Use the `ip unnumbered` command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the `no` form of this command to remove the unnumbered configuration.

**Syntax Description**

| <interface> | Specifies the interface (in the format **type slot/port**) that contains the IP address to be used as the source address for all packets transmitted on this interface. |

**Default Values**

By default, all interfaces are configured to use a specified IP address (using the `ip address` command).

**Functional Notes**

If `ip unnumbered` is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying `ip unnumbered ppp 1` while in the Ethernet Interface Configuration mode configures the Ethernet interface to use the IP address assigned to the PPP interface for all IP processing. In addition, the SROS uses the specified interface information when sending route updates over the unnumbered interface.

**Usage Examples**

The following example configures the Ethernet interface (labeled `eth 0/1`) to use the IP address assigned to the PPP interface (`ppp 1`):

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#ip unnumbered ppp 1
```
lldp receive

Use the `lldp receive` command to allow LLDP packets to be received on this interface.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces are configured to send and receive LLDP packets.

**Usage Examples**

The following example configures Ethernet interface 0/1 to receive LLDP packets:

```
ProCurve(config-eth 0/1)#lldp receive
```
lldp send [management-address | port-description | system-capabilities | system-description | system-name]

Use the `lldp send` command to configure this interface to transmit LLDP packets or to control the types of information contained in the LLDP packets transmitted by this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>management-address</td>
<td>Enables transmission of management address information on this interface.</td>
</tr>
<tr>
<td>port-description</td>
<td>Enables transmission of port description information on this interface.</td>
</tr>
<tr>
<td>system-capabilities</td>
<td>Enables transmission of this device’s system capabilities on this interface.</td>
</tr>
<tr>
<td>system-description</td>
<td>Enables transmission of this device’s system description on this interface.</td>
</tr>
<tr>
<td>system-name</td>
<td>Enables transmission of this device’s system name on this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

Be default, all interfaces are configured to transmit and receive LLDP packets of all types.

**Functional Notes**

Individual LLDP information can be enabled or disabled using the various forms of the `lldp send` command. For example, use the `lldp send-and-receive` command to enable transmit and receive of all LLDP information. Then use the `no lldp send port-description` command to prevent LLDP from transmitting port description information.

**Usage Examples**

The following example configures Ethernet interface 0/1 to transmit LLDP packets containing all enabled information types:

ProCurve(config-eth 0/1)#lldp send
lldp send-and-receive

Use the lldp send-and-receive command to configure this interface to transmit and receive LLDP packets.

Syntax Description

No subcommands.

Default Values

By default, all interfaces are configured to transmit and receive LLDP packets of all types.

Functional Notes

Individual LLDP information can be enabled or disabled using the various forms of the lldp send command. For example, use the lldp send-and-receive command to enable transmit and receive of all LLDP information. Then use the no lldp send port-description command to prevent LLDP from transmitting port description information.

Usage Examples

The following example configures Ethernet interface 0/1 to transmit and receive LLDP packets containing all information types:

ProCurve(config-eth 0/1)#lldp send-and-receive
**mac-address <address>**

Use the `mac-address` command to specify the Media Access Control (MAC) address of the unit. The first three values contain the reserved Vendor ID (00:12:79) by default. Use the `no` form of this command to return to the default MAC address.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>MAC address entered in a series of six dual-digit hexadecimal values separated by colons (for example 00:12:79:5F:00:D2).</td>
</tr>
</tbody>
</table>

**Default Values**

A unique default MAC address is programmed in each unit.

**Functional Notes**

Some network providers require MAC address registration to connect to their networks. "Locking" access to the public network (based on MAC addresses) can cause problems for multi-computer offices. For example, many cable internet providers register the MAC address of your computer's Ethernet card, limiting the use of the network access to the registered computer. Use the `mac-address` command to program the computer's Ethernet card MAC address into the router to spoof the PC connection. (The cable internet provider cannot distinguish between a router using the registered MAC address and the actual registered computer.) Non-registered computers can connect to the network through the router.

**Usage Examples**

The following example configures a MAC address of **00:12:79:05:30:05**:

```plaintext
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#mac-address 00:12:79:05:30:05
```
max-reserved-bandwidth <percent>

Use the **max-reserved-bandwidth** command to specify the percentage of interface bandwidth reserved for use in user-defined (priority or class-based) queues. The remainder of the interface bandwidth is reserved for system critical traffic and is not available to user-defined queues. Use the **no** form of this command to restore the default values.

**Caution**  
Reserving a portion of the interface bandwidth for system critical traffic is necessary for proper operation. Specifying the entire interface bandwidth for use in user-defined queues can cause undesirable operation.

**Syntax Description**

| <percent> | Specifies the percentage of interface bandwidth to make available for user-defined (priority or class-based) queues. Enter an integer 1 to 100. |

**Default Values**

By default, **max-reserved-bandwidth** is set to **75**, which reserves 25 percent of the interface bandwidth for system critical traffic.

**Usage Examples**

The following example specifies 85 percent of the bandwidth on the eth 0/1 interface to be available for use in user-defined queues:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#max-reserved-bandwidth 85
```
mtu <size>

Use the mtu command to configure the maximum transmit unit (MTU) size for the active interface. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>64 to 256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>64 to 18,190</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>64 to 1500</td>
</tr>
</tbody>
</table>

Default Values

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>The default values for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>1500</td>
</tr>
</tbody>
</table>

Functional Notes

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

Usage Examples

The following example specifies an MTU of 1200 on the Ethernet interface:

ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#mtu 1200
port-auth supplicant [username <username> | password <password>]

Use the **port-auth supplicant** command to enable supplicant functionality and to specify the username and password used for IEEE 802.1x port authentication. The supplicant is the port that will receive services from the port authenticator.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username &lt;username&gt;</td>
<td>Specifies the username to use during the authentication process. The default username is <em>username</em>.</td>
</tr>
<tr>
<td>password &lt;password&gt;</td>
<td>Specifies the password to use during the authentication process. The default password is <em>password</em>.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command disabled.

**Functional Notes**

If your network infrastructure is configured to use 802.1x on every port, configure the router to function as an 802.1x client. The router (when configured as a 802.1x client) passes username and password information to the connected switch allowing the switch to authenticate the user (router) and enable the port.

**Usage Examples**

The following example sets the username to **User1** and sets the password to **securePass** for Ethernet interface 0/2:

```
ProCurve(config)#int eth 0/2
ProCurve(config-eth 0/2)#port-auth supplicant username User1 password securePass
```
qos-policy out <mapname>

Use the qos-policy out command to apply a previously-configured QoS map to an interface. Use the no form of this command to remove the map from the interface. The out keyword specifies that this policy will be applied to outgoing packets.

Syntax Description

- `<mapname>` Specifies the name of a previously-created QoS map (see qos map `<mapname>` `<sequence number>` on page 451 for more information).

Default Values

No default value is necessary for this command.

Usage Examples

The following example applies the QoS map MYMAP to the Ethernet 0/1 interface:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#qos-policy out MYMAP
```
**snmp trap**

Use the `snmp trap` command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces (except virtual Frame Relay interfaces and sub-interfaces) have SNMP traps enabled.

**Usage Examples**

The following example enables SNMP capability on the Ethernet interface:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#snmp trap
```
**snmp trap link-status**

Use the `snmp trap link-status` command to control the Simple Network Management Protocol (SNMP) variable `ifLinkUpDownTrapEnable` (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the `no` form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the `ifLinkUpDownTrapEnable` OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The `snmp trap link-status` command is used to control the RFC2863 `ifLinkUpDownTrapEnable` OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the interface:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#no snmp trap link-status
```
spanning-tree bpdufilter [enable | disable]

Use the `spanning-tree bpdufilter` command to enable or disable the bpdufilter on a specific interface. This setting overrides the related global setting. Use the `no` version of the command to return to the default setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enable</code></td>
<td>Enables bpdufilter for this interface.</td>
</tr>
<tr>
<td><code>disable</code></td>
<td>Disables bpdufilter for this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this setting is disabled.

**Functional Notes**

The bpdufilter blocks any BPDUs from being transmitted and received on an interface.

**Usage Examples**

The following example enables the bpdufilter on the interface eth 0/1:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#spanning-tree bpdufilter enable
```

The bpdufilter can be disabled on the eth 0/1 by issuing the following commands:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#spanning-tree bpdufilter disable
```
spanning-tree bpduguard [enable | disable]

Use the `spanning-tree bpduguard` command to enable or disable the bpduguard on a specific interface. This setting overrides the related global setting (see `spanning-tree forward-time <seconds>` on page 480). Use the `no` version of the command to return to the default setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables bpduguard for this interface.</td>
</tr>
<tr>
<td>disable</td>
<td>Disables bpduguard for this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this setting is disabled.

**Functional Notes**

The bpduguard blocks any BPDUs from being received on an interface.

**Usage Examples**

The following example enables the bpduguard on the interface eth 0/1:

```plaintext
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#spanning-tree bpduguard enable
```

The bpduguard can be disabled on the eth 0/1 by issuing the following commands:

```plaintext
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#spanning-tree bpduguard disable
```
**spanning-tree edgeport**

Use the `spanning-tree edgeport` command to configure the interface to be an edgeport. This command overrides the related Global setting. Use the `no` version of the command to return to the default setting.

**Syntax Description**

No subcommands.

**Default Values**

By default, this setting is disabled.

**Functional Notes**

Enabling this command configures the interface to go to a forwarding state when the link goes up.

**Usage Examples**

The following example configures the interface to be an edgeport:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#spanning-tree edgeport
```

An individual interface can be configured to not be considered an edgeport. For example:

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#spanning-tree edgeport disable
```

or

```
ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#no spanning-tree edgeport
```
spanning-tree link-type [auto | point-to-point | shared]

Use the `spanning-tree link-type` command to configure the spanning tree protocol link type for each interface. Use the `no` version of the command to return to the default setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>auto</th>
<th>Link type is determined by the port's duplex settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>point-to-point</td>
<td>Link type is manually set to point-to-point, regardless of duplex settings.</td>
</tr>
<tr>
<td>shared</td>
<td>Link type is manually set to shared, regardless of duplex settings.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the interface is set to auto.

**Functional Notes**

This command overrides the default link type setting determined by the duplex of the individual port. By default, a port configured for half-duplex is set to `shared` link type, and a port configured for full-duplex is set to `point-to-point` link type. Setting the link type manually overrides the default and forces the port to use the specified link type. Use the `link-type auto` command to restore the convention of determining link type based on duplex settings.

**Usage Examples**

The following example forces the link type to `point-to-point`, even if the port is configured to be half-duplex:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#spanning-tree link-type point-to-point
```

**Technology Review**

Rapid transitions are possible in RSTP (rapid spanning-tree protocol) by taking advantage of point-to-point links (a port is connected to exactly one other bridge) and edge-port connections (a port is not connected to any additional bridges). Setting the link-type to `auto` allows the spanning-tree to automatically configure the link type based on the duplex of the link. Setting the link type to `point-to-point` allows a half-duplex link to act as if it were a point-to-point link.
spanning-tree port-priority <priority level>

Use the spanning-tree port-priority command to select the priority level of this interface. To return to the default setting, use the no version of this command.

Syntax Description

| <priority level> | Specifies a value from 0 to 255. |

Default Values

By default, this set to 128.

Functional Notes

The only time that this priority level is used is when two interfaces with a path to the root have equal cost. At that point, the level set in this command will determine which port the spanning-tree will use. Set the priority value lower to increase the chance the interface will be used.

Usage Examples

The following example sets the interface to a priority of 100:

ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#spanning-tree port-priority 100
speed [10 | 100 | auto]

Use the speed command to configure the speed of an Ethernet interface. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>Speed</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10 Mb Ethernet</td>
</tr>
<tr>
<td>100</td>
<td>100 Mb Ethernet</td>
</tr>
<tr>
<td>auto</td>
<td>Automatically detects 10 or 100 Mb Ethernet and negotiates the duplex setting in the following order: 100/full, 100/half, 10/full, 10/half.</td>
</tr>
</tbody>
</table>

**Note** Some Ethernet equipment (though rare) is unable to negotiate duplex if speed is manually determined. To avoid incompatibilities, manually set the duplex if the speed is manually set. See speed [10 | 100 | auto] on page 623 for more information.

Default Values

By default, speed is set to **auto**.

Usage Examples

The following example configures the Ethernet port for 100 Mb operation:

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#speed 100
traffic-shape rate <rate> <burstrate>

Use the traffic-shape rate command to specify and enforce an output bandwidth for Ethernet and VLAN interfaces.

Syntax Description

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;rate&gt;</td>
<td>Specifies the rate (in bits per second) at which the interface should be shaped.</td>
</tr>
<tr>
<td>&lt;burstrate&gt;</td>
<td>Optional. Specifies the allowed burst in bytes. By default, this is specified to the rate divided by 5 to represent the number of bytes that would flow within 200 ms.</td>
</tr>
</tbody>
</table>

Default Values

By default, traffic-shaping rate is disabled.

Functional Notes

Traffic shaping can be used to limit an Ethernet segment to a particular rate or to specify use of QoS on Ethernet or VLAN interfaces.

Usage Examples

The following example sets the outbound rate of eth 0/1 to 128 kbps and applies a QoS policy that all RTP traffic is given priority over all other traffic:

ProCurve(config)#qos map voip 1
ProCurve(config-qos-map)#match ip rt p 10000 105000 all
ProCurve(config-qos-map)#priority unlimited
ProCurve(config-qos-map)#interface eth 0/1
ProCurve(config-eth 0/1)#traffic-shape rate 128000
ProCurve(config-eth 0/1)#qos-policy out voip
Use the `vlan-id` command to set a VLAN ID for the Ethernet interface. Use the `no` form of this command to remove an entry.

### Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;vlan id&gt;</code></td>
<td>Specifies a valid VLAN interface ID number (1 to 4095).</td>
</tr>
<tr>
<td><code>native</code></td>
<td>Optional. Specifies that data for that VLAN ID goes out untagged. If <code>native</code> is not specified, data for that VLAN ID goes out tagged.</td>
</tr>
</tbody>
</table>

### Default Values

By default, no VLAN ID is set.

### Usage Examples

The following example configures a native VLAN of 5 for the Ethernet interface 0/1:

```
ProCurve(config)#interface eth 0/1
ProCurve(config-eth 0/1)#vlan-id 5 native
```
G.703 INTERFACE CONFIGURATION COMMAND SET

To activate the G.703 Interface Configuration mode, enter the `interface e1` command (and specify the G.703 port) at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface e1 1/2
ProCurve(config-e1 1/2)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `coding [ami | hdb3]` on page 627
- `framing [crc4]` on page 628
- `loopback network [line | payload]` on page 629
- `snmp trap link-status` on page 630
- `test-pattern [ones | zeros]` on page 631
- `ts16` on page 632
coding [ami | hdb3]

Use the coding command to configure the line coding for the G.703 physical interface. This setting must match the line coding supplied on the circuit by the PBX.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ami</td>
<td>Configures the line coding for alternate mark inversion (AMI).</td>
</tr>
<tr>
<td>hdb3</td>
<td>Configures the line coding for high-density bipolar 3 (HDB3).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all E1 interfaces are configured with HDB3 line coding.

**Functional Notes**

The line coding configured in the unit must match the line coding of the E1 circuit. A mismatch will result in line errors (e.g., BPVs).

**Usage Examples**

The following example configures the G.703 interface for AMI line coding:

```
ProCurve(config)#interface e1 1/2
ProCurve(config-e1 1/2)#coding ami
```
framing [crc4]

Use the **framing** command to configure the framing format for the G.703 interface. This parameter should match the framing format set on the external device. Use the **no** form of this command to return to the default value.

**Syntax Description**

| **crc4** | Enables CRC-4 bits to be transmitted in the outgoing data stream. Also, the received signal is checked for CRC-4 errors. |

**Default Values**

By default, CRC-4 framing is enabled.

**Functional Notes**

The framing value must match the configuration of the E1 circuit. A mismatch will result in a loss of frame alarm.

**Usage Examples**

The following example configures the G.703 interface for CRC-4 framing:

```
ProCurve(config)#interface e1 1/2
ProCurve(config-e1 1/2)#framing crc4
```
loopback network [line | payload]

Use the `loopback network` command to initiate a loopback on the interface toward the network. Use the `no` form of this command to deactivate the loopback.

**Syntax Description**

- **line**
  - Initiates a metallic loopback of the physical E1 network interface.

- **payload**
  - Initiates a loopback of the E1 framer (CSU portion) of the E1 network interface.

**Default Values**

No default necessary for this command.

**Functional Notes**

The following diagram depicts a line loopback.

**Usage Examples**

The following example initiates a line loopback of the G.703 interface:

ProCurve(config)#interface e1 1/2
ProCurve(config-e1 1/2)#loopback network line
snmp trap link-status

Use the **snmp trap link-status** to control the Simple Network Management Protocol (SNMP) variable *ifLinkUpDownTrapEnable* (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the *ifLinkUpDownTrapEnable* OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The **snmp trap link-status** command is used to control the RFC2863 *ifLinkUpDownTrapEnable* OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the G.703 interface:

```
ProCurve(config)#interface e1 1/2
ProCurve(config-e1 1/2)#no snmp trap link-status
```
test-pattern [ones | zeros]

Use the **test-pattern** command to activate the built-in pattern generator and begin sending the specified test pattern. This pattern generation can be used to verify a data path when used in conjunction with an active loopback. Use the **no** form of this command to cease pattern generation.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ones</td>
<td>Generates a test pattern of continuous ones.</td>
</tr>
<tr>
<td>zeros</td>
<td>Generates a test pattern of continuous zeros.</td>
</tr>
</tbody>
</table>

**Default Values**

No defaults necessary for this command.

**Usage Examples**

The following example activates the pattern generator for a stream of continuous ones:

```
ProCurve(config)#interface e1 1/2
ProCurve(config-e1 1/2)#test-pattern ones
```
ts16

Use the **ts16** command to enable timeslot 16 multiframe to be checked on the receive signal. Use the **no** form of this command to disable timeslot 16.

**Syntax Description**

No subcommands.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

If timeslot 16 is used on the incoming E1, do not map timeslot 16 using the **tdm-group** command. By default, all timeslots not physically mapped using the **tdm-group** command are passed through to the G.703 interface. Leaving timeslot 16 unmapped makes it available for multiframe signaling by the connected E1 device.

**Usage Examples**

The following example enables timeslot 16 multframing:

```
ProCurve(config)#interface e1 1/2
ProCurve(config-e1 1/2)#ts16
```
SERIAL INTERFACE CONFIGURATION COMMAND SET

To activate the Serial Interface Configuration mode, enter the `interface serial` command at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `et-clock-source [rxclock | txclock]` on page 634
- `ignore dcd` on page 635
- `invert etclock` on page 636
- `invert rxclock` on page 637
- `invert txclock` on page 638
- `serial-mode [v35 | x21]` on page 639
- `snmp trap` on page 640
- `snmp trap link-status` on page 641
et-clock-source [rxclock | txclock]

Use the et-clock-source command to configure the clock source used when creating the external transmit reference clock (et-clock). Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rxclock</td>
<td>Specifies using the clock recovered from the receive signal to generate et-clock.</td>
</tr>
<tr>
<td>txclock</td>
<td>Specifies using the clock recovered from the transmit signal to generate et-clock.</td>
</tr>
</tbody>
</table>

Default Values

By default, the clock recovered from the transmit signal is used to generate the et-clock.

Functional Notes

The et-clock is an interface timing signal (provided by the DTE device) used to synchronize the transfer of transmit data.

Usage Examples

The following example configures the serial interface to recover the clock signal from the received signal and use it to generate et-clock:

ProCurve(config)#interface serial 1/1
ProCurve(config-serial 1/1)#et-clock-source rxclock
**ignore dcd**

Use the **ignore dcd** command to specify the behavior of the serial interface when the Data Carrier Detect (DCD) signal is lost. Use the **no** form of this command to return to the default value.

**Syntax Description**

No subcommands.

**Default Values**

By default, the serial interface does not ignore a change in status of the DCD signal.

**Functional Notes**

When configured to follow DCD (default condition), the serial interface will not attempt to establish a connection when DCD is not present. When configured to ignore DCD, the serial interface will continue to attempt to establish a connection even when DCD is not present.

**Usage Examples**

The following example configures the serial interface to ignore a loss of the DCD signal:

ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#ignore dcd
invert etclock

Use the invert etclock command to configure the serial interface to invert the external transmit reference clock (et-clock) in the data stream before transmitting. Use the no form of this command to return to the default value.

Syntax Description

No subcommands.

Default Values

By default, the serial interface does not invert et-clock.

Functional Notes

If the serial interface cable is long, causing a phase shift in the data, the et-clock can be inverted using the invert etclock command. This switches the phase of the clock, which compensates for a long cable.

Usage Examples

The following example configures the serial interface to invert et-clock:

ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#invert etclock
invert rxclock

Use the invert rxclock command to configure the serial interface to expect an inverted receive clock (found in the received data stream). Use the no form of this command to return to the default value.

**Syntax Description**

No subcommands.

**Default Values**

By default, the serial interface does not expect an inverted receive clock (rxclock).

**Functional Notes**

If the serial interface cable is long, causing a phase shift in the data, the transmit clock can be inverted (using the invert txclock command). This switches the phase of the clock, which compensates for a long cable. If the transmit clock of the connected device is inverted, use the invert rxclock command to configure the receiving interface appropriately.

**Usage Examples**

The following example configures the serial interface to invert receive clock:

```
ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#invert rxclock
```
invert txclock

Use the invert txclock command to configure the serial interface to invert the transmit clock (found in the transmitted data stream) before sending the signal. Use the no form of this command to return to the default value.

Syntax Description

No subcommands.

Default Values

By default, the serial interface does not invert transmit clock (txclock).

Functional Notes

If the serial interface cable is long, causing a phase shift in the data, the transmit clock can be inverted (using the invert txclock command). This switches the phase of the clock, which compensates for a long cable. If the transmit clock of the connected device is inverted, use the invert rxclock command to configure the receiving interface appropriately.

Usage Examples

The following example configures the serial interface to invert the transmit clock:

ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#invert txclock
serial-mode [v35 | x21]

Use the `serial-mode` command to specify the electrical mode for the interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V35</td>
<td>Configures the interface for use with the V.35 adapter cable (J8757A).</td>
</tr>
<tr>
<td>X21</td>
<td>Configures the interface for use with the X.21 adapter cable (J8755A).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the serial interface is configured for a V.35 adapter cable.

**Functional Notes**

The pinouts for each of the available interfaces can be found in the *Hardware Configuration Guide* located on the *ProCurve SROS Documentation* CD (provided in your shipment).

**Usage Examples**

The following example configures the serial interface to work with the X.21 adapter cable:

```
ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#serial-mode X21
```
snmp trap

Use the snmp trap command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface. Use the no form of this command to disable SNMP on the interface.

Syntax Description

No subcommands.

Default Values

By default, all interfaces (except virtual Frame Relay interfaces and sub-interfaces) have SNMP traps enabled.

Usage Examples

The following example enables SNMP on the serial interface:

ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#snmp trap
snmp trap link-status

Use the **snmp trap link-status** command to control the Simple Network Management Protocol (SNMP) variable `ifLinkUpDownTrapEnable` per RFC2863 to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the `ifLinkUpDownTrapEnable` OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The **snmp trap link-status** command is used to control the RFC2863 `ifLinkUpDownTrapEnable` OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the serial interface:

```
ProCurve(config)#interface serial 1/1
ProCurve(config-ser 1/1)#no snmp trap link-status
```
MODEM INTERFACE CONFIGURATION COMMAND SET

To activate the Modem Interface Configuration mode, enter the `interface modem` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface modem 1/2
ProCurve(config-modem 1/2)#
```

**Note**  The modem interface number in the example above is shown as `modem 1/2`. This number is based on the interface's location (slot/port) and could vary depending on the unit’s configuration. Use the **do show interfaces** command to determine the appropriate interface number.

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `caller-id override [always <number> | if-no-cid <number>]` on page 643
- `dialin` on page 644
- `init-string <string>` on page 645
- `resource pool-member <pool-name> [\<cost\>]` on page 646
caller-id override [always <number> | if-no-cid <number>]

Use the caller-id override command to configure the unit to replace caller ID information with a user-specified number. Use the no form of this command to disable any caller ID overrides.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>always &lt;number&gt;</td>
<td>Always forces replacement of the incoming caller ID number with the number given.</td>
</tr>
<tr>
<td>if-no-cid &lt;number&gt;</td>
<td>Replaces the incoming caller ID number with the number given only if there is no caller ID information available for the incoming call.</td>
</tr>
</tbody>
</table>

Default Values

By default, this command is disabled.

Functional Notes

This command forces a replacement of the incoming caller ID number with the number given. The received caller ID, if any, is discarded, and the given override number is used to connect the incoming call to a circuit of the same number.

Usage Examples

The following example configures the unit to always provide the given number as the caller ID number:

ProCurve(config)#interface modem 1/2
ProCurve(config-modem 1/2)#caller-id override always 5555555
dialin

Use the **dialin** command to enable the modem for remote console dial-in, disabling the use of the modem for backup. Use the **no** form of this command to disable this feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, **dialin** is disabled.

**Usage Examples**

The following example enables remote console dial-in:

```
ProCurve(config)#interface modem 1/2
ProCurve(config-modem 1/2)#dialin
```
init-string <string>

Use the init-string command to specify an initialization string for the modem using standard AT commands. Use the no form of this command to return to the default initialization string.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;string&gt;</th>
<th>Specifies an initialization string using standard AT commands. This string must start with AT and cannot contain spaces.</th>
</tr>
</thead>
</table>

Default Values

<table>
<thead>
<tr>
<th>&lt;string&gt;</th>
<th>ate0q0v1x4\n0</th>
</tr>
</thead>
<tbody>
<tr>
<td>at</td>
<td>All initialization strings must begin with AT.</td>
</tr>
<tr>
<td>e0</td>
<td>Disables command echo.</td>
</tr>
<tr>
<td>q0</td>
<td>Response messages on.</td>
</tr>
<tr>
<td>v1</td>
<td>Formats result codes in long word form.</td>
</tr>
<tr>
<td>x4</td>
<td>Specifies extended response set, dial tone, and busy signal detection for result codes following modem operations.</td>
</tr>
<tr>
<td>\n0</td>
<td>Selects standard buffered connection only.</td>
</tr>
</tbody>
</table>

Usage Examples

The following example configures the modem to perform a hang-up at each initialization (to verify that the line is free) and maintains the default initialization:

ProCurve(config)#interface modem 1/2
ProCurve(config-modem 1/2)#init-string ate0h0q0v1x4\n0
resource pool-member <pool-name> [<cost>]

Use the resource pool-member command to assign the interface to a resource pool, making it a demand routing resource. Use the no form of this command to return to the default value.

Syntax Description

<pool-name>
Specifies the name of the resource pool to which this interface is assigned.

<cost>
Optional. Specifies the cost of using this resource interface within the specified pool. In the event of a tie, a resource with a lower cost will be selected first. Interfaces with the same cost will be selected in alphabetical order by interface name.

Default Values

By default, the interface is not assigned to any resource pool.

Usage Examples

The following example configures an analog modem interface as a member of resource pool MyPool:

ProCurve(config)#interface modem 1/2
ProCurve(config-modem 1/2)#resource pool-member MyPool
T1 INTERFACE CONFIGURATION COMMAND SET

To activate the T1 Interface Configuration mode, enter the `interface t1` command at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# interface t1 1/1
ProCurve(config-t1 1/1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `clock source [internal | line | through]` on page 648
- `coding [ami | b8zs]` on page 649
- `fdl [ansi | att | none]` on page 650
- `framing [d4 | esf]` on page 651
- `lbo [long <-22.5, -15, -7.5, 0> | short <0-655>]` on page 652
- `loopback commands` begin on page 653
- `remote-alarm [rai]` on page 656
- `remote-loopback` on page 657
- `snmp trap line-status` on page 658
- `snmp trap link-status` on page 659
- `tdm-group <group number> timeslots <1-24> speed [56 | 64]` on page 660
- `test-pattern [clear | errors | insert | ones | p215 | p220 | p511 | qrss | zeros]` on page 661
clock source [internal | line | through]

Use the `clock source` command to configure the source timing used for the interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>internal</strong></td>
<td>Configures the unit to provide clocking using the internal oscillator.</td>
</tr>
<tr>
<td><strong>line</strong></td>
<td>Configures the unit to recover clocking from the T1 circuit.</td>
</tr>
<tr>
<td><strong>through</strong></td>
<td>Configures the unit to recover clocking from the circuit connected to the DSX-1 interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the `clock source` is set to `line`.

**Functional Notes**

When operating on a circuit that is providing timing, setting the `clock source` to `line` can avoid errors such as Clock Slip Seconds (CSS).

**Usage Examples**

The following example configures the unit to recover clocking from the primary circuit:

ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#clock source line
coding [ami | b8zs]

Use the coding command to configure the line coding for a T1 physical interface. This setting must match the line coding supplied on the circuit by the service provider.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ami</td>
<td>Configures the line coding for alternate mark inversion (AMI).</td>
</tr>
<tr>
<td>b8zs</td>
<td>Configures the line coding for bipolar eight zero substitution (B8ZS).</td>
</tr>
</tbody>
</table>

Default Values

By default, all T1 interfaces are configured with B8ZS line coding.

Functional Notes

The line coding configured in the unit must match the line coding of the T1 circuit. A mismatch will result in line errors (e.g., BPVs).

Usage Examples

The following example configures the T1 interface for AMI line coding:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#coding ami
```
Use the `fdl` command to configure the format for the facility data link (FDL) channel on the T1 circuit. FDL channels are only available on point-to-point circuits. Use the `no` form of this command to return to the default value.

**Syntax Description**
- `ansi` Configures the FDL for ANSI T1.403 standard.
- `att` Configures the FDL for AT&T TR 54016 standard.
- `none` Disables FDL on this circuit.

**Default Values**
By default, the FDL is configured for `ansi`.

**Functional Notes**
T1 circuits using ESF framing format (specified using the `framing` command) reserve 12 bits as a data link communication channel, referred to as the FDL, between the equipment on either end of the circuit. The FDL allows the transmission of trouble flags such as the Yellow Alarm signal. See `framing [d4 | esf]` on page 651 for related information.

**Usage Examples**
The following example disables the FDL channel for the T1 circuit:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#fdl none
```
framing [d4 | esf]

Use the `framing` command to configure the framing format for the T1 interface. This parameter should match the framing format supplied by your network provider. Use the `no` form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d4</td>
<td>Specifies D4 superframe (SF) format.</td>
</tr>
<tr>
<td>esf</td>
<td>Specifies extended superframe (ESF) format.</td>
</tr>
</tbody>
</table>

Default Values

By default, the framing format is configured for `esf`.

Functional Notes

A frame is comprised of a single byte from each of the T1’s timeslots; there are 24 timeslots on a single T1 circuit. Framing bits are used to separate the frames and indicate the order of information arriving at the receiving equipment. D4 and ESF are two methods of collecting and organizing frames over the circuit.

Usage Examples

The following example configures the T1 interface for D4 framing:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#framing d4
```
lbo [long <-22.5, -15, -7.5, 0> | short <0-655>]

Use the lbo command to configure the line build out (LBO) for the T1 interface. Use the no form of this command to return to the default value.

**Syntax Description**

- **long <-22.5, -15, -7.5, 0>** Configures the LBO (in dB) for T1 interfaces with cable lengths greater than 655 feet. Choices are -22.5, -15, -7.5, and 0 dB.

- **short <0-655>** Configures the LBO (in feet) for T1 interfaces with cable lengths less than 655 feet. Range is 0 to 655 feet.

**Default Values**

By default, the build out is set to 0 dB.

**Functional Notes**

Line build out (LBO) is artificial attenuation of a T1 output signal to simulate a degraded signal. This is useful to avoid overdriving a receiver’s circuits. The shorter the distance between T1 equipment (measured in cable length), the greater the attenuation value. For example, two units in close proximity should be configured for the maximum attenuation (-22.5 dB).

**Usage Examples**

The following example configures the T1 interface LBO for -22.5 dB:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#lbo long -22.5
```
loopback network [line | payload]

Use the **loopback network** command to initiate a loopback on the interface toward the network. Use the **no** form of this command to deactivate the loopback.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>line</td>
<td>Initiates a metallic loopback of the physical T1 network interface.</td>
</tr>
<tr>
<td>payload</td>
<td>Initiates a loopback of the T1 framer (CSU portion) of the T1 network interface.</td>
</tr>
</tbody>
</table>

**Default Values**

No default necessary for this command.

**Functional Notes**

The following diagram depicts the difference between a line and payload loopback.

![Diagram of T1 Network Interface showing line and payload loopback](image)

**Usage Examples**

The following example initiates a payload loopback of the T1 interface:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#loopback network payload
```
loopback remote line [fdl | inband]

Use the `loopback remote line` command to send a loopback code to the remote unit to initiate a line loopback. Use the `no` form of this command to send a loopdown code to the remote unit to deactivate the loopback.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fdl</td>
<td>Uses the facility data link (FDL) to initiate a full 1.544 Mbps physical (metallic) loopback of the signal received by the remote unit from the network.</td>
</tr>
<tr>
<td>inband</td>
<td>Uses the inband channel to initiate a full 1.544 Mbps physical (metallic) loopback of the signal received by the remote unit from the network.</td>
</tr>
</tbody>
</table>

**Default Values**

No defaults necessary for this command.

**Functional Notes**

The following diagram depicts the difference between a line and payload loopback.

![Diagram showing the difference between line and payload loopback]

**Usage Examples**

The following example initiates a remote line loopback using the FDL:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#loopback remote line fdl
```
**loopback remote payload**

Use the `loopback remote payload` command to send a loopback code to the remote unit to initiate a payload loopback. A payload loopback is a 1.536 Mbps loopback of the payload data received from the network maintaining bit-sequence integrity for the information bits by synchronizing (regenerating) the timing. Use the `no` form of this command to send a loopdown code to the remote unit to deactivate the loopback.

**Syntax Description**

No subcommands.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

The following diagram depicts the difference between a line and payload loopback.

![Diagram of T1 Network Interface showing Line and Payload Loopback]

**Usage Examples**

The following example initiates a remote payload loopback:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#loopback remote payload
```
remote-alarm [rai]

The remote-alarm command selects the alarm signaling type to be sent when a loss of frame is detected on the T1 receive signal. Use the no form of this command to disable all transmitted alarms.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rai</td>
<td>Specifies sending a remote alarm indication (RAI) in response to a loss of frame. Also prevents a received RAI from causing a change in interface operational status.</td>
</tr>
</tbody>
</table>

Default Values

The default for this command is rai.

Usage Examples

The following example enables transmission of RAI in response to a loss of frame:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#remote-alarm rai
```
remote-loopback

Use the `remote-loopback` command to configure the interface to respond to loopbacks initiated by a remote unit (or the service provider). Use the `no` form of this command to disable this feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces respond to remote loopbacks.

**Usage Examples**

The following example enables remote loopbacks on the T1 interface:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#remote-loopback
```
snmp trap line-status

Use the `snmp trap line-status` command to control the Simple Network Management Protocol (SNMP) variable dsx1LineStatusChangeTrapEnable (RFC2495) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the `no` form of this command to disable this trap.

Syntax Description

No subcommands.

Default Values

By default, the dsx1LineStatusChangeTrapEnable OID is set to enabled for all interfaces except virtual Frame Relay Interfaces.

Functional Notes

The `snmp trap line-status` command is used to control the RFC2495 dsx1LineStatusChangeTrapEnable OID (OID number 1.3.6.1.2.1.10.18.6.1.17.0).

Usage Examples

The following example disables the line-status trap on the T1 interface:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#no snmp trap line-status
```
snmp trap link-status

Use the snmp trap link-status to control the Simple Network Management Protocol (SNMP) variable ifLinkUpDownTrapEnable (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the no form of this command to disable this trap.

Syntax Description

No subcommands.

Default Values

By default, the ifLinkUpDownTrapEnable OID is enabled for all interfaces except virtual Frame Relay interfaces.

Functional Notes

The snmp trap link-status command is used to control the RFC2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the T1 interface:

ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1 1/1)#no snmp trap link-status
tdm-group <group number> timeslots <1-24> speed [56 | 64]

Use the tdm-group command to create a group of contiguous DS0s on this interface to be used during the bind process. See bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port> on page 1304 for related information.

**Caution** Changing tdm-group settings could result in service interruption.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group number&gt;</td>
<td>Identifies the created tdm-group by numerical label (valid range: 1 to 255).</td>
</tr>
<tr>
<td>timeslots &lt;1-24&gt;</td>
<td>Specifies the DS0s to be used in the TDM group. This can be entered as a single number representing one of the 24 T1 channel timeslots or as a contiguous group of DS0s. (For example, 1-10 specifies the first 10 channels of the T1.)</td>
</tr>
<tr>
<td>speed [56</td>
<td>64]</td>
</tr>
</tbody>
</table>

### Default Values

By default, there are no configured TDM group.

### Usage Examples

The following example creates a TDM group (labeled 5) of 10 DS0s at 64 kbps each:

```plaintext
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#tdm-group 5 timeslots 1-10 speed 64
```
test-pattern [clear | errors | insert | ones | p215 | p220 | p511 | qrss | zeros]

Use the **test-pattern** command to activate the built-in pattern generator and begin sending the specified test pattern. This pattern generation can be used to verify a data path when used in conjunction with an active loopback. Use the **no** form of this command to cease pattern generation.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear</td>
<td>Clears the test pattern error count. Display the error count using the <strong>errors</strong> keyword.</td>
</tr>
<tr>
<td>errors</td>
<td>Displays the test pattern error count.</td>
</tr>
<tr>
<td>insert</td>
<td>Inserts an error into the currently active test pattern. Display the error count using the <strong>errors</strong> keyword.</td>
</tr>
<tr>
<td>ones</td>
<td>Generates a test pattern of continuous ones.</td>
</tr>
<tr>
<td>p215</td>
<td>Generates a pseudorandom test pattern sequence based on a 15-bit shift register.</td>
</tr>
<tr>
<td>p220</td>
<td>Generates a pseudorandom test pattern sequence based on a 20-bit shift register.</td>
</tr>
<tr>
<td>p511</td>
<td>Generates a repeating test pattern of ones and zeros.</td>
</tr>
<tr>
<td>qrss</td>
<td>Generates a random test pattern of ones and zeroes.</td>
</tr>
<tr>
<td>zeros</td>
<td>Generates a test pattern of continuous zeros.</td>
</tr>
</tbody>
</table>

**Default Values**

No defaults necessary for this command.

**Usage Examples**

The following example activates the pattern generator for a stream of continuous ones:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#test-pattern ones
```
The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `snmp trap` on page 663
- `snmp trap link-status` on page 664
Use the **snmp trap** command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces (except virtual Frame Relay interfaces and sub-interfaces) have SNMP traps enabled.

**Usage Examples**

The following example enables SNMP on the ATM interface:

```
ProCurve(config)#interface atm 1
ProCurve(config-atm 1)#snmp trap
```
**snmp trap link-status**

Use the **snmp trap link-status** command to control the Simple Network Management Protocol (SNMP) variable *ifLinkUpDownTrapEnable* (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the *ifLinkUpDownTrapEnable* OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The **snmp trap link-status** command is used to control the RFC2863 *ifLinkUpDownTrapEnable* OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the ATM interface:

```
ProCurve(config)#interface atm 1
ProCurve(config-atm 1)#no snmp trap link-status
```
ATM SUB-INTERFACE CONFIG COMMAND SET

To activate the ATM Interface Configuration mode, enter the `interface atm` command (and specify a sub-interface) at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `access-policy <policyname>` on page 667
- `atm routed-bridged ip` on page 668
- `backup commands` begin on page 669
- `bandwidth <value>` on page 686
- `bridge-group <group#>` on page 687
- `crypto map <mapname>` on page 688
- `dynamic-dns` on page 690
- `encapsulation [aal5mux ip | aal5mux ppp | aal5snap]` on page 692
- `fair-queue <threshold>` on page 693
- `hold-queue <queue size> out` on page 694
- `ip commands` begin on page 695
- `max-reserved-bandwidth <percent>` on page 722
- `mtu <size>` on page 723
- `oam-pvc managed <frequency>` on page 724
- `oam retry <up-count> <down-count> <retry-frequency>` on page 725
- `pvc <VPI/VCI>` on page 726
qos-policy out <mapname> on page 727

spanning-tree commands begin on page 729
access-policy <policyname>

Use the access-policy command to assign a specified access policy for the inbound traffic on an interface. Use the no form of this command to remove an access policy association.

**Note** Configured access policies will only be active if the ip firewall command has been entered at the Global Configuration mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.

**Syntax Description**

| <policyname> | Identifies the configured access policy by alphanumeric descriptor for (all access policy descriptors are case-sensitive). |

**Default Values**

By default, there are no configured access policies associated with an interface.

**Functional Notes**

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter access policy <policy name>. For more details on creating and using access policies, refer to ip policy-class <policyname> on page 400.

**Usage Examples**

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the ATM sub-interface labeled 1.1:

Enable the SROS security features:
ProCurve(config)#ip firewall

Create the access list (this is the packet selector):
ProCurve(config)#ip access-list extended InWeb
ProCurve(config-ext-nacl)#permit tcp any host 63.12.5.253 eq 80

Create the access policy that contains the access list InWeb:
ProCurve(config)#ip policy-class UnTrusted
ProCurve(config-policy-class)#allow list InWeb

Associate the access list with the ATM 1.1 interface:
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#access-policy UnTrusted
**atm routed-bridged ip**

Use the `atm routed-bridged ip` command to enable routed IP bridge encapsulation (RBE) on an interface. Use the `no` form of this command to disable RBE operation.

**Syntax Description**

No subcommands.

**Default Values**

By default, routed bridge encapsulation is disabled.

**Usage Examples**

The following example enables routed bridge encapsulation:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#atm routed-bridged ip
```
backup auto-backup

Use the backup auto-backup command to configure the sub-interface to automatically attempt a backup upon failure. Use the no form of this command to disable automatic backup on an interface.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically attempt backup upon a failure.

Usage Examples

The following enables automatic backup on the endpoint:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup auto-backup
**backup auto-restore**

Use the `backup auto-restore` command to configure the sub-interface to automatically discontinue backup when all network conditions are operational. Use the `no` form of this command to disable the auto-restore feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, all backup endpoints will automatically restore the primary connection when the failure condition clears.

**Usage Examples**

The following example configures the SROS to restore the primary connection automatically when the failure condition clears:

```plaintext
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup auto-restore
```
backup backup-delay <seconds>

Use the backup backup-delay command to configure the amount of time the router will wait after the failure condition is recognized before attempting to backup the link. Use the no form of this command to return to the default value.

Syntax Description

| <seconds> | Specifies the delay period (in seconds) a failure must be active before the SROS will enter backup operation on the interface (Range: 10 to 86,400 seconds) |

Default Values

By default, the backup delay period is set to 10 seconds.

Usage Examples

The following example configures the SROS to wait 60 seconds (on an endpoint with an active alarm condition) before attempting backup operation:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup backup-delay 60
**backup call-mode** *<role>*

Use the `backup call-mode` command to specify whether the configured backup interface answers or originates (or a combination of both) backup calls. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th><em>&lt;role&gt;</em></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer</td>
<td>Answers and backs up primary link on failure.</td>
</tr>
<tr>
<td>answer-always</td>
<td>Answers and backs up regardless of primary link state.</td>
</tr>
<tr>
<td>originate</td>
<td>Originates backup call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer</td>
<td>Originates or answers call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer-always</td>
<td>Originates on failure; answers and backs up always.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, `backup call-mode` is set to `originate-answer`.

**Functional Notes**

The majority of the configuration for the SROS backup implementation is configured via the backup PPP interface configuration commands. However, the numbers dialed are configured in the primary interface. Full sample configurations follow:

**Sample config for remote router (dialing out)**

```
hostname "Remote7203dl"
enable password password
!
interface eth 0/1
  ip address 192.168.1.254 255.255.255.0
  no shutdown
!
interface modem 1/3
  no shutdown
!
interface t1 1/1
  coding b8zs
  framing esf
  clock source line
  tdm-group 1 timeslots 1-24
  no shutdown
!
```
interface fr 1 point-to-point
  frame-relay lmi-type ansi
  no shutdown
  bind 1 t1 1/1 1 fr 1

interface fr 1.16 point-to-point
  frame-relay interface-dlci 16
  ip address 10.1.1.2 255.255.255.252
  backup call-mode originate
  backup number 5551111 analog ppp 1
  backup number 5552222 analog ppp 1
  no shutdown

interface ppp 1
  ip address 172.22.56.1 255.255.255.252
  ppp authentication chap
  username remoterouter password remotepass
  ppp chap hostname localrouter
  ppp chap password procurve
  no shutdown

ip route 192.168.2.0 255.255.255.0 172.22.56.2 255.255.255.252

line telnet 0 4
  password password

Sample config for central router (dialing in)

hostname "Central7203dl"
enable password password

interface eth 0/1
  ip address 192.168.100.254 255.255.255.0
  no shutdown

interface modem 1/3
  no shutdown

interface t1 1/1
  coding b8zs
  framing esf
  clock source line
tdm-group 1 timeslots 1-24
no shutdown
!
interface fr 1 point-to-point
  frame-relay lmi-type ansi
  no shutdown
  bind 1 t1 1/1 1 fr 1
!
interface fr 1.100 point-to-point
  frame-relay interface-dlci 100
  ip address 10.1.1.1 255.255.255.252
  backup call-mode answer
  backup number 555-8888 analog ppp 1
!
interface ppp 1
  ip address 172.22.56.2 255.255.255.252
  ppp authentication chap
  username localrouter password procurve
  ppp chap hostname remoterouter
  ppp chap password remotepass
  no shutdown
!
ip route 192.168.1.0 255.255.255.0 172.22.56.1 255.255.255.252

line telnet 0 4
  password password

Usage Examples
The following example configures the SROS to generate backup calls for this endpoint using an analog
modem interface (to phone number 555-1111) but never answer calls, and specifies ppp 2 as the backup
interface:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup call-mode originate
ProCurve(config-atm 1.1)#backup number 5551111 analog ppp 2

Technology Review
This technology review provides information regarding specific backup router behavior (for example, when the
router will perform backup, where in the configuration the SROS accesses specific routing information, etc.):
Dialing Out
1. The SROS determines to place an outbound call when either Layer 1 or Layer 2 has a failure.
2. When placing outbound calls, the SROS matches the number dialed to a PPP interface.
3. When placing the call, the SROS uses the configuration of the related PPP interface for authentication and IP negotiation.
4. If the call fails to connect on the first number dialed, the SROS places a call to the second number (if a second number is configured). The second number to be dialed references a separate PPP interface.

Dialing In
1. The SROS receives an inbound call on a physical interface.
2. Caller ID is used to match the backup number command to the configured PPP interface.
3. If a match is found, the call connects and the SROS pulls down the primary connection if it is not already in a down state.
4. If no match is found from Caller ID, the call is terminated.
**backup connect-timeout <seconds>**

Use the `backup connect-timeout` command to specify the number of seconds to wait for a connection after a call is attempted before trying to call again or dialing a different number. It is recommended this number be greater than 60.

**Syntax Description**

| <seconds> | Selects the amount of time in seconds that the router will wait for a connection before attempting another call (valid range: 10 to 300). |

**Default Values**

By default, the backup connect-timeout period is set to 60 seconds.

**Usage Examples**

The following configures the SROS to wait 120 seconds before retrying a failed backup call:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup connect-timeout 120
```
**backup force [backup | primary]**

Use the `backup force` command to manually override the automatic backup feature. This can be used to force a link into backup to allow maintenance to be performed on the primary link without disrupting data. Use the `no` form of this command to return to the normal backup operation state.

**Syntax Description**

- **backup**
  - Forces backup regardless of primary link state.
- **primary**
  - Forces primary link regardless of its state.

**Default Values**

By default, this feature is disabled.

**Usage Examples**

The following configures the SROS to force this endpoint into backup:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup force backup
```
backup maximum-retry <attempts>

Use the backup maximum-retry command to select the number of calls the router will make when attempting to backup a link. Use the no form of this command to return to the default state. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 672.

Syntax Description

| <attempts> | Selects the number of call retries that will be made after a link failure (valid range is 0 to 15). Setting this value to 0 will allow unlimited retries during the time the network is failed. |

Default Values

By default, backup maximum-retry is set to 0 attempts.

Usage Examples

The following example configures the SROS to retry a dial-backup call four times before considering backup operation unavailable:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup maximum-retry 4
Use the `backup number` command to configure the phone number and the call type the router will dial upon network failure. Multiple entries can be made for an interface to allow alternate sites to be dialed. For more detailed information on backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 672.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;digits&gt;</code></td>
<td>Specifies the phone number to call when the backup is initiated.</td>
</tr>
<tr>
<td>analog</td>
<td>Indicates number connects to an analog modem.</td>
</tr>
<tr>
<td>digital-56k</td>
<td>Indicates number belongs to a digital 56 kbps per DS0 connection.</td>
</tr>
<tr>
<td>digital-64k</td>
<td>Indicates number belongs to a digital 64 kbps per DS0 connection.</td>
</tr>
<tr>
<td><code>&lt;isdn min chan&gt;</code></td>
<td>Specifies the minimum number of DS0s required for a digital 56 or 64 kbps connection (Range: 1 to 24).</td>
</tr>
<tr>
<td><code>&lt;isdn max chan&gt;</code></td>
<td>Specifies the maximum number of DS0s desired for a digital 56 or 64 kbps connection (Range: 1 to 24).</td>
</tr>
<tr>
<td>ppp <code>&lt;interface&gt;</code></td>
<td>Specifies the PPP interface to use as the backup for this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no configured backup numbers.

**Usage Examples**

The following example configures the SROS to dial 704-555-1212 (digital 64 kbps connection) to initiate backup operation for this endpoint using the configured `ppp 1` backup interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup number 7045551212 digital-64k 1 1 ppp 1
```
backup priority <value>

Use the backup priority command to select the backup priority for this interface. This allows the user to establish the highest priority backup link and ensure that link will override backups attempted by lower priority links. Use the no form of this command to return to the default value. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 672.

**Syntax Description**

| <value> | Sets the relative priority of this link (valid range: 0 to 100). A value of 100 designates the highest priority. |

**Default Values**

By default, backup priority is set to 50.

**Usage Examples**

The following example assigns the highest priority to this endpoint:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup priority 100
backup randomize-timers

Use the **backup randomize-timers** command to randomize the call timers to minimize potential contention for resources. Use the **no** form of this command to return to the default value. For more detailed information on backup functionality, refer to the *Functional Notes* section of the command *backup call-mode <role>* on page 672.

**Syntax Description**

No subcommands.

**Default Values**

By default, the SROS does not randomize the backup call timers.

**Usage Examples**

The following example configures the SROS to randomize the backup timers associated with this endpoint:

```plaintext
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup randomize-timers
```
backup redial-delay <seconds>

Use the backup redial-delay command to configure the delay after an unsuccessful call until the call will be re-tryed. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 672.

Syntax Description

| <seconds>        | Specifies the delay in seconds between attempting to re-dial a failed backup attempt. Range: 10 to 3600. |

Default Values

By default, backup redial-delay is set to 10 seconds.

Usage Examples

The following example configures a redial delay of 25 seconds on this endpoint:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup redial-delay 25
backup restore-delay <seconds>

Use the `backup restore-delay` command to configure the amount of time the router will wait after the network is restored before disconnecting the backup link and reverting to the primary. This setting is used to prevent disconnecting the backup link if the primary link is “bouncing” in and out of alarm. For more detailed information on backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 672.

**Syntax Description**

- `<seconds>`: Specifies the number of seconds the SROS will wait (after a primary link is restored) before disconnecting backup operation. Range: 10 to 86,400.

**Default Values**

By default, `backup restore-delay` is set to 10 seconds.

**Usage Examples**

The following example configures the SROS to wait 30 seconds before disconnecting backup operation and restoring the primary connection for this endpoint:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup restore-delay 30
```
backup schedule [day | enable-time | disable-time]

Use the `backup schedule` command to set the time of day that backup will be enabled. Use this command if backup is desired only during normal business hours and on specific days of the week. Use the `no` form of this command to disable backup (as specified). For more detailed information on backup functionality, refer to the Functional Notes section of the command `backup call-mode <role>` on page 672.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>Sets the days to allow backup (valid range: Monday through Sunday).</td>
</tr>
<tr>
<td>enable-time</td>
<td>Sets the time of day to enable backup. Time is entered in 24-hour format (00:00).</td>
</tr>
<tr>
<td>disable-time</td>
<td>Sets the time of day to disable backup.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, backup is enabled for all days and times if the `backup auto-backup` command has been issued and the backup schedule has not been entered.

**Usage Examples**

The following example enables backup Monday through Friday 8:00 am to 7:00 pm:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup schedule enable-time 08:00
ProCurve(config-atm 1.1)#backup schedule disable-time 19:00
ProCurve(config-atm 1.1)#no backup schedule day Saturday
ProCurve(config-atm 1.1)#no backup schedule day Sunday
```
backup shutdown

Use the **backup shutdown** command to deactivate all backup functionality in the unit. Backup configuration parameters are kept intact, but the unit will not initiate (or respond) to backup sequences in the event of a network outage. Use the **no** form of this command to reactivate the backup interface. For more detailed information on backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 672.

**Syntax Description**

No subcommands.

**Default Values**

By default, all SROS backup interfaces are disabled.

**Usage Examples**

The following example deactivates the configured backup interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#backup shutdown
```
**bandwidth <value>**

Use the `bandwidth` command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the `no` form of this command to restore the default values.

**Syntax Description**

| <value> | Enter bandwidth in kbps. |

**Default Values**

To view default values, use the `show interfaces` command.

**Functional Notes**

The `bandwidth` command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the ATM sub-interface to 10 Mbps:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#bandwidth 10000
```
**bridge-group** `<group#>`

Use the `bridge-group` command to assign an interface to the specified bridge group. Use the `no` form of this command to remove the interface from the bridge group.

**Syntax Description**

| `<group#>` | Assigns a bridge group number to the interface (range is 1 to 255). |

**Default Values**

By default, there are no configured bridge groups.

**Functional Notes**

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (Ethernet to T1 bridge, Ethernet to Frame Relay sub-interface).

**Usage Examples**

The following example assigns the ATM sub-interface labeled 1.1 to bridge group 1:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#bridge-group 1
```
crypto map <mapname>

Use the crypto map command to associate crypto maps with the interface.

**Note**

When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the VPN Configuration Guide located on the ProCurve SROS Documentation CD provided with your unit.

**Syntax Description**

<mapname> Specifies the crypto map name that you wish to assign to the interface.

**Default Values**

By default, no crypto maps are assigned to an interface.

**Functional Notes**

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the unencrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the previous diagram, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only unencrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy class on an interface. When specifying the ACLs for a crypto map, the source information is the private local side, unencrypted source of the data. The destination information will be the far end, unencrypted destination of the data. However, ACLs for a policy class work in reverse. The source information for the ACL in a policy class is the far end. The destination information is the local side.

**Usage Examples**

The following example applies all crypto maps with the name *MyMap* to the ATM sub-interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#crypto map MyMap
```
**dynamic-dns**

Use the `dynamic-dns` command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the `no` versions of these commands to disable these features. Variations of this command include:

- `dynamic-dns custom <hostname> <minutes>
- `dynamic-dns dyndns <hostname> <username> <password>
- `dynamic-dns dyndns-custom <hostname> <username> <password>
- `dynamic-dns dyndns-static <hostname> <username> <password>

**Syntax Description**

- `<hostname>` Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).
- `<minutes>` Specifies the intervals in minutes to update the server with information (updates also occur when the interface’s IP address changes, regardless of the update intervals).

Refer to *Functional Notes* below for additional argument descriptions.

**Default Values**

No default is necessary for this command.

**Functional Notes**

- `custom` - Constanttime.com’s Custom Dynamic DNS service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows you to manage IP address mappings (A records), domain aliases (CNAME records), and mail servers (MX records).

- `dyndns` - The Dynamic DNS service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- `dyndns-custom` - DynDNS.org’s Custom DNS service provides a full DNS solution, giving you complete control over an entire domain name. A web-based interface provides two levels of control over your domain, catering to average or power users. Five globally redundant DNS servers ensure that your domain will always resolve.

A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background and provides layout and functionality similar to a BIND zone file, allowing for the creation of nearly any record type.

Custom DNS can be used with both static and dynamic IPs and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
**dyndns-static** - The Static DNS service is similar to Dynamic DNS service in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate through the DNS system. This service is provided for up to five hostnames.

If your IP address doesn't change often or at all but you still want an easy name to remember it by (without having to purchase your own domain name), Static DNS service is ideal for you.

If you would like to use your own domain name (such as yourname.com), you need Custom DNS service, which also provides full dynamic and static IP address support.

**Usage Examples**

The following example sets the dynamic-dns to dyndns-custom with hostname host, username user, and password pass:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#dynamic-dns dyndns-custom host user pass
```
encapsulation [aal5mux ip | aal5mux ppp | aal5snap]

Use the `encapsulation` command to configure the encapsulation type for the ATM Adaptation Layer (AAL) of the ATM Protocol Reference model. Use the `no` form of this command to return to the default setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aal5mux ip</code></td>
<td>Specifies encapsulation type for multiplexed virtual circuits using the IP protocol.</td>
</tr>
<tr>
<td><code>aal5mux ppp</code></td>
<td>Specifies encapsulation type for multiplexed virtual circuits using the Point-to-Point (PPP) protocol.</td>
</tr>
<tr>
<td><code>aal5snap</code></td>
<td>Specifies encapsulation type that supports LLC/SNAP protocols.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the encapsulation type is `aal5snap`.

**Functional Notes**

For PPP and PPPoE, the encapsulation type can be `aal5snap` or `aal5mux ppp`.  
For IP with no bridging, the encapsulation type can be `aal5snap` or `aal5mux ip`.  
For IP with bridging, the encapsulation type can only be `aal5snap`  
For bridging, the encapsulation type can only be `aal5snap`.

**Usage Examples**

The following example sets the encapsulation type to `aal5snap`:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#encapsulation aal5snap
```
**fair-queue <threshold>**

Use the `fair-queue` command to enable weighted fair queuing (WFQ) on an interface. Use the `no` form of this command to disable WFQ and enable first-in-first-out (FIFO) queueing for an interface. WFQ is enabled by default for WAN interfaces.

**Syntax Description**

| `<threshold>` | Optional. Specifies the maximum number of packets that can be present in each conversation sub-queue. Packets received for a conversation after this limit is reached are discarded. Range: 16 to 512. |

**Default Values**

By default, `fair-queue` is enabled with a threshold of 64 packets.

**Usage Examples**

The following example enables WFQ on the interface with a threshold set at 100 packets:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#fair-queue 100
```
**hold-queue** `<queue size>` **out**

Use the `hold-queue out` command to change the overall size of an interface's WAN output queue. Use the `no` form of this command to return to the default setting.

**Syntax Description**

| `<queue size>` | Specifies the total number of packets the output queue can contain before packets are dropped. Range: 16 to 1000. |

**Default Values**

The default queue size for WFQ is 400. The default queue size for PPP FIFO and Frame Relay round-robin is 200.

**Usage Examples**

The following example sets the overall output queue size to 700:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#hold-queue 700 out
```
ip access-group <listname> [in | out]

Use the ip access-group command to create an access list to be used for packets transmitted on or received from the specified interface. Use the no form of this command to disable this type of control.

Syntax Description

<table>
<thead>
<tr>
<th>listname</th>
<th>Specifies the assigned IP access list name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>Enables access control on packets received on the specified interface.</td>
</tr>
<tr>
<td>out</td>
<td>Enables access control on packets transmitted on the specified interface.</td>
</tr>
</tbody>
</table>

Default Values

By default, these commands are disabled.

Functional Notes

When this command is enabled, the IP destination address of each packet must be validated before being passed through to the router system. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the router to only allow Telnet traffic into the ATM sub-interface:

ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#interface atm 1.1
ProCurve(config-atm 1.1)#ip access-group TelnetOnly in
ip address dhcp

Use the `ip address dhcp` command to use Dynamic Host Configuration Protocol (DHCP) to obtain an address on the interface. Use the `no` form of this command to remove a configured IP address (using DHCP) and disable DHCP operation on the interface.

```
ip address dhcp [client-id [<interface> | <identifier>] hostname <"string"> ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-id</td>
<td>Optional. Specifies the client identifier used when obtaining an IP address from a DHCP server.</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>Specifying an interface defines the client identifier as the hexadecimal MAC address of the specified interface (including a hexadecimal number added to the front of the MAC address to identify the media type). For example, specifying the <code>client-id ethernet 0/1</code> (where the Ethernet interface has a MAC address of 0012.7991.1150) defines the client identifier as <code>01:00:12:79:91:11:50</code> (where 01 defines the media type as Ethernet). Refer to <code>hardware-address &lt;hardware-address&gt; &lt;type&gt;</code> on page 1287 for a detailed listing of media types.</td>
</tr>
<tr>
<td>&lt;identifier&gt;</td>
<td>Specifies a custom client-identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters). For example, a custom client identifier of <code>0f:ff:ff:ff:ff:51:04:99:a1</code> may be entered using the <code>&lt;identifier&gt;</code> option.</td>
</tr>
<tr>
<td>hostname</td>
<td>Specifies a text string (to override the global router name) to use as the name in the DHCP option 12 field.</td>
</tr>
<tr>
<td>&lt;&quot;string&quot;&gt;</td>
<td>String (encased in quotation marks) of up to 35 characters to use as the name of the host for DHCP operation.</td>
</tr>
<tr>
<td>no-default-route</td>
<td>Keyword used to specify that the SROS not install the default-route obtained via DHCP.</td>
</tr>
<tr>
<td>no-domain-name</td>
<td>Keyword used to specify that the SROS not install the domain-name obtained via DHCP.</td>
</tr>
<tr>
<td>no-nameservers</td>
<td>Keyword used to specify that the SROS not install the DNS servers obtained via DHCP.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| client-id  | Optional. By default, the client identifier is populated using the following formula: TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS  

Where TYPE specifies the media type in the form of one hexadecimal byte (refer to `hardware-address <hardware-address> <type>` on page 1287 for a detailed listing of media types) and the MAC ADDRESS is the Media Access Control (MAC) address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to Ethernet 0/1 is used in this field). |
hostname <"string">  By default, the hostname is the name configured using the Global Configuration hostname command.

Functional Notes
Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the client-id and hostname fields.

Usage Examples
The following example enables DHCP operation on the ATM sub-interface 1.1:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip address dhcp
**ip address <address> <mask> [secondary]**

Use the `ip address` command to define an IP address on the specified interface. Use the optional `secondary` keyword to define a secondary IP address. Use the `no` form of this command to remove a configured IP address.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).</td>
</tr>
<tr>
<td>&lt;mask&gt;</td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
<tr>
<td>secondary</td>
<td>Optional. Defines a secondary IP address for the specified interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no assigned IP addresses.

**Functional Notes**

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

**Usage Examples**

The following example configures a secondary IP address of 192.168.72.101/30:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip address 192.168.72.101 255.255.255.252 secondary

To view configured secondary IP address(es), use the `show running-config` command. Secondary IP address information is not available through other `show` commands.
ip dhcp [release | renew]

Use the ip dhcp command to release or renew the DHCP IP address. This command is only applicable when using DHCP for IP address assignment.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>release</td>
<td>Releases DHCP IP address.</td>
</tr>
<tr>
<td>renew</td>
<td>Renews DHCP IP address.</td>
</tr>
</tbody>
</table>

Default Values

No default values required for this command.

Usage Examples

The following example releases the IP DHCP address for the ATM sub-interface 1.1:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip dhcp release
ip helper-address <address>

Use the `ip helper-address` command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the `no` form of this command to disable forwarding packets.

**Syntax Description**

| `<address>` | Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets. |

**Default Values**

By default, broadcast UDP packets are not forwarded.

**Functional Notes**

When used in conjunction with the `ip forward-protocol` command, the `ip helper-address` feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the `ip forward-protocol` command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the `ip helper-address` feature:

1. The packet IP protocol is UDP.
2. Any UDP port specified using the `ip forward-protocol` command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

**Usage Examples**

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.168.5.99:

```
ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip helper-address 192.168.5.99
```
ip igmp

Use the `ip igmp` command to configure multicasting-related functions for the interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>immediate-leave</code></td>
<td>Specifies that when only one host (or IGMP snooping switch) is connected to the interface a leave is received, multicast of that group is immediately terminated instead of sending a group query when a leave is received and timing out the group if no device responds. Works in conjunction with <code>ip igmp last-member-query-interval</code>. Applies to all groups when configured.</td>
</tr>
<tr>
<td><code>last-member-query-interval &lt;milliseconds&gt;</code></td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group's multicast packets to the interface. Range: 100 to 65535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td><code>querier-timeout &lt;seconds&gt;</code></td>
<td>Specifies the number of seconds that the router waits after the current querier's last query before it takes over as querier (IGMP V2). Range: 60-300 seconds. Default: 2x the <code>query-interval</code> value.</td>
</tr>
<tr>
<td><code>query-interval &lt;seconds&gt;</code></td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td><code>query-max-response-time &lt;seconds&gt;</code></td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td><code>static-group &lt;group-address&gt;</code></td>
<td>Configures the router's interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>`version [1</td>
<td>2]`</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example sets the query message interval on the interface to 200 milliseconds:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip igmp last-member-query-interval 200
ip mcast-stub downstream

Use the `ip mcast-stub downstream` command to enable multicast forwarding and IGMP (router mode) on an interface and place it in multicast stub downstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub upstream` commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub upstream` on page 705 for more information.

**Usage Examples**

The following example enables multicast forwarding and IGMP on the interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip mcast-stub downstream
```
**ip mcast-stub fixed**

Use the **ip mcast-stub fixed** command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the **no** form of this command to disable this mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Multicast routing must be enabled prior to setting **ip mcast-stub fixed** on the selected interface. Also, use the **ip igmp static-group <A.B.C.D>** command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

**Usage Examples**

The following example enables multicast traffic forwarding and IGMP on the interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip mcast-stub fixed
```
ip mcast-stub helper-enable

Use the `ip mcast-stub helper-enable` command to assign the `ip mcast-stub helper-address` as the IGMP proxy. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address`, `ip mcast-stub upstream`, and `ip mcast-stub downstream` commands. When enabled, the interface becomes a helper forwarding interface. The IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Refer to `ip mcast-stub helper-address <ip address>` on page 397, `ip mcast-stub downstream` on page 702, and `ip mcast-stub upstream` on page 705 for more information.

**Usage Examples**

The following example sets the helper address as the IGMP proxy:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip mcast-stub helper-enable
```
ip mcast-stub upstream

Use the `ip mcast-stub upstream` command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub downstream` commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router's unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub downstream` on page 702 for more information.

**Usage Examples**

The following example enables multicast forwarding on the interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip mcast-stub upstream
```
**ip ospf**

Use the `ip ospf` command to customize OSPF settings (if needed).

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>authentication-key</strong></td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td><strong>cost &lt;value&gt;</strong></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td><strong>dead-interval &lt;seconds&gt;</strong></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32767.</td>
</tr>
<tr>
<td><strong>hello-interval &lt;seconds&gt;</strong></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>message-digest-key &lt;keyid&gt; md5 &lt;key&gt;</strong></td>
<td>Configures OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td><strong>priority &lt;value&gt;</strong></td>
<td>Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td><strong>retransmit-interval &lt;seconds&gt;</strong></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>transmit-delay &lt;seconds&gt;</strong></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32,767.</td>
</tr>
</tbody>
</table>

**Default Values**

- `retransmit-interval <seconds>`: 5 seconds
- `transmit-delay <seconds>`: 1 second
- `hello-interval <seconds>`: 10 seconds: Ethernet, point-to-point, Frame Relay, and PPP
- `dead-interval <seconds>`: 40 seconds

**Usage Example**

The following example sets the maximum number of seconds allowed between hello packets to 25,000:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip ospf dead-interval 25000
```
ip ospf authentication [message-digest | null]

Use the `ip ospf authentication` command to authenticate an interface that is performing OSPF authentication.

**Syntax Description**

- **message-digest**
  - Specifies message-digest authentication type.

- **null**
  - Specifies that no authentication be used.

**Default Values**

By default, this is set to null (meaning no authentication is used).

**Usage Examples**

The following example specifies that no authentication will be used on the ATM sub-interface 1.1:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip ospf authentication null
```
**ip ospf network [broadcast | point-to-point]**

Use the `ip ospf network` command to specify the type of network on this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcast</td>
<td>Sets the network type for broadcast.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Sets the network type for point-to-point.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

**Functional Notes**

A point-to-point network will not elect designated routers.

**Usage Examples**

The following example designates a broadcast network type:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip ospf network broadcast
```
**ip pim sparse-mode**

Use the *ip pim sparse-mode* command to enable PIM Sparse Mode on the interface. Use the *no* form of this command to disable PIM Sparse Mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, PIM Sparse Mode is disabled for all interfaces.

**Functional Notes**

PIM Sparse Mode is a multicast routing protocol that makes use of the unicast forwarding table. PIM-systems builds unidirectional shared trees rooted at a Rendezvous Point (RP) for a multicast group or a shortest-path tree rooted at a specific source for a multicast group.

**Usage Examples**

The following example enables PIM sparse-mode on the ATM interface 1.1:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip pim sparse-mode
ip pim-sparse dr-priority <value>

Use the ip pim-sparse dr-priority command to specify the priority of this PIM interface for use when selecting the designated router (DR). Use the no form of this command to return to the default value.

Syntax Description

<value> Specifies the priority of this interface (to be used when determining the DR). Valid range is 1 to 4,294,967,295.

Default Values

By default, the priority of all PIM interfaces is 1.

Functional Notes

Interfaces advertise their configured priority values in the hello messages transmitted on the interface. Routers use the priority values to determine the appropriate DR. The router on the network segment with the highest priority is selected as the DR. If a hello message is received on the interface from a router on the network segment and it does not contain a priority, the entire network segment defaults to DR selection based on IP addresses instead of priority. In this instance, the DR is selected as the router on the network segment that has the highest IP address. ProCurve Secure Routers will always include a priority in all transmitted hello messages. If no priority is specifically designated by the user, the priority is set as the default of 1.

Usage Examples

The following example specifies a priority of 100 on the ATM interface 1.1:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip pim-sparse dr-priority 100
ip pim-sparse hello-timer <time>

Use the \texttt{ip pim-sparse hello-timer} command to set the time interval at which periodic hello messages are transmitted out the interface. Each PIM interface has an independent hello-timer. Use the \texttt{no} form of this command to return to the default value.

\textbf{Syntax Description}

\begin{tabular}{|l|}
\hline
\texttt{<time>} & Specifies the interval (in seconds) at which periodic hellos are sent out the interface. Valid range is 10 to 3600 seconds. \\
\hline
\end{tabular}

\textbf{Default Values}

By default, hellos are transmitted on PIM interfaces every 60 seconds.

\textbf{Functional Notes}

Hello messages are used to inform neighbors of a router's presence. Hello messages normally generate a small amount of traffic on an interface. Setting the \texttt{hello-timer} to a small interval increases the amount of hellos sent (thus increasing the amount of traffic). Set the \texttt{hello-timer} to a reasonable value, taking into consideration the bandwidth available on the interface.

\textbf{Usage Examples}

The following example specifies hellos to be sent on the ATM interface every 3600 seconds:

\begin{verbatim}
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip pim-sparse hello-timer 3600
\end{verbatim}
ip pim-sparse nbr-timeout <time>

Use the `ip pim-sparse nbr-timeout` command to specify the interval the PIM interface waits before declaring that the neighbor is not present. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the time interval (in seconds) the PIM interface waits before a neighbor is considered not present. Valid range is 30 to 10,800 seconds. |

**Default Values**

By default, the `nbr-timeout` is set to 105 seconds.

**Usage Examples**

The following example specifies a wait interval of 360 seconds on the ATM interface 1.1:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip pim-sparse nbr-timeout 360
```
ip pim-sparse override-interval <time>

Use the `ip pim-sparse override-interval` command to specify the delay interval after a join/prune in which another router on the LAN may override the join/prune. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the delay interval (in milliseconds) after a join/prune in which another router on the LAN may override the join/prune. Valid range is 0 to 65,535 milliseconds. |

**Default Values**

By default, the `override-interval` is set to 2500 milliseconds.

**Usage Examples**

The following example sets the delay interval for join/prune overrides to 3000 milliseconds on the ATM interface `1.1`:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip pim-sparse override-interval 3000
```
ip pim-sparse propagation-delay <time>

Use the ip pim-sparse propagation-delay command to specify the expected propagation delay for join/prune messages. Set the propagation delay (in milliseconds) to estimate the amount of delay found in the local link. Use the no form of this command to return to the default value.

Syntax Description

<time> Specifies the expected propagation delay in the local link in milliseconds. Valid range is 0 to 32,767 milliseconds.

Default Values

By default, the propagation-delay is set to 500 milliseconds.

Usage Examples

The following example sets the expected propagation delay to 300 milliseconds on the ATM interface 1.1:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm1.1)#ip pim-sparse propagation-delay 300
ip policy route-map <mapname>

Use the ip policy route-map command to assign a policy route map to this interface. Use the no form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

Syntax Description

<mapname> Specifies the route map to associate with this interface.

Default Values

By default, policy-based routing is disabled for all interfaces.

Usage Examples

The following example associates the route map named MyMap with ATM interface 1.1:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip policy route-map MyMap
**ip proxy-arp** `<address> <subnet mask>`

Use the `ip proxy-arp` to enable proxy Address Resolution Protocol (ARP) on the interface. Use the `no` form of this command to disable this feature.

**Syntax Description**

| `<address>` | Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101). |
| `<subnet mask>` | Specifies the subnet mask that corresponds to the listed IP address. |

**Default Values**

By default, proxy ARP is enabled.

**Functional Notes**

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all proxy ARP requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

**Usage Examples**

The following enables proxy ARP on the ATM sub-interface 1.1:

```plaintext
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip proxy-arp
```
ip rip receive version [1 | 2]

Use the `ip rip receive version` command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accepts only RIP version 1 packets received on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Accepts only RIP version 2 packets received on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip receive version` to specify a RIP version that will override the `version` (in the Router RIP) configuration.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the ATM sub-interface 1.1 to accept only RIP version 2 packets:

```sh
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip rip receive version 2
```
ip rip send version [1 | 2]

Use the `ip rip send version` command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transmits only RIP version 1 packets on the interface</td>
</tr>
<tr>
<td>2</td>
<td>Transmits only RIP version 2 packets on the interface</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces transmit RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip send version` to specify a RIP version that will override the `version` (in the Router RIP) configuration.

The SROS only transmits one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the ATM sub-interface 1.1 to transmit only RIP version 2 packets:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip rip send version 2
```
**ip rip summary-address** `<network address> <network mask>`

Use the `ip rip summary-address` command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the `no` form of this command to disable this mode.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network address&gt;</code></td>
<td>Specifies the IP address of the network to be summarized.</td>
</tr>
<tr>
<td><code>&lt;network mask&gt;</code></td>
<td>Specifies the network mask to be applied to the specific network to be summarized.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no manual summarization is applied by RIP.

**Functional Notes**

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the `ip rip summary-address` command. This command is only effective if RIP version 2 is configured.

**Usage Examples**

The following example enables manual summarization on the specified IP address:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip rip summary-address 10.10.123.0 255.255.255.0
```
**ip route-cache**

Use the `ip route-cache` command to enable route caching on the interface. Use the `no` form of this command to disable route caching and return to process switching mode.

**Note**  
Using Network Address Translation (NAT) or the SROS firewall capabilities on an interface requires process switching mode (using the `no ip route-cache` command).

**Syntax Description**
No subcommands.

**Default Values**
By default, route caching is enabled on all interfaces.

**Functional Notes**
Route caching allows an IP interface to provide optimum performance when processing IP traffic.

**Usage Examples**
The following example enables route caching on the ATM sub-interface 1.1:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip route-cache
```
ip unnumbered <interface>

Use the ip unnumbered command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the no form of this command to remove the unnumbered configuration.

Syntax Description

| <interface> | Specifies the interface (in the format type slot/port) that contains the IP address to use as the source address for all packets transmitted on this interface. |

Default Values

By default, all interfaces are configured to use a specified IP address (using the ip address command).

Functional Notes

If ip unnumbered is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying ip unnumbered eth 0/1 while in the ATM Sub-Interface Configuration mode configures the ATM sub-interface to use the IP address assigned to the Ethernet interface for all IP processing. In addition, the SROS uses the specified interface information when sending route updates over the unnumbered interface.

Usage Examples

The following example configures the ATM sub-interface 1.1 to use the IP address assigned to the Ethernet interface (eth 0/1):

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#ip unnumbered eth 0/1
**max-reserved-bandwidth <percent>**

Use the `max-reserved-bandwidth` command to specify the percentage of interface bandwidth reserved for use in user-defined (priority or class-based) queues. The remainder of the interface bandwidth is reserved for system critical traffic and is not available to user-defined queues. Use the `no` form of this command to restore the default value.

**Caution**

Reserving a portion of the interface bandwidth for system critical traffic is necessary for proper operation. Specifying the entire interface bandwidth for use in user-defined queues can cause undesirable operation.

**Syntax Description**

| <percent> | Specifies the percentage of interface bandwidth to make available for user-defined (priority or class-based) queues. Enter an integer 1 to 100. |

**Default Values**

By default, `max-reserved-bandwidth` is set to **75** which reserves 25 percent of the interface bandwidth for system critical traffic.

**Usage Examples**

The following example specifies 85 percent of the bandwidth on the `atm 1.1` interface to be available for use in user-defined queues:

```
(config)#interface atm 1.1
(config-atm 1.1)#max-reserved-bandwidth 85
```
**mtu <size>**

Use the `mtu` command to configure the maximum transmit unit (MTU) size for the active interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

```
<size>  
```

Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:

- ATM interfaces: 64 to 1520
- Demand interfaces: 64 to 1520
- Ethernet interfaces: 64 to 1500
- FDL interfaces: 64 to 256
- HDLC interfaces: 64 to 1520
- Loopback interfaces: 64 to 1500
- Tunnel interfaces: 64 to 18,190
- Virtual Frame Relay sub-interfaces: 64 to 1520
- Virtual PPP interfaces: 64 to 1500

**Default Values**

```
<size>  
```

The default values for the various interfaces are listed below:

- ATM interfaces: 1500
- Demand interfaces: 1500
- Ethernet interfaces: 1500
- FDL interfaces: 256
- HDLC interfaces: 1500
- Loopback interfaces: 1500
- Tunnel interfaces: 1500
- Virtual Frame Relay sub-interfaces: 1500
- Virtual PPP interfaces: 1500

**Functional Notes**

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

**Usage Examples**

The following example specifies an MTU of 1200 on the ATM sub-interface 1.1:

`ProCurve(config)#interface atm 1.1`  
`ProCurve(config-atm 1.1)#mtu 1200`
oam-pvc managed <frequency>

Use the oam-pvc managed command to enable end-to-end F5 Operation, Administration, and Maintenance (OAM) loopback cell generation and OAM management for an ATM interface. Use the no form of this command to disable generation of OAM loopback cells.

Syntax Description>

| <frequency> | Specifies the time delay between transmitting OAM loopback cells. The range is from 0 to 600 seconds. |

Default Values

By default, the frequency is 1 second.

Usage Examples

The following example enables OAM loopback cell generation with a frequency of 5 seconds:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#oam-pvc manage 5
oam retry <up-count> <down-count> <retry-frequency>

Use the `oam retry` command to configure parameters related to Operation, Administration, and Maintenance (OAM) management for an ATM interface. Use the `no` form of this command to disable OAM management parameters.

**Syntax Description**

- `<up-count>`: Specifies the number of consecutive end-to-end F5 OAM loopback cell responses that must be received in order to change a PVC connection state to up. The range is from 1 to 255.
- `<down-count>`: Specifies the number of consecutive end-to-end F5 OAM loopback cell responses that are not received in order to change a PVC state to down. The range is from 1 to 255.
- `<retry-frequency>`: Specifies the frequency (in seconds) that end-to-end F5 OAM loopback cells are transmitted when a change in the up/down state of a PVC is being verified. The range is from 1 to 600.

**Default Values**

By default, the up-count is set to 3, the down-count is set to 5, and the retry frequency is 1.

**Usage Examples**

The following example configures the OAM parameters with an up-count of 2, down-count of 2, and retry frequency of 10:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#oam retry 2 2 10
```
pvc <VPI/VCI>

Use the pvc command to select the ATM virtual link for this sub-interface. Use the no form of this command to remove the link.

**Syntax Description**

| <VPI/VCI> | Specifies the ATM network virtual path identifier (VPI) for this PVC and the ATM network virtual path identifier (VPI) for this PVC. The VPI value range is 0 to 255, and the VCI value range is 32 to 65,535. |

**Default Values**

No default value is necessary for this command.

**Usage Examples**

The following example sets the VPI to 8 and the VCI to 35:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#pvc 8/35
```
**qos-policy out <mapname>**

Use the `qos-policy out` command to apply a previously-configured QoS map to an interface. Use the **no** form of this command to remove the map from the interface. The keyword **out** specifies that this policy will be applied to outgoing packets.

**Syntax Description**

```plaintext
<mapname> Specifies the name of a previously-created QoS map (refer to `qos map <mapname> <sequence number>` on page 451 for more information).
```

**Default Values**

No default value is necessary for this command.

**Usage Examples**

The following example applies the QoS map **VOICEMAP** to the ATM sub-interface 1.1:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#qos-policy out VOICEMAP
```
spanning-tree bpdufilter [enable | disable]

Use the `spanning-tree bpdufilter` command to block BPDUs from being transmitted and received on this interface. To return to the default value, use the `no` form of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>enable</th>
<th>Enables the BPDU filter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>disable</td>
<td>Disables the BPDU filter.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is set to `disable`.

**Functional Notes**

The purpose of this command is to remove a port from participation in the spanning tree. This might be beneficial while debugging a network setup. It normally should not be used in a live network.

**Usage Examples**

The following example enables the BPDU filter on the interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#spanning-tree bpdufilter enable
```
spanning-tree bpduguard [enable | disable]

Use the spanning-tree bpduguard command to block BPDUs from being received on this interface. To return to the default value, use the no form of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables the BPDU block.</td>
</tr>
<tr>
<td>disable</td>
<td>Disables the BPDU block.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is set to disable.

**Usage Examples**

The following example enables the BPDU guard on the interface:

ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#spanning-tree bpduguard enable
**spanning-tree edgeport**

Use the `spanning-tree edgeport` command to set this interface to be an edgeport. This command overrides the Global setting. Use the `no` form of this command to return the interface to normal operation (non-edgeport).

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is set to disable.

**Usage Examples**

The following example configures the interface to be an edgeport:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#spanning-tree edgeport
```

An individual interface can be configured to not be considered an edgeport. For example:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#no spanning-tree edgeport
```
spanning-tree link-type [auto | point-to-point | shared]

Use the `spanning-tree link-type` command to configure the spanning-tree protocol link type for an interface. To return to the default value, use the `no` form of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto</td>
<td>Determines link type by the port's duplex settings.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Manually sets link type to point-to-point, regardless of duplex settings.</td>
</tr>
<tr>
<td>shared</td>
<td>Manually sets link type to shared, regardless of duplex settings.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, a port is set to `auto`.

**Functional Notes**

This command overrides the default link-type setting determined by the duplex of the individual port. By default a port configured for half-duplex is set to `shared` link type, and a port configured for full-duplex is set to `point-to-point` link type. Setting the link type manually overrides the default and forces the port to use the specified link type. Using the `link-type auto` command, restores the convention of determining link type based on duplex settings.

**Usage Examples**

The following example forces the link type to point-to-point, even if the port is configured to be half-duplex:

```
ProCurve(config)#bridge 1 protocol ieee
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#spanning-tree link-type point-to-point
```

**Technology Review**

Rapid transitions are possible in rapid spanning-tree protocol (RSTP) by taking advantage of point-to-point links (a port is connected to exactly one other bridge) and edge-port connections (a port is not connected to any additional bridges). Setting the link type to `auto` allows the spanning tree to automatically configure the link type based on the duplex of the link. Setting the link type to `point-to-point` allows a half-duplex link to act as if it were a point-to-point link.
**spanning-tree path-cost** `<value>`

Use the `spanning-tree path-cost` command to assign a cost to a bridge group that is used when computing the spanning-tree root path. To return to the default path-cost value, use the `no` form of this command.

**Syntax Description**

| `<value>` | Assigns a number to the bridge interface to be used as the path cost in spanning calculations (valid range: 0 to 65,535). |

**Default Values**

By default, the path-cost value is set to 19.

**Functional Notes**

The specified value is inversely proportional to the likelihood the bridge interface will be chosen as the root path. Set the path-cost value lower to increase the chance the interface will be the root. To obtain the most accurate spanning tree calculations, develop a system for determining path costs for links and apply it to all bridged interfaces.

**Usage Examples**

The following example assigns a path cost of 100 for bridge group 17 on an ATM sub-interface:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#spanning-tree path-cost 100
```

**Technology Review**

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.
spanning-tree port-priority <value>

Use the `spanning-tree port-priority` command to select the priority level of a port associated with a bridge. To return to the default bridge-group priority value, use the `no` version of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;value&gt;</code></td>
<td>Assigns a priority value for the bridge group; the lower the value, the higher the priority (valid range: 0 to 255).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the priority value is set to 128.

**Functional Notes**

The only time that this priority level is used is when two interfaces with a path to the root have equal cost. At that point, the level set in this command will determine which port the bridge will use. Set the priority value lower to increase the chance the interface will be used.

**Usage Examples**

The following example sets the maximum priority on the ATM sub-interface labeled 1.1 in bridge group 17:

```
ProCurve(config)#interface atm 1.1
ProCurve(config-atm 1.1)#spanning-tree priority 0
```
DEMAND INTERFACE CONFIGURATION COMMAND SET

To activate the Demand Interface Configuration mode, enter the `interface demand` command at the Global Configuration mode prompt. For example:

ProCurve#configure terminal
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `access-policy <policyname>` on page 736
- `bandwidth <value>` on page 737
- `called-number <DNIS number>` on page 738
- `caller-number <CLID number>` on page 739
- `connect-mode [answer | originate | either]` on page 740
- `connect-order [last-successful | round-robin | sequential]` on page 741
- `connect-sequence` on page 742
- `connect-sequence attempts <value>` on page 743
- `connect-sequence interface-recovery [retry-interval <seconds> | max-retries <value>]` on page 744
- `crypto map <mapname>` on page 745
- `demand-hold-queue <packets> timeout <seconds>` on page 747
- `dynamic-dns` on page 748
- `fair-queue <threshold>` on page 750
fast-idle <seconds> on page 751
hold-queue <queue size> out on page 752
idle-timeout <seconds> on page 753
ip commands begin on page 754
keepalive <seconds> on page 773
lldp receive on page 774
lldp send [management-address | port-description | system-capabilities | system-description | system-name] on page 775
match-interesting [list <acl name> | reverse list <acl name>] [in | out] on page 777
max-reserved-bandwidth <percent> on page 778
mtu <size> on page 779
peer default ip address <address> on page 780
ppp commands begin on page 781
qos-policy out <mapname> on page 789
resource pool <pool name> on page 790
snmp trap link-status on page 791
username <username> password <password> on page 792
access-policy <policyname>

Use the access-policy command to assign a specified access policy for the inbound traffic to an interface. Use the no form of this command to remove an access policy association.

**Note** Configured access policies will only be active if the ip firewall command has been entered at the Global Configuration mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.

**Syntax Description**

- `<policyname>` Identifies the configured access policy by alphanumeric descriptor. (All access policies are case-sensitive.)

**Default Values**

By default, there are no configured access policies associated with an interface.

**Functional Notes**

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter access policy <policy name>. For more details on creating and using access policies, refer to ip policy-class <policyname> on page 400.

**Usage Examples**

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the demand interface:

Enable the SROS security features:
ProCurve(config)#ip firewall

Create the access list (this is the packet selector):
ProCurve(config)#ip access-list extended InWeb
ProCurve(config-ext-nacl)#permit tcp any host 10.12.5.253 eq 80

Create the access policy that contains the access list InWeb:
ProCurve(config)#ip policy-class UnTrusted
ProCurve(config-policy-class)#allow list InWeb

Associate the access list with the demand interface (labeled 1):
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#access-policy UnTrusted
bandwidth <value>

Use the `bandwidth` command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the `no` form of this command to restore the default values.

**Syntax Description**

| <value> | Specifies the bandwidth value in kbps. |

**Default Values**

To view default values, use the `show interfaces` command.

**Functional Notes**

The `bandwidth` command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets the bandwidth of the demand interface to 10 Mbps:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#bandwidth 10000
```
called-number <DNIS number>

Use the called-number command to link calls to specific interfaces based on their dialed number identification service (DNIS) numbers. Multiple called numbers may be specified for an interface. Use the no form of this command to remove a configured called number.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax of DNIS number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;DNIS number&gt;</td>
<td>Identifies the called number to be linked to an interface. The DNIS number is limited to 20 digits.</td>
</tr>
</tbody>
</table>

**Default Values**

By default no called numbers are defined.

**Usage Examples**

The following example links calls with a DNIS number of **9165551212** to the demand interface 1:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#called-number 9165551212
```
**caller-number** `<CLID number>`

Use the **caller-number** command to link calls to specific interfaces based on its caller ID (CLID) number. Multiple caller ID numbers may be specified, allowing the interface to accept calls from different remote resources. Use the **no** form of this command to remove a configured caller number.

**Syntax Description**

| `<CLID number>` | Identifies the caller’s number to be linked to an interface. The CLID number is limited to 20 digits. |

**Default Values**

By default, no caller numbers are defined.

**Usage Examples**

The following example links calls with a CLID number of **9165551212** to the demand interface **1**:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#caller-number 9165551212
```
connect-mode [answer | originate | either]

Use the connect-mode command to configure the interface to only answer calls, only originate calls, or to both answer and originate calls. Use the no form of this command to restore the default values.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer</td>
<td>Specifies the interface may be used to answer calls but not originate calls.</td>
</tr>
<tr>
<td>originate</td>
<td>Specifies the interface may be used to originate calls but not answer calls.</td>
</tr>
<tr>
<td>either</td>
<td>Specifies the interface may be used to answer and originate calls.</td>
</tr>
</tbody>
</table>

Default Values

By default the connect mode is set to both answer and originate calls.

Usage Examples

The following example configures demand interface 1 to only answer calls:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#connect-mode answer
```
connect-order [last-successful | round-robin | sequential]

Use the connect-order command to specify the starting point in the connection sequence for each sequence activation. The connection sequence is a circular list. Use the no form of this command to restore the default values.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>last-successful</td>
<td>Specifies the connect sequence be processed beginning with the last successful entry or the first entry if there are no previous connections.</td>
</tr>
<tr>
<td>round-robin</td>
<td>Specifies the connect sequence be processed beginning with the entry that follows the last successful entry or the first entry if there are no previous connections.</td>
</tr>
<tr>
<td>sequential</td>
<td>Specifies the connect sequence be processed from the beginning of the list.</td>
</tr>
</tbody>
</table>

Default Values

By default, connect sequences are processed sequentially.

Usage Examples

The following example configures the connection sequence to begin with the last successful entry:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#connect-order last-successful
connect-sequence

Use the `connect-sequence` command to provide instructions to the interface on how to use the resource pool and telephone numbers to connect to demand destinations. Use the `no` form of this command to restore the default values. Variations of this command include the following:

```
connect-sequence <sequence number> dial-string <string> forced-analog
connect-sequence <sequence number> dial-string <string> forced-analog busyout-threshold <value>
connect-sequence <sequence number> dial-string <string> forced-isdn-56k
connect-sequence <sequence number> dial-string <string> forced-isdn-56k busyout-threshold <value>
connect-sequence <sequence number> dial-string <string> forced-isdn-64k
connect-sequence <sequence number> dial-string <string> forced-isdn-64k busyout-threshold <value>
connect-sequence <sequence number> dial-string <string> isdn-56k
connect-sequence <sequence number> dial-string <string> isdn-56k busyout-threshold <value>
connect-sequence <sequence number> dial-string <string> isdn-64k
connect-sequence <sequence number> dial-string <string> isdn-64k busyout-threshold <value>
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;sequence number&gt;</code></td>
<td>Specifies the number for this connection specification entry. Range: 1 to 65,535.</td>
</tr>
<tr>
<td><code>&lt;string&gt;</code></td>
<td>Specifies the telephone number to dial when using this connection. The dial string is limited to 20 digits.</td>
</tr>
<tr>
<td><code>forced-analog</code></td>
<td>Specifies that only analog resources may be used.</td>
</tr>
<tr>
<td><code>forced-isdn-56k</code></td>
<td>Specifies that only ISDN resources may be used. Call is placed using ISDN 56k.</td>
</tr>
<tr>
<td><code>forced-isdn-64k</code></td>
<td>Specifies that only ISDN resources may be used. Call is placed using ISDN 64k.</td>
</tr>
<tr>
<td><code>isdn-56k</code></td>
<td>Specifies any dial resource may be used if ISDN 56k call-type is used.</td>
</tr>
<tr>
<td><code>isdn-64k</code></td>
<td>Specifies any dial resource may be used if ISDN 64k call-type is used.</td>
</tr>
<tr>
<td><code>busyout-threshold &lt;value&gt;</code></td>
<td>Optional. Specifies the maximum number of connect sequence cycles during an activation attempt that must fail before it is skipped until the next activation attempt.</td>
</tr>
</tbody>
</table>

**Default Values**

By default any dial resource may be used.

**Usage Examples**

The following example instructs demand interface 1 to place the call using ISDN 64k:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#connect-sequence 65 dial-string 9165551212 forced-isdn-64k
```
connect-sequence attempts <value>

Use the `connect-sequence attempts` command to limit the number of times the connect sequence will cycle when its entries are unable to establish a connection. When the maximum number of attempts is exhausted, the interface will go into recovery mode. Refer to `connect-sequence interface-recovery [retry-interval <seconds>] | max-retries <value>` on page 744 for more information. Use the `no` form of this command to restore the default values.

**Syntax Description**

| <value> | Specifies the number of times the connect sequence will cycle through its entries if it is unable to make a connection. Range is 0 to 65,535. |

**Default Values**

By default the connect-sequence attempts are unlimited.

**Usage Examples**

The following example instructs demand interface 1 to attempt its connection sequence 500 times:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#connect-sequence attempts 500
```
connect-sequence interface-recovery [retry-interval <seconds> | max-retries <value>]

Use the connect-sequence interface-recovery command to allow the interface to go down in the event that the connect-sequence attempts value is exhausted. Refer to connect-sequence attempts <value> on page 743 for more information. Use the no form of this command to restore the default values.

Syntax Description

retry-interval <seconds>  Optional. Specifies the number of seconds the interface will wait between connect sequence cycles during recovery attempts.

max-retries <value>  Optional. Specifies the maximum number of times the connect sequence will cycle in an attempt to bring the interface back up. When in interface recovery mode, this value overrides the connect-sequence attempts value.

Default Values

By default, the connect-sequence interface-recovery retry-interval is set to 120 seconds and max-retries are unlimited.

Usage Examples

The following example configures demand interface 1 to wait 60 seconds between retry attempts with a maximum number of 500 retries:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#connect-sequence interface-recovery retry-interval 60 max-retries 500
**crypto map <mapname>**

Use the `crypto map` command to associate crypto maps with the interface.

---

**Note**  
When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the *VPN Configuration Guide* located on the *ProCurve SROS Documentation* CD provided with your unit.

---

**Syntax Description**

- `<mapname>` Assigns a crypto map name to the interface.

**Default Values**

By default, no crypto maps are assigned to an interface.

**Functional Notes**

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the unencrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the diagram above, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only unencrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, unencrypted source of the data. The destination information will be the far end, unencrypted destination of the data. However, ACLs for a policy class work in reverse. The source information for the ACL in a policy class is the far end. The destination information is the local side.

**Usage Examples**

The following example applies all crypto maps with the name *MyMap* to the demand interface:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#crypto map MyMap
```
The following example configures demand interface 1 to hold 50 packets in the queue for up to 120 seconds:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#demand-hold-queue 50 timeout 120
dynamic-dns

Use the **dynamic-dns** command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the **no** versions of these commands to disable these features. Variations of this command include:

- **dynamic-dns custom** `<hostname> <minutes>`
- **dynamic-dns dyndns** `<hostname> <username> <password>`
- **dynamic-dns dyndns-custom** `<hostname> <username> <password>`
- **dynamic-dns dyndns-static** `<hostname> <username> <password>`

**Syntax Description**

- `<hostname>` Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).
- `<minutes>` Specifies the intervals in minutes to update the server with information (updates also occur when the interface’s IP address changes regardless of the update intervals).

Refer to **Functional Notes** below for additional argument descriptions.

**Default Values**

No default is necessary for this command.

**Functional Notes**

- **custom** - Constanttime.com’s Custom Dynamic DNS℠ service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows to manage IP address mappings (A records), domain aliases (CNAME records) and mail servers (MX records).

- **dyndns** - The Dynamic DNS℠ service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- **dyndns-custom** - DynDNS.org’s Custom DNS℠ service provides a full DNS solution, giving you complete control over an entire domain name. A web-based interface provides two levels of control over your domain, catering to average or power users. Five globally redundant DNS servers ensure that your domain will always resolve.

A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background, and provides layout and functionality similar to a BIND zone file allowing for the creation of nearly any record type.

Custom DNS℠ can be used with both static and dynamic IPs, and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
dyndns-static - The Static DNS service is similar to Dynamic DNS service, in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate though the DNS system. This service is provided for up to five hostnames.

If your IP address doesn't change often or at all, but you still want an easy name to remember it by (without having to purchase your own domain name) Static DNS service is ideal for you.

If you would like to use your own domain name (such as yourname.com) you need Custom DNS service which also provides full dynamic and static IP address support.

**Usage Examples**

The following example sets the dynamic-dns to dyndns-custom with hostname **host**, username **user**, and password **pass**:  

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#dynamic-dns dyndns-custom host user pass
**fair-queue <threshold>**

Use the `fair-queue` command to enable weighted fair queuing (WFQ) on an interface. Use the `no` form of this command to disable WFQ and enable FIFO queuing for an interface. WFQ is enabled by default for WAN interfaces.

**Syntax Description**

```
<threshold>
```

Optional. Specifies the maximum number of packets that can be present in each conversation sub-queue. Packets received for a conversation after this limit is reached are discarded. Range: 16 to 512 packets.

**Default Values**

By default, `fair-queue` is enabled with a threshold of 64 packets.

**Usage Examples**

The following example enables WFQ on the interface with a threshold set at 100 packets:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#fair-queue 100
```
fast-idle <seconds>

Use the fast-idle command to set the amount of time the demand interface connection will remain active in the absence of interesting traffic when there is contention for the demand resources being used by this interface. Use the no form of this command to restore the default values.

**Syntax Description**

| <seconds> | Specifies the number of seconds the interface will remain up in the absence of interesting traffic. Range is 1 to 2,147,483. |

**Default Values**

By default, fast-idle is set to 120 seconds.

**Usage Examples**

The following example sets fast idle to 1,073,752 seconds:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#fast-idle 1073752
```
hold-queue <queue size> out

Use the hold-queue out command to change the overall size of an interface's WAN output queue.

Syntax Description

<queue size> Specifies the total number of packets the output queue can contain before packets are dropped. Range is 16 to 1000.

Default Values

The default queue size for WFQ is 400. The default queue size for PPP FIFO and Frame Relay round-robin is 200.

Usage Examples

The following example sets the overall output queue size to 700:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#hold-queue 700 out
idle-timeout <seconds>

Use the `idle-timeout` command to set the amount of time the interface link/bundle will remain up in the absence of interesting traffic. Interesting traffic and direction logic are set using the `match-interesting` commands. Refer to `match-interesting [list <acl name> | reverse list <acl name>] [in | out]` on page 777 for more information. Use the `no` form of this command to restore the default values.

**Syntax Description**

| <seconds> | Specifies the number of seconds the interface will remain up in the absence of interesting traffic. Range is 1 to 2,147,483. |

**Default Values**

By default, `idle-timeout` is set to 120 seconds.

**Usage Examples**

The following example configures demand interface 1 to time out after 360 seconds:

```plaintext
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#idle-timeout 360
```
**ip access-group** <listname> [in | out]

Use the `ip access-group` command to create an access list to be used for packets transmitted on or received from the specified interface. Use the `no` form of this command to disable this type of control.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;listname&gt;</code></td>
<td>Indicates the assigned IP access list name.</td>
</tr>
<tr>
<td>in</td>
<td>Enables access control on packets received on the specified interface.</td>
</tr>
<tr>
<td>out</td>
<td>Enables access control on packets transmitted on the specified interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, these commands are disabled.

**Functional Notes**

When this command is enabled, the IP destination address of each packet must be validated before being passed through. If the packet is not acceptable per these settings, it is dropped.

**Usage Examples**

The following example sets up the router to only allow Telnet traffic into the demand interface:

```
ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#interface demand 1
ProCurve(config-demand 1)#ip access-group TelnetOnly in
```
ip address negotiated [no-default]

Use the `ip address negotiated` command to allow the interface to negotiate (i.e., be assigned) an IP address from the far end PPP connection. Use the `no` form of this command to disable the negotiation for an IP address.

**Syntax Description**

| no-default | Optional. Prevents the insertion of a default route. Some systems already have a default route configured and need a static route to the PPP interface to function correctly. |

**Default Values**

By default, the interface is assigned an address with the `ip address <address><mask>` command.

**Usage Examples**

The following example enables the demand interface to negotiate an IP address from the far end connection:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip address negotiated
```

The following example enables the demand interface to negotiate an IP address from the far end connection without inserting a default route:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip address negotiated no-default
```
ip address <address> <mask> secondary

Use the `ip address` command to define an IP address on the specified interface. Use the optional keyword `secondary` to define a secondary IP address. Use the `no` form of this command to remove a configured IP address.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).</td>
</tr>
<tr>
<td><code>&lt;mask&gt;</code></td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
<tr>
<td><code>secondary</code></td>
<td>Optional. Configures a secondary IP address for the specified interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no assigned IP addresses.

**Functional Notes**

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

**Usage Examples**

The following example configures a secondary IP address of 192.168.72.101/30:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip address 192.168.72.101 255.255.255.252 secondary
```
**ip helper-address** <address>

Use the `ip helper-address` command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the `no` form of this command to disable forwarding packets.

**Syntax Description**

<address> Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

**Default Values**

By default, broadcast UDP packets are not forwarded.

**Functional Notes**

When used in conjunction with the `ip forward-protocol` command, the `ip helper-address` feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign a helper address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the `ip forward-protocol` command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the `ip helper-address` feature:

1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the `ip forward-protocol` command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

**Usage Examples**

The following example forwards all DNS broadcast traffic to the DNS server with IP address `192.168.5.99`:

```
ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip helper-address 192.168.5.99
```
**ip igmp**

Use the `ip igmp` command to configure multicasting-related functions for the interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>immediate-leave</td>
<td>Specifies that if only one host (or IGMP snooping switch) is connected to the interface, when a leave is received, multicast of that group is immediately terminated as opposed to sending a group query and timing out the group if no device responds. Applies to all groups when configured.</td>
</tr>
<tr>
<td>last-member-query-interval</td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group's multicast packets to the interface. Range: 100 to 65535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td>querier-timeout</td>
<td>Specifies the number of seconds that the router waits after the current querier’s last query before it takes over as querier (IGMP V2). Range: 60-300 seconds. Default: 2x the <code>query-interval</code> value.</td>
</tr>
<tr>
<td>query-interval</td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td>query-max-response-time</td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td>static-group</td>
<td>Configures the router’s interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>version [1</td>
<td>2]</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example sets the query message interval on the interface to 200 milliseconds:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip igmp last-member-query-interval 200
```
ip mcast-stub downstream

Use the **ip mcast-stub downstream** command to enable multicast forwarding and IGMP (router mode) on an interface and place it in multicast stub downstream mode. Use the **no** form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the **ip mcast-stub helper-address** and **ip mcast-stub upstream** commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. Refer to **ip mcast-stub helper-address <ip address>** on page 397 and **ip mcast-stub upstream** on page 762 for more information.

**Usage Examples**

The following example enables multicast forwarding and IGMP on the interface:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip mcast-stub downstream
ip mcast-stub fixed

Use the **ip mcast-stub fixed** command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the **no** form of this command to disable this mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Multicast routing must be enabled prior to setting **ip mcast-stub fixed** on the selected interface. Also, use the **ip igmp static-group \(<A.B.C.D>\)** command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

**Usage Examples**

The following example enables multicast traffic forwarding and IGMP on the interface:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip mcast-stub fixed
```
ip mcast-stub helper-enable

Use the `ip mcast-stub helper-enable` command to assign the `ip mcast-stub helper-address` as the IGMP proxy. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address`, `ip mcast-stub upstream`, and `ip mcast-stub downstream` commands. When enabled, the interface becomes a helper forwarding interface. The IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the unit to perform as an IGMP proxy. Refer to `ip mcast-stub helper-address <ip address>` on page 397, `ip mcast-stub downstream` on page 759, and `ip mcast-stub upstream` on page 762 for more information.

**Usage Examples**

The following example sets the helper address as the IGMP proxy:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip mcast-stub helper-enable
```
ip mcast-stub upstream

Use the `ip mcast-stub upstream` command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub downstream` commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router's unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub downstream` on page 759 for more information.

**Usage Examples**

The following example enables multicast forwarding on the interface:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip mcast-stub upstream
```
ip ospf

Use the `ip ospf` command to customize OSPF settings (if needed).

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>authentication-key</strong></td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td><strong>cost &lt;value&gt;</strong></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td><strong>dead-interval &lt;seconds&gt;</strong></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>hello-interval &lt;seconds&gt;</strong></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>message-digest-key &lt;keyid&gt; md5 &lt;key&gt;</strong></td>
<td>Configures OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td><strong>priority &lt;value&gt;</strong></td>
<td>Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td><strong>retransmit-interval &lt;seconds&gt;</strong></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>transmit-delay &lt;seconds&gt;</strong></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32,767.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>retransmit-interval &lt;seconds&gt;</strong></td>
<td>5 seconds</td>
</tr>
<tr>
<td><strong>transmit-delay &lt;seconds&gt;</strong></td>
<td>1 second</td>
</tr>
<tr>
<td><strong>hello-interval &lt;seconds&gt;</strong></td>
<td>10 seconds</td>
</tr>
<tr>
<td><strong>dead-interval &lt;seconds&gt;</strong></td>
<td>40 seconds</td>
</tr>
</tbody>
</table>

**Usage Example**

The following example sets the maximum number of seconds allowed between hello packets to 25,000:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip ospf dead-interval 25000
```
ip ospf authentication [message-digest | null]

Use the `ip ospf authentication` command to authenticate an interface that is performing OSPF authentication.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-digest</td>
<td>Optional. Selects message-digest authentication type.</td>
</tr>
<tr>
<td>null</td>
<td>Optional. Specifies that no authentication be used.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, `ip ospf authentication` is set to null (meaning no authentication is used).

**Usage Examples**

The following example specifies that no authentication will be used on the demand interface:

```plaintext
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip ospf authentication null
```
ip ospf network [broadcast | point-to-point]

Use the `ip ospf network` command to specify the type of network on this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcast</td>
<td>Sets the network type for broadcast.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Sets the network type for point-to-point.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

**Functional Notes**

A point-to-point network will not elect designated routers.

**Usage Examples**

The following example designates a broadcast network type:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip ospf network broadcast
```
ip policy route-map <mapname>

Use the ip policy route-map command to assign a policy route map to this interface. Use the no form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

Syntax Description

<mapname> Specifies the route map to associate with this interface.

Default Values

By default, policy-based routing is disabled for all interfaces.

Usage Examples

The following example associates the route map named MyMap with demand interface 1:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip policy route-map MyMap
ip proxy-arp <address> <subnet mask>

Use the `ip proxy-arp` to enable proxy Address Resolution Protocol (ARP) on the interface. Use the `no` form of this command to disable this feature.

**Syntax Description**

```plaintext
<address>
Defines the IP address for the interface in dotted decimal notation (for example, 192.22.73.101).

<subnet mask>
Specifies the subnet mask that corresponds to the listed IP address.
```

**Default Values**

By default, proxy ARP is enabled.

**Functional Notes**

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all proxy ARP requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

**Usage Examples**

The following example enables proxy ARP on the virtual demand interface:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip proxy-arp
```
ip rip receive version [1 | 2]

Use the `ip rip receive version` command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accepts only received RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Accepts only received RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip receive version` to specify a RIP version that overrides the `version` (in the Router RIP) configuration.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the virtual demand interface to accept only RIP version 2 packets:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip rip receive version 2
**ip rip send version [1 | 2]**

Use the `ip rip send version` command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transmits only RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Transmits only RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces transmit RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip send version` to specify a RIP version that overrides the `version` (in the Router RIP) configuration.

The SROS only transmits one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the virtual demand interface to transmit only RIP version 2 packets:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip rip send version 2
```
**ip rip summary-address** `<network address> <network mask>`

Use the `ip rip summary-address` command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the `no` form of this command to disable this mode.

**Syntax Description**

<table>
<thead>
<tr>
<th><strong>Syntax</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network address&gt;</code></td>
<td>Specifies the IP address of the network to be summarized.</td>
</tr>
<tr>
<td><code>&lt;network mask&gt;</code></td>
<td>Specifies the network mask to be applied to the specific network to be summarized.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no manual summarization is applied by RIP.

**Functional Notes**

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the `ip rip summary-address` command. This command is only effective if RIP version 2 is configured.

**Usage Examples**

The following example enables manual summarization on the specified IP address:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip rip summary-address 10.10.123.0 255.255.255.0
ip route-cache

Use the ip route-cache command to enable fast-cache switching on the interface. Use the no form of this command to disable fast-cache switching and return to process switching mode.

Note Using Network Address Translation (NAT) or the SROS firewall capabilities on an interface requires process switching mode (using the no ip route-cache command).

Syntax Description
No subcommands.

Default Values
By default, route caching is enabled on all interfaces.

Functional Notes
Fast-cache switching allows an IP interface to provide optimum performance when processing IP traffic.

Usage Examples
The following example enables fast-cache switching on the virtual demand interface:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip route-cache
**ip unnumbered <interface>**

Use the `ip unnumbered` command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the `no` form of this command to remove the unnumbered configuration.

**Syntax Description**

| `<interface>` | Specifies the interface (in the format `type slot/port`) that contains the IP address to use as the source address for all packets transmitted on this interface. Type `show ip unnumbered interface ?` for a list of valid interfaces. |

**Default Values**

By default, all interfaces are configured to use a specified IP address (using the `ip address` command).

**Functional Notes**

If `ip unnumbered` is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying `ip unnumbered eth 0/1` while in the Demand Interface Configuration mode configures the demand interface to use the IP address assigned to the Ethernet interface for all IP processing. In addition, the SROS uses the specified interface information when sending route updates over the unnumbered interface. Static routes may either use the interface name (`ppp 1`) or the far-end address (if it will be discovered).

**Usage Examples**

The following example configures the demand interface (labeled `demand 1`) to use the IP address assigned to the Ethernet interface (`eth 0/1`):

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ip unnumbered eth 0/1
```
keepalive <seconds>

Use the keepalive command to enable the transmission of keepalive packets on the interface and specify the time interval in seconds between transmitted packets. Use the no form of this command to return to the default setting.

Syntax Description

| <seconds> | Defines the time interval (in seconds) between transmitted keepalive packets (valid range: 0 to 32,767 seconds). |

Default Values

By default, the time interval between transmitted keepalive packets is 10 seconds.

Functional Notes

If three keepalive packets are sent to an interface with no response, the interface is considered down. To detect interface failures quickly, specify a smaller keepalive time.

Usage Examples

The following example specifies a keepalive time of 5 seconds on the virtual demand interface:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#keepalive 5
lldp receive

Use the lldp receive command to allow LLDP packets to be received on this interface.

Syntax Description

No subcommands.

Default Values

By default, all interfaces are configured to send and receive LLDP packets.

Usage Examples

The following example configures the demand interface to receive LLDP packets:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#lldp receive
lldp send [management-address | port-description | system-capabilities | system-description | system-name]

Use the **lldp send** command to configure this interface to transmit LLDP packets or to control the types of information contained in the LLDP packets transmitted by this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>management-address</td>
<td>Enables transmission of management address information on this interface.</td>
</tr>
<tr>
<td>port-description</td>
<td>Enables transmission of port description information on this interface.</td>
</tr>
<tr>
<td>system-capabilities</td>
<td>Enables transmission of this device’s system capabilities on this interface.</td>
</tr>
<tr>
<td>system-description</td>
<td>Enables transmission of this device’s system description on this interface.</td>
</tr>
<tr>
<td>system-name</td>
<td>Enables transmission of this device’s system name on this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

Be default, all interfaces are configured to transmit and receive LLDP packets of all types.

**Functional Notes**

Individual LLDP information can be enabled or disabled using the various forms of the **lldp send** command. For example, use the **lldp send and-receive** command to enable transmit and receive of all LLDP information. Then use the **no lldp send port-description** command to prevent LLDP from transmitting port description information.

**Usage Examples**

The following example configures the demand interface to transmit LLDP packets containing all enabled information types:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#lldp send
```
lldp send-and-receive

Use the lldp send-and-receive command to configure this interface to transmit and receive LLDP packets.

Syntax Description
No subcommands.

Default Values
Be default, all interfaces are configured to transmit and receive LLDP packets of all types.

Functional Notes
Individual LLDP information can be enabled or disabled using the various forms of the lldp send command. For example, use the lldp send-and-receive command to enable transmit and receive of all LLDP information. Then use the no lldp send port-description command to prevent LLDP from transmitting port description information.

Usage Examples
The following example configures the demand interface (demand 1) to transmit and receive LLDP packets containing all information types:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#lldp send-and-receive
**match-interesting [list <acl name>] | reverse list <acl name>] [in | out]**

Use the `match-interesting` command to allow an access list (ACL) to specify which traffic attempting to cross this interface will be considered interesting. Use the `no` form of this command to restore the default values.

**Syntax Description**

- **list <acl name>**
  - Specifies using an ACL with normal (source, destination) ACL matching logic.

- **reverse list <acl name>**
  - Specifies using an ACL with reverse (destination, source) ACL matching logic.

- **in**
  - Optional. Specifies that only incoming traffic is interesting.

- **out**
  - Optional. Specifies that only outgoing traffic is interesting.

**Default Values**

By default, no interesting traffic is defined.

**Usage Examples**

The following example instructs demand interface 1 to use the access list `MyACL` when checking for interesting traffic:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#match-interesting list MyACL in
```
max-reserved-bandwidth <percent>

Use the `max-reserved-bandwidth` command to specify the percentage of interface bandwidth reserved for use in user-defined (priority or class-based) queues. The remainder of the interface bandwidth is reserved for system critical traffic and is not available to user-defined queues. Use the `no` form of this command to restore the default values.

**Caution** Reserving a portion of the interface bandwidth for system critical traffic is necessary for proper operation. Specifying the entire interface bandwidth for use in user-defined queues can cause undesirable operation.

**Syntax Description**

| `<percent>` | Specifies the percentage of interface bandwidth to make available for user-defined (priority or class-based) queues. Enter an integer 1 to 100. |

**Default Values**

By default, `max-reserved-bandwidth` is set to **75** which reserves 25 percent of the interface bandwidth for system critical traffic.

**Usage Examples**

The following example specifies 85 percent of the bandwidth on demand interface 1 to be available for use in user-defined queues:

```
(config)#interface demand 1
(config-demand 1)#max-reserved-bandwidth 85
```
mtu <size>

Use the mtu command to configure the maximum transmit unit (MTU) size for the active interface. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ATM interfaces 64 to 1520</td>
</tr>
<tr>
<td></td>
<td>Demand interfaces 64 to 1520</td>
</tr>
<tr>
<td></td>
<td>Ethernet interfaces 64 to 1500</td>
</tr>
<tr>
<td></td>
<td>FDL interfaces 64 to 256</td>
</tr>
<tr>
<td></td>
<td>HDLC interfaces 64 to 1520</td>
</tr>
<tr>
<td></td>
<td>Loopback interfaces 64 to 1500</td>
</tr>
<tr>
<td></td>
<td>Tunnel interfaces 64 to 18,190</td>
</tr>
<tr>
<td></td>
<td>Virtual Frame Relay sub-interfaces 64 to 1520</td>
</tr>
<tr>
<td></td>
<td>Virtual PPP interfaces 64 to 1500</td>
</tr>
</tbody>
</table>

Default Values

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>The default values for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ATM interfaces 1500</td>
</tr>
<tr>
<td></td>
<td>Demand interfaces 1500</td>
</tr>
<tr>
<td></td>
<td>Ethernet interfaces 1500</td>
</tr>
<tr>
<td></td>
<td>FDL interfaces 256</td>
</tr>
<tr>
<td></td>
<td>HDLC interfaces 1500</td>
</tr>
<tr>
<td></td>
<td>Loopback interfaces 1500</td>
</tr>
<tr>
<td></td>
<td>Tunnel interfaces 1500</td>
</tr>
<tr>
<td></td>
<td>Virtual Frame Relay sub-interfaces 1500</td>
</tr>
<tr>
<td></td>
<td>Virtual PPP interfaces 1500</td>
</tr>
</tbody>
</table>

Functional Notes

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

Usage Examples

The following example specifies an MTU of 1200 on the virtual demand interface:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#mtu 1200
```
peer default ip address <address>

Use the peer default ip address command to specify the default IP address of the remote end of this interface.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;address&gt;</th>
<th>Specifies the default IP address for the remote end (A.B.C.D).</th>
</tr>
</thead>
</table>

Default Values

By default, there is no assigned peer default IP address.

Functional Notes

This command is useful if the peer does not send the IP address option during PPP negotiations.

Usage Examples

The following example sets the default peer IP address to 192.168.71.50:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#peer default ip address 192.168.71.50
ppp authentication [chap | pap]

Use the **ppp authentication** command to specify the authentication protocol on the PPP virtual interface that the peer should use to authenticate itself.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chap</td>
<td>Configures CHAP authentication on the interface.</td>
</tr>
<tr>
<td>pap</td>
<td>Configures PAP authentication on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, PPP endpoints have no authentication configured.

**Technology Review**

CHAP and PAP are two authentication methods that enjoy widespread support. Both methods are included in the SROS and are easily configured.

**Note**  
The authentication method set up on the local router can be different from that on the peer. Also, just because one router requires authentication from its peer does not mean it also has to authenticate itself to the peer.

**Defining PAP**

The Password Authentication Protocol (PAP) is used to verify that the PPP peer is a permitted device by checking a username and password configured on the peer. The username and password are both sent unencrypted across the connecting private circuit.

PAP requires two-way message passing. First, the router that is required to be authenticated (say the peer) sends an authentication request with its username and password to the router requiring authentication (say the local router). The local router then looks up the username and password in the username database within the PPP interface, and if they match sends an authentication acknowledge back to the peer.

**Note**  
The PPP username and password database is separate and distinct from the global username password database. For PAP and CHAP, use the database under the PPP interface configuration.

Several example scenarios are given below for clarity.

**Configuring PAP Example 1: Only the local router requires the peer to authenticate itself.**

On the local router (hostname Local):

Local(config-demand 1)#**ppp authentication pap**
Local(config-demand 1)#**username farend password same**
On the peer (hostname Peer):
Peer(config-demand 1)#ppp pap sent-username farend password same

The first line of the configuration sets the authentication mode as PAP. This means the peer is required to authenticate itself to the local router via PAP. The second line is the username and password expected to be sent from the peer. On the peer, the **ppp pap sent-username** command is used to specify the appropriate matching username and password.

**Configuring PAP Example 2: Both routers require the peer to authenticate itself.**
On the local router (hostname Local):
Local(config-demand 1)#ppp authentication pap
Local(config-demand 1)#username farend password far
Local(config-demand 1)#ppp pap sent-username nearend password near

On the peer (hostname Peer):
Peer(config-demand 1)#ppp authentication pap
Peer(config-demand 1)#username nearend password near
Peer(config-demand 1)#ppp pap sent-username farend password far

Now both routers send the authentication request, verify that the username and password sent match what is expected in the database, and send an authentication acknowledge.

**Defining CHAP**
The Challenge-Handshake Authentication Protocol (CHAP) is a three-way authentication protocol composed of a challenge response and success or failure. The MD5 protocol is used to protect usernames and passwords in the response.

First, the local router (requiring its peer to be authenticated) sends a "challenge" containing only its own unencrypted username to the peer. The peer then looks up the username in the username database within the PPP interface, and if found takes the corresponding password and its own hostname and sends a "response" back to the local router. This data is encrypted. The local router verifies that the username and password are in its own username database within the PPP interface, and if so sends a "success" back to the peer.

---

**Note**
The PPP username and password database is separate and distinct from the global username password database. For PAP and CHAP, use the database under the PPP interface configuration.
Several example scenarios are given below for clarity.

**Configuring CHAP Example 1: Only the local router requires the peer to authenticate itself.**

On the local router (hostname Local):
```
Local(config-demand 1)# ppp authentication chap
Local(config-demand 1)# username Peer password same
```

On the peer (hostname Peer):
```
Peer(config-demand 1)# username Local password same
```

The first line of this configuration sets the authentication mode to CHAP. This means the peer is required to authenticate itself to the local router via CHAP. The second line is the username and password expected to be sent from the peer. The peer must also have the **username** up both to verify the incoming username from the local router and to use the password (along with its hostname) in the response to the local router.

**Note**  
Both ends must have identical passwords.

**Configuring CHAP Example 2: Both routers require the peer to authenticate itself.**

On the local router (hostname Local):
```
Local(config-demand 1)# ppp authentication chap
Local(config-demand 1)# username Peer password same
```

On the peer (hostname Peer):
```
Peer(config-demand 1)# ppp authentication chap
Peer(config-demand 1)# username Local password same
```

This is basically identical to Example 1 except that both routers will now challenge each other and respond.

**Configuring CHAP Example 3: Using the ppp chap hostname command as an alternate solution.**

On the local router (hostname Local):
```
Local(config-demand 1)# ppp authentication chap
Local(config-demand 1)# username Peer password same
Local(config-demand 1)# ppp chap hostname nearend
```

On the peer (hostname Peer):
```
Peer(config-demand 1)# username nearend password same
```

Notice the peer is expecting username “nearend” even though the local router’s hostname is “Local.” Therefore the local router can use the **ppp chap hostname** command to send the correct name on the challenge.
Configuring CHAP Example 4: Using the ppp chap password command as an alternate solution.

On the local router (hostname Local):

Local(config-demand 1)#ppp authentication chap
Local(config-demand 1)#username Peer password different

On the peer (hostname Peer):

Peer(config-demand 1)#username Local password same
Peer(config-demand 1)#ppp chap password different

Here the local router challenges with hostname "Local." The peer verifies the name in the username database, but instead of sending the password "same" in the response, it uses the one in the ppp chap password command. The local router then verifies that user "Peer" with password "different" is valid and sends a "success."
**ppp chap hostname** `<hostname>`

Use the `ppp chap hostname` command to configure an alternate hostname for CHAP PPP authentication. Use the `no` form of this command to remove a configured hostname. For more information on PAP and CHAP functionality, refer to the *Technology Review* section for the command `ppp authentication [chap | pap]` on page 781.

**Syntax Description**

- `<hostname>` Specifies a hostname using an alphanumerical string up to 80 characters in length.

**Default Values**

By default, there are no configured PPP CHAP hostnames.

**Usage Examples**

The following example specifies a PPP CHAP hostname of `my_host`:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ppp chap hostname my_host
```
ppp chap password  <password>

Use the ppp chap password command to configure an alternate password when the peer requires CHAP PPP authentication. Use the no form of this command to remove a configured password. For more information on PAP and CHAP functionality, refer to the Technology Review section for the command ppp authentication [chap | pap] on page 781.

Syntax Description

<password>  Specifies a password using an alphanumeric string up to 80 characters in length.

Default Values

By default, there is no defined PPP CHAP password.

Usage Examples

The following example specifies a PPP CHAP password of my_password:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ppp chap password my_password
ppp multilink [fragmentation | interleave]

Use the `ppp multilink` command to enable multilink PPP (MPPP) operation on an existing PPP interface. Use the `no` form of this command to disable.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fragmentation</td>
<td>Enables multilink fragmentation operation.</td>
</tr>
<tr>
<td>interleave</td>
<td>Enables multilink interleave operation.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, MPPP is disabled.

**Functional Notes**

When enabled, this interface is capable of the following:

- Combining multiple physical links into one logical link.
- Receiving upper layer protocol data units (PDU), fragmenting and transmitting over the physical links.
- Receiving fragments over the physical links and reassembling them into PDUs.

The fragmentation and interleave options can be used to enhance the multilink operation. Fragmentation is used to reduce serialization delays of large packets. The fragmentation process evenly divides the data among all links in the bundle with a minimum packet size of 96 bytes. The interleave operation is used with streaming protocols to reduce delay by giving priority to packets identified as high priority. In order delivery is guaranteed with multilink fragmentation, but is not guaranteed with multilink interleave operation.

The multilink bundle will remain active with a minimum of one physical link. Physical links may be dynamically added or removed from the multilink bundle with minor interruption to traffic flow.

**Usage Examples**

The following example enables MPPP:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ppp multilink
```
**ppp pap sent-username** `<username>` **password** `<password>`

Use the **ppp pap sent-username/password** command to configure a username and password when the peer requires PAP PPP authentication. Use the **no** form of this command to remove a configured password. For more information on PAP and CHAP functionality, refer to the Technology Review section for the command **ppp authentication [chap | pap]** on page 781.

**Syntax Description**

| `<username>` | Specifies a username by alphanumeric string up to 80 characters in length (the username is case-sensitive). |
| `<password>` | Specifies a password by alphanumeric string up to 80 characters in length (the password is case-sensitive). |

**Default Values**

By default, there is no defined **ppp pap sent-username** and **password**.

**Usage Examples**

The following example specifies a PPP PAP sent-username of **local** and a password of **my_password**:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#ppp pap sent-username local password my_password
```
**qos-policy out <mapname>**

Use the `qos-policy out` command to apply a previously-configured QoS map to an interface. Use the `no` form of this command to remove the map from the interface. The keyword `out` specifies that this policy will be applied to outgoing packets.

**Syntax Description**

| <mapname> | Specifies the name of a previously-created QoS map (refer to `qos map <mapname> <sequence number>` on page 451 for more information). |

**Default Values**

No default value is necessary for this command.

**Usage Examples**

The following example applies the QoS map `VOICEMAP` to the demand 1 interface:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#qos-policy out VOICEMAP
```
resource pool <pool name>

Use the resource pool command to associate a resource pool with the demand interface. No more than one resource pool may be associated with an interface. Refer to resource pool-member <pool-name> [<cost>] on page 1119 for more information. Use the no form of this command to restore the default values.

Syntax Description

| <pool name> | Specifies the resource pool that this interface will use to originate/answer demand connections. |

Default Values

By default, no resource pool is associated with this interface.

Usage Examples

The following example associates the resource pool named Pool1 with demand interface 1:

ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#resource pool Pool1
snmp trap link-status

Use the `snmp trap link-status` command to control the Simple Network Management Protocol (SNMP) variable ifLinkUpDownTrapEnable (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the `no` form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the ifLinkUpDownTrapEnable OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The `snmp trap link-status` command is used to control the RFC2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the virtual demand interface:

```
ProCurve(config)#interface demand 1
ProCurve(config-demand 1)#no snmp trap link-status
```
**username <username> password <password>**

Configures the username and password of the peer to use for demand authentication.

**Syntax Description**

- `<username>` Specifies a username by alphanumerical string up to 30 characters in length (the username is case-sensitive).
- `<password>` Specifies a password by alphanumerical string up to 30 characters in length (the password is case-sensitive).

**Default Values**

By default, there is no established username and password.

**Functional Notes**

PAP uses this entry to check received information from the peer. CHAP uses this entry to check the received peer hostname and a common password.

**Usage Examples**

The following example creates a username of PROCURVE with password ROUTER for the demand link labeled 5:

```
(config)#interface demand 5
(config-demand 5)#username PROCURVE password ROUTER
```
To activate the Frame Relay Interface Configuration mode, enter the `interface frame-relay` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `bandwidth <value>` on page 794
- `encapsulation frame-relay ietf` on page 795
- `fair-queue <threshold>` on page 796
- `frame-relay commands` begin on page 797
- `hold-queue <queue size> out` on page 809
- `max-reserved-bandwidth <percent>` on page 810
- `qos-policy out <mapname>` on page 811
- `snmp trap` on page 812
- `snmp trap link-status` on page 813
**bandwidth <value>**

Use the `bandwidth` command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the `no` form of this command to restore the default values.

**Syntax Description**

<value> Specifies bandwidth in kbps.

**Default Values**

No default value is necessary for this command.

**Functional Notes**

The `bandwidth` command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the Frame Relay interface to 10 Mbps:

```bash
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#bandwidth 10000
```
encapsulation frame-relay ietf

Use the `encapsulation frame-relay ietf` command to configure the encapsulation on a virtual Frame Relay interface as IETF (RFC 1490). Currently, this is the only encapsulation setting. Settings for this option must match the far-end router’s settings in order for the Frame Relay interface to become active.

**Syntax Description**

No subcommands.

**Default Values**

By default, all Frame Relay interfaces use IETF encapsulation.

**Usage Examples**

The following example configures the endpoint for IETF encapsulation:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#encapsulation frame-relay ietf
```
**fair-queue <threshold>**

Use the **fair-queue** command to enable weighted fair queuing (WFQ) on an interface. Use the **no** form of this command to disable WFQ and enable FIFO (first-in-first-out) queueing for an interface. WFQ is enabled by default for WAN interfaces.

**Syntax Description**

| <threshold> | Optional. Specifies the maximum number of packets that can be present in each conversation sub-queue. Packets received for a conversation after this limit is reached are discarded. Range: 16 to 512. |

**Default Values**

By default, **fair-queue** is enabled with a threshold of 64 packets.

**Usage Examples**

The following example enables WFQ on the interface with a threshold set at 100 packets:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#fair-queue 100
```
frame-relay intf-type [dce | dte | nni]

Use the frame-relay intf-type command to define the Frame Relay signaling role needed for the endpoint. Use the no form of this command to return to the default value.

Syntax Description

dce
- Specifies DCE or network signaling role. Use this interface type when you need the unit to emulate the frame switch.

dte
- Specifies DTE or user signaling role. Use this interface type when connecting to a Frame Relay switch (or piece of equipment emulating a frame switch).

nni
- Configures the interface to support both network and user signaling (DTE or DCE) when necessary.

Default Values

By default, frame-relay intf-type is set to dte.

Usage Examples

The following example configures the Frame Relay endpoint for DCE signaling:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay intf-type dce
frame-relay lmi-n391dce <polls>

Use the `frame-relay lmi-n391dce` command to set the N391 full status polling counter for the DCE endpoint. Typical applications should leave the default value for this timer. Use the `no` form of this command to return to the default value.

**Syntax Description**

- `<polls>` Sets the counter value (Range: 1 to 255).

**Default Values**

By default, the polling counter for the DCE endpoint is set to six polls.

**Functional Notes**

The N391 counter determines how many link integrity polls occur in between full status polls. The number of link integrity polls between full status polls is n - 1, where n represents the full status poll. n can be set to any number between 1 and 255, but the default is used for most applications.

**Usage Examples**

The following example sets the N391 counter for three polls:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-n391dce 3
```
**frame-relay lmi-n391dte <polls>**

Use the `frame-relay lmi-n391dte` command to set the n391 full status polling counter for the DTE endpoint. Typical applications should leave the default value for this timer. Use the `no` form of this command to return to the default value.

**Syntax Description**

- `<polls>`: Sets the counter value (Range: 1 to 255).

**Default Values**

By default, the polling counter for the DTE endpoint is set to six polls.

**Functional Notes**

The N391 counter determines how many link integrity polls occur in between full status polls. The number of link integrity polls between full status polls is \( n - 1 \), where \( n \) represents the full status poll. \( n \) can be set to any number between 1 and 255, but the default is used for most applications.

**Usage Examples**

The following example sets the N391 counter for three polls:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-n391dte 3
```
frame-relay lmi-n392dce <threshold>

Use the frame-relay lmi-n392dce command to set the N392 error threshold for the DCE endpoint. Typical applications should leave the default value for this setting. Use the no form of this command to return to the default value.

Syntax Description

| <threshold> | Sets the threshold value (Range: 1 to 10). |

Default Values

By default, the error threshold for the DCE endpoint is set to three errors.

Functional Notes

If the error threshold is met, the signaling state status is changed to down, indicating a service-affecting condition. This condition is cleared once N393 consecutive error-free events are received. N392 defines the number of errors required in a given event window, while N393 defines the number of polling events in each window.

For example:
If N392=3 and N393=4, then if three errors occur within any four events, the interface is determined inactive.

Usage Examples

The following example sets the N392 threshold for 5 seconds:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-n392dce 5
```
**frame-relay lmi-n392dte <threshold>**

Use the `frame-relay lmi-n392dte` command to set the N392 error threshold for the DTE endpoint. Typical applications should leave the default value for this setting. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <threshold> | Sets the threshold value (Range: 1 to 10). |

**Default Values**

By default, the error threshold for the DTE endpoint is set to three errors.

**Functional Notes**

If the error threshold is met, the signaling state status is changed to down, indicating a service-affecting condition. This condition is cleared once N393 consecutive error-free events are received. N392 defines the number of errors required in a given event window, while N393 defines the number of polling events in each window.

For example:

If N392=3 and N393=4, then if three errors occur within any four events, the interface is determined inactive.

**Usage Examples**

The following example sets the N392 threshold for five errors:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-n392dte 5
```
frame-relay lmi-n393dce <counter>

Use the frame-relay lmi-n393dce to set the N393 LMI monitored event counter for the DCE endpoint. Typical applications should leave the default value for this counter. Use the no form of this command to return to the default value.

Syntax Description

<counter> Sets the counter value (Range: 1 to 10).

Default Values

By default, the LMI monitored event counter for the DCE endpoint is set to four events.

Usage Examples

The following example sets the N393 threshold for five events:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-n393dce 5
**frame-relay lmi-n393dte** `<counter>`

Use the `frame-relay lmi-n393dte` command to set the N393 LMI monitored event counter for the DTE endpoint. Typical applications should leave the default value for this counter. Use the `no` form of this command to return to the default value.

**Syntax Description**

| `<counter>` | Sets the counter value (Range: 1 to 10). |

**Default Values**

By default, the LMI monitored event counter for the DTE endpoint is set to four events.

**Usage Examples**

The following example sets the N393 threshold for five events:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-n393dte 5
```
frame-relay lmi-t391dte <seconds>

Use the `frame-relay lmi-t391dte` command to set the T391 signal polling timer for the DTE endpoint. Typical applications should leave the default value for this timer. Use the `no` form of this command to return to the default value.

**Syntax Description**

| `<seconds>` | Sets the timer value in seconds (Range: 5 to 30). |

**Default Values**

By default, the signal polling timer for the DTE endpoint is set to 10 seconds.

**Functional Notes**

The T391 timer sets the time (in seconds) between polls to the Frame Relay network.

**Usage Examples**

The following example sets the T391 timer for 15 seconds:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-t391dte 15
```
frame-relay lmi-t392dce <seconds>

Use the `frame-relay lmi-t392dce` command to set the T392 polling verification timer for the DCE endpoint. Typical applications should leave the default value for this timer. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <seconds> | Sets the timer value in seconds (Range: 5 to 30). |

**Default Values**

By default, the polling verification timer for the DCE endpoint is set to 10 seconds.

**Functional Notes**

The T392 sets the timeout (in seconds) between polling intervals. This parameter needs to be a few seconds longer than the T391 setting of the attached Frame Relay device.

**Usage Examples**

The following example sets the T392 timer for 15 seconds:

```plaintext
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-t392dce 15
```
frame-relay lmi-type [ansi | auto | cisco | none | q933a]

Use the frame-relay lmi-type command to define the Frame Relay signaling (LMI) type. Use the no form of the command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ansi</td>
<td>Specifies Annex D signaling method (based on ANSI T1.617 standard for Frame Relay).</td>
</tr>
<tr>
<td>auto</td>
<td>Automatically determines signaling type by messages received on the frame circuit.</td>
</tr>
<tr>
<td>cisco</td>
<td>Specifies Cisco LMI signaling method (reserves DLCI 1023).</td>
</tr>
<tr>
<td>none</td>
<td>Turns off signaling on the endpoint. This is used for backup connections.</td>
</tr>
<tr>
<td>q933a</td>
<td>Specifies Annex A signaling method (based on the ITU-T Q.933A frame format for Frame Relay).</td>
</tr>
</tbody>
</table>

Default Values

By default, the Frame Relay signaling type is set to ansi.

Usage Examples

The following example sets the signaling method for the endpoint to cisco:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-type cisco
frame-relay multilink [ack <seconds> | bandwidth-class <class> <threshold> | bid <string> | hello <seconds> | retry <number>]

Use the frame-relay multilink command to enable the Frame Relay multilink interface. When the no form of this command is issued, all configuration options associated with this command and binds made to this interface are removed.

Syntax Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ack &lt;seconds&gt;</strong></td>
<td>Optional. Specifies a wait for acknowledgement time (in seconds) for every bundle link in the bundle. Range: 1 to 180 seconds.</td>
</tr>
<tr>
<td><strong>bandwidth-class</strong></td>
<td>Optional. Specifies the class of operation, placing a minimum limit on the acceptable amount of bandwidth required for a bundle to initialize.</td>
</tr>
<tr>
<td><strong>&lt;class&gt;</strong></td>
<td>Optional. Specifies the class of operation. Range: a to c.</td>
</tr>
<tr>
<td>Class A</td>
<td>A single active link is sufficient for the bundle to be up.</td>
</tr>
<tr>
<td>Class B</td>
<td>All defined bundle links must be active for the bundle to be up.</td>
</tr>
<tr>
<td>Class C</td>
<td>A minimum threshold of links must be active for the bundle to be up.</td>
</tr>
<tr>
<td><strong>&lt;threshold&gt;</strong></td>
<td>Optional. Specifies the minimum number of active bundle links required for a Class C bundle to be in the up state. This option will not be available unless Class C is specified. Range: 1 to 65,535 links.</td>
</tr>
<tr>
<td><strong>bid &lt;string&gt;</strong></td>
<td>Optional. Specifies a bundle ID (up to 48 characters) for the multilink bundle. All hello messages sent on links belonging to the multilink bundle contain the bundle ID. By default, the SROS creates a generic bundle ID for each configured multilink bundle using the following: MFR&lt;interface number&gt; where the interface number corresponds to the interface number of the frame relay interface. For example, if multilink is enabled on frame-relay interface 1, by default the bundle ID is MFR1. Changing the bundle ID causes the multilink connection to go down for renegotiation.</td>
</tr>
<tr>
<td><strong>hello &lt;seconds&gt;</strong></td>
<td>Optional. Specifies the time (in seconds) between hello messages for every bundle link in the bundle. Range: 1 to 180 seconds.</td>
</tr>
<tr>
<td><strong>retry &lt;number&gt;</strong></td>
<td>Optional. Specifies the number of times a bundle link will retransmit a message while waiting for acknowledgement. Range: 1 to 5 times.</td>
</tr>
</tbody>
</table>

Default Values

The default ack value is 4 seconds. The default hello value is 10 seconds. The default <class> value is a. The default retry value is 2.

Functional Note

This command is different from ppp multilink. In ppp multilink, if multiple binds are configured for the PPP interface without multilink PPP being enabled, the first link to bring up LCP will be the only link to actually bind. In Frame Relay multilink, since there is no protocol corresponding to LCP, all binds will be removed and the user will be free to re-issue any bind.
Usage Examples

The following example enables the Frame Relay multilink interface and sets the time between **hello** messages to 45 seconds:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay multilink hello 45

The following example specifies Class B operation:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay multilink bandwidth-class b

The following example specifies Class C operation with a threshold of 5:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay multilink bandwidth-class c 5
hold-queue <queue size> out

Use the hold-queue out command to change the overall size of an interface's WAN output queue. Use the no form of this command to return to the default settings.

Syntax Description

| <queue size> | The total number of packets the output queue can contain before packets are dropped. Range: 16 to 1000. |

Default Values

The default queue size for WFQ is 400. The default queue size for PPP FIFO and Frame Relay round robin is 200.

Usage Examples

The following example sets the overall output queue size to 700:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#hold-queue 700 out
max-reserved-bandwidth <percent>

Use the **max-reserved-bandwidth** command to specify the percentage of interface bandwidth reserved for use in user-defined (priority or class-based) queues. The remainder of the interface bandwidth is reserved for system critical traffic and is not available to user-defined queues. Use the **no** form of this command to restore the default values.

**Caution**

Reserving a portion of the interface bandwidth for system critical traffic is necessary for proper operation. Specifying the entire interface bandwidth for use in user-defined queues can cause undesirable operation.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;percent&gt;</th>
<th>Specifies the percentage of interface bandwidth to make available for user-defined (priority or class-based) queues. Enter an integer 1 to 100.</th>
</tr>
</thead>
</table>

**Default Values**

By default, **max-reserved-bandwidth** is set to **75** which reserves 25 percent of the interface bandwidth for system critical traffic.

**Usage Examples**

The following example specifies 85 percent of the bandwidth on the frame relay 1 interface to be available for use in user-defined queues:

```
(config)#interface frame-relay 1
(config-fr 1)#max-reserved-bandwidth 85
```
qos-policy out <mapname>

Use the `qos-policy out` command to apply a previously-configured QoS map to an interface. Use the `no` form of this command to remove the map from the interface. The `out` keyword specifies that this policy will be applied to outgoing packets.

Syntax Description

| `<map name>` | Specifies the name of a previously-created QoS map (see `qos map <mapname> <sequence number>` on page 451 for more information). |

Default Values

No default value is necessary for this command.

Usage Examples

The following example applies the QoS map `VOICEMAP` to the Frame Relay 1 interface:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#qos-policy out VOICEMAP
```
**snmp trap**

Use the `snmp trap` command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface. Use the `no` form of this command to disable SNMP traps.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces (except virtual Frame Relay interfaces and sub-interfaces) have SNMP traps enabled.

**Usage Examples**

The following example enables SNMP on the virtual Frame Relay interface:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#snmp trap
```
**snmp trap link-status**

Use the `snmp trap link-status` command to control the Simple Network Management Protocol (SNMP) variable `ifLinkUpDownTrapEnable` (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the `no` form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the `ifLinkUpDownTrapEnable` OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The `snmp trap link-status` command is used to control the RFC2863 `ifLinkUpDownTrapEnable` OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the Frame Relay interface:

```
ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#no snmp trap link-status
```
FRAME RELAY SUB-INTERFACE CONFIG COMMAND SET

To activate the Frame Relay Interface Configuration mode, enter the `interface frame-relay` command (and specify a sub-interface) at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port> on page 1304`
- `description <text> on page 1307`
- `do on page 1308`
- `end on page 1309`
- `exit on page 1310`
- `ping <address> on page 1311`
- `shutdown on page 1315`

All other commands for this command set are described in this section in alphabetical order.

- `access-policy <policyname> on page 816`
- `backup commands begin on page 817`
- `bandwidth <value> on page 834`
- `bridge-group <group#> on page 835`
- `crypto map <mapname> on page 836`
- `dynamic-dns on page 838`
- `frame-relay bc <committed burst value> on page 840`
- `frame-relay be <excessive burst value> on page 841`
- `frame-relay fragment <threshold> on page 842`
- `frame-relay interface-dlci <dlci> on page 843`
- `ip commands begin on page 844`
- `lldp receive on page 871`
- `lldp send [management-address l port-description l system-capabilities l system-description l system-name] on page 872`
- `lldp send-and-receive on page 873`
MTU Size on page 874

Spanning-tree commands begin on page 875
**access-policy <policyname>**

Use the `access-policy` command to assign a specified access policy for the inbound traffic on an interface. Use the `no` form of this command to remove an access policy association.

**Syntax Description**

- `<policyname>`: Identifies the configured access policy by alphanumeric descriptor (all access policy descriptors are case-sensitive).

**Default Values**

By default, there are no configured access policies associated with an interface.

**Functional Notes**

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter `access policy <policy name>`. For more details on creating and using access policies, refer to `ip policy-class <policyname>` on page 400.

**Usage Examples**

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the Frame Relay sub-interface labeled 1.16:

Enable the SROS security features:
ProCurve(config)#ip firewall

Create the access list (this is the packet selector):
ProCurve(config)#ip access-list extended InWeb
ProCurve(config-ext-nacl)#permit tcp any host 10.12.5.253 eq 80

Create the access policy that contains the access list **InWeb**:
ProCurve(config)#ip policy-class UnTrusted
ProCurve(config-policy-class)#allow list InWeb

Associate the access list with the Frame Relay sub-interface labeled 1.16:
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#access-policy UnTrusted

---

**Note**

Configured access policies will only be active if the `ip firewall` command has been entered at the Global Configuration mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.
backup auto-backup

Use the backup auto-backup command to configure the sub-interface to automatically attempt a backup upon failure. Use the no form of this command to disable automatic backup on an interface.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically attempt backup upon a failure.

Usage Examples

The following enables automatic backup on the endpoint:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup auto-backup
backup auto-restore

Use the backup auto-restore command to configure the sub-interface to automatically discontinue backup when all network conditions are operational. Use the no form of this command to disable the auto-restore feature.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically restore the primary connection when the failure condition clears.

Usage Examples

The following configures the SROS to automatically restore the primary connection when the failure condition clears:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup auto-restore
backup backup-delay <seconds>

Use the `backup backup-delay` command to configure the amount of time the router will wait after the failure condition is recognized before attempting to backup the link. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <seconds> | Specifies the delay period (in seconds) a failure must be active before the SROS will enter backup operation on the interface (valid range: 10 to 86400 seconds). |

**Default Values**

By default, the `backup backup-delay` is set to 10 seconds.

**Usage Examples**

The following configures the SROS to wait 60 seconds (on an endpoint with an active alarm condition) before attempting backup operation:

```
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup backup-delay 60
```
backup call-mode <role>

Use the backup call-mode command to specify whether the configured backup interfaces answers or originates (or a combination of both) backup calls. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;role&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer</td>
<td>Answers and backs up primary link on failure.</td>
</tr>
<tr>
<td>answer-always</td>
<td>Answers and backs up regardless of primary link state.</td>
</tr>
<tr>
<td>originate</td>
<td>Originates backup call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer</td>
<td>Originates or answers call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer-always</td>
<td>Originates on failure; answers and backs up always.</td>
</tr>
</tbody>
</table>

Default Values

By default, the backup call-mode is set to originate-answer.

Functional Notes

The majority of the configuration for the SROS backup implementation is configured via the backup PPP interface configuration commands. However, the numbers dialed are configured in the primary interface. Full sample configurations follow:

Sample config for remote router (dialing out)

```
hostname "Remote7203dl"
enable password password
!
interface eth 0/1
   ip address 192.168.1.254 255.255.255.0
   no shutdown
!
interface modem 1/3
   no shutdown
!
interface t1 1/1
   coding b8zs
   framing esf
   clock source line
   tdm-group 1 timeslots 1-24
   no shutdown
!
```
interface fr 1 point-to-point
   frame-relay lmi-type ansi
   no shutdown
   bind 1 t1 1/1 1 fr 1
!
interface fr 1.16 point-to-point
   frame-relay interface-dlci 16
   ip address 10.1.1.2 255.255.255.252
   backup call-mode originate
   backup number 5551111 analog ppp 1
   backup number 5552222 analog ppp 1
   no shutdown
!
interface ppp 1
   ip address 172.22.56.1 255.255.255.252
   ppp authentication chap
   username remoterouter password remotepass
   ppp chap hostname localrouter
   ppp chap password procurve
   no shutdown
!
ip route 192.168.2.0 255.255.255.0 172.22.56.2 255.255.255.252
!
line telnet 0 4
   password password

**Sample config for central router (dialing in)**

hostname "Central7203dl"
enable password password
!
interface eth 0/1
   ip address 192.168.100.254 255.255.255.0
   no shutdown
!
interface modem 1/3
   no shutdown
!
interface t1 1/1
   coding b8zs
   framing esf
   clock source line
tdm-group 1 timeslots 1-24
no shutdown
!
interface fr 1 point-to-point
  frame-relay lmi-type ansi
  no shutdown
  bind 1 t1 1/1 1 fr 1
!
interface fr 1.100 point-to-point
  frame-relay interface-dlci 100
  ip address 10.1.1.1 255.255.255.252
  backup call-mode answer
  backup number 555-8888 analog ppp 1
!
interface ppp 1
  ip address 172.22.56.2 255.255.255.252
  ppp authentication chap
  username localrouter password procurve
  ppp chap hostname remoterouter
  ppp chap password remotepass
  no shutdown
!
ip route 192.168.1.0 255.255.255.0 172.22.56.1 255.255.255.252

line telnet 0 4
  password password

Usage Examples
The following configures the SROS to generate backup calls for this endpoint using an analog modem interface (to phone number 555 1111) but never answer calls and specifies ppp 2 as the backup interface:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup call-mode originate
ProCurve(config-fr 1.1)#backup number 555 1111 analog ppp 2

Technology Review
This technology review provides information regarding specific backup router behavior (i.e., when the router will perform backup, where in the configuration the SROS accesses specific routing information, etc.):
### Dialing Out
1. The SROS determines to place an outbound call when either the Layer 1 or Layer 2 has a failure.
2. When placing outbound calls, the SROS matches the number dialed to a PPP interface.
3. When placing the call, the SROS uses the configuration of the related PPP interface for authentication and IP negotiation.
4. If the call fails to connect on the first number dialed, the SROS places a call to the second number if configured. The second number to be dialed references a separate PPP interface.

### Dialing In
1. The SROS receives an inbound call on a physical interface.
2. Caller ID is used to match the **backup number** command to the configured PPP interface.
3. If a match is found, the call connects and the SROS pulls down the primary connection if it is not already in a down state.
4. If no match is found from Caller ID, the call is terminated.
backup connect-timeout <seconds>

Use the `backup connect-timeout` command to specify the number of seconds to wait for a connection after a call is attempted before trying to call again or dialing a different number. It is recommended this number be greater than 60.

**Syntax Description**

| <seconds> | Selects the amount of time in seconds that the router will wait for a connection before attempting another call (valid range: 10 to 300). |

**Default Values**

By default, `backup connect-timeout` is set to 60 seconds.

**Usage Examples**

The following configures the SROS to wait 120 seconds before retrying a failed backup call:

```
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup connect-timeout 120
```
backup force [backup | primary]

Use the backup force command to manually override the automatic backup feature. This can be used to force a link into backup to allow maintenance to be performed on the primary link without disrupting data. Use the no form of this command to return to the normal backup operation state.

Syntax Description

- **backup**: Forces backup regardless of primary link state.
- **primary**: Forces primary link regardless of its state.

Default Values

By default, this feature is disabled.

Usage Examples

The following configures the SROS to force this endpoint into backup:

```
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup force backup
```
backup maximum-retry <attempts>

Use the backup maximum-retry command to select the number of calls the router will make when attempting to backup a link. Use the no form of this command to return to the default state. For more detailed information on Frame Relay backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 820.

Syntax Description

| <attempts> | Selects the number of call retries that will be made after a link failure (valid range: 0 to 15). Setting this value to 0 will allow unlimited retries during the time the network is failed. |

Default Values

By default, backup maximum-retry is set to 0 attempts.

Usage Examples

The following example configures the SROS to retry a backup call four times before considering backup operation not available:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup maximum-retry 4
backup number <digits> [analog | digital-56k | digital 64k] <isdn min chan> <isdn max chan> ppp <interface>

Use the backup number command to configure the phone number and the call type the router will dial upon network failure. Multiple entries can be made for an interface to allow alternate sites to be dialed. For more detailed information on Frame Relay backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 820.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;digits&gt;</th>
<th>Specifies the phone numbers to call when the backup is initiated.</th>
</tr>
</thead>
<tbody>
<tr>
<td>analog</td>
<td>Indicates number connects to an analog modem.</td>
</tr>
<tr>
<td>digital-56k</td>
<td>Indicates number belongs to a digital 56 kbps per DS0 connection.</td>
</tr>
<tr>
<td>digital-64k</td>
<td>Indicates number belongs to a digital 64 kbps per DS0 connection.</td>
</tr>
<tr>
<td>&lt;isdn min chan&gt;</td>
<td>Specifies the minimum number of DS0s required for a digital 56 or 64 kbps connection (Range: 1 to 24).</td>
</tr>
<tr>
<td>&lt;isdn mas chan&gt;</td>
<td>Specifies the maximum number of DS0s desired for a digital 56 or 64 kbps connection (Range: 1 to 24).</td>
</tr>
<tr>
<td>ppp &lt;interface&gt;</td>
<td>Specifies the Frame Relay sub-interface (e.g., fr 1.16) to use when originating or answering using this number.</td>
</tr>
</tbody>
</table>

Default Values

By default, there are no configured backup numbers.

Usage Examples

The following example configures the SROS to dial 704-555-1212 (digital 64 kbps connection) to initiate backup operation on this endpoint using the ppp 1 interface:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup number 7045551212 digital-64k 1 1 ppp 1
backup priority <value>

Use the backup priority command to select the backup priority for this interface. This allows the user to establish the highest priority backup link and ensure that link will override backups attempted by lower priority links. Use the no form of this command to return to the default value. For more detailed information on Frame Relay backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 820.

Syntax Description

| <value> | Sets the relative priority of this link (valid range: 0 to 100). A value of 100 designates the highest priority. |

Default Values

By default, backup priority is set to 50.

Usage Examples

The following example assigns the highest priority to this endpoint:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#backup priority 100
backup randomize-timers

Use the backup randomize-timers command to randomize the call timers to minimize potential contention for resources. Use the no form of this command to return to the default value. For more detailed information on Frame Relay backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 820.

Syntax Description

No subcommands.

Default Values

By default, the SROS does not randomize the backup call timers.

Usage Examples

The following example configures the SROS to randomize the backup timers associated with this endpoint:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup randomize-timers
backup redial-delay <seconds>

Use the backup redial-delay command to configure the delay after an unsuccessful call until the call will be re-tried. For more detailed information on Frame Relay backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 820.

Syntax Description

| <seconds>  | Specifies the delay in seconds between attempting to re-dial a failed backup attempt. Range: 10 to 3600. |

Default Values

By default, backup redial-delay is set to 10 seconds.

Usage Examples

The following example configures a redial delay of 25 seconds on this endpoint:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup redial-delay 25
### backup restore-delay <seconds>

Use the **backup restore-delay** command to configure the amount of time the router will wait after the network is restored before disconnecting the backup link and reverting to the primary. This setting is used to prevent disconnecting the backup link if the primary link is “bouncing” in and out of alarm. For more detailed information on Frame Relay backup functionality, refer to the *Functional Notes* section of the command **backup call-mode <role>** on page 820.

#### Syntax Description

<table>
<thead>
<tr>
<th>&lt;seconds&gt;</th>
<th>Specifies the number of seconds the SROS will wait (after a primary link is restored) before disconnecting backup operation. Range: 10 to 86,400.</th>
</tr>
</thead>
</table>

#### Default Values

By default, **backup restore-delay** is set to 10 seconds.

#### Usage Examples

The following example configures the SROS to wait 30 seconds before disconnecting backup operation and restoring the primary connection for this endpoint:

```
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup restore-delay 30
```
**backup schedule [day | enable-time | disable-time]**

Use the `backup schedule` command to set the time of day that backup will be enabled. Use this command if backup is desired only during normal business hours and on specific days of the week. Use the `no` form of this command to disable backup (as specified). For more detailed information on Frame Relay backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 820.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>Sets the days to allow backup (valid range: Monday through Sunday).</td>
</tr>
<tr>
<td>enable-time</td>
<td>Sets the time of day to enable backup. Time is entered in 24-hour format (00:00).</td>
</tr>
<tr>
<td>disable-time</td>
<td>Sets the time of day to disable backup.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, backup is enabled for all days and times if the `backup auto-backup` command has been issued and the `backup schedule` has not been entered.

**Usage Examples**

The following example enables backup Monday through Friday 8:00 am to 7:00 pm:

```
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#backup schedule enable-time 08:00
ProCurve(config-fr 1.1)#backup schedule disable-time 19:00
ProCurve(config-fr 1.1)#no backup schedule day Saturday
ProCurve(config-fr 1.1)#no backup schedule day Sunday
```
backup shutdown

Use the backup shutdown command to deactivate all backup functionality in the unit. Backup configuration parameters are kept intact, but the unit will not initiate (or respond) to backup sequences in the event of a network outage. Use the no form of this command to reactivate the backup interface. For more detailed information on Frame Relay backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 820.

Syntax Description

No subcommands.

Default Values

By default, all SROS backup interfaces are disabled.

Usage Examples

The following example deactivates the configured backup interface:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#backup shutdown
```
**bandwidth <value>**

Use the `bandwidth` command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the `no` form of this command to restore the default values.

**Syntax Description**

`<value>` Specifies bandwidth in kbps.

**Default Values**

To view default values use the `show interfaces` command.

**Functional Notes**

The `bandwidth` command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the Frame Relay interface to 10 Mbps:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#bandwidth 10000
```
**bridge-group <group#>**

Use the `bridge-group` command to assign an interface to the specified bridge group. This command is supported on all Ethernet interfaces, PPP virtual interfaces, and Frame Relay virtual sub-interfaces. Use the `no` form of this command to remove the interface from the bridge group.

**Syntax Description**

| `<group#>` | Specifies the bridge group number (1 to 255). |

**Default Values**

By default, there are no configured bridge groups.

**Functional Notes**

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (Ethernet to T1 bridge, Ethernet to Frame Relay sub-interface).

**Usage Examples**

The following example assigns the Frame Relay sub-interface labeled 1.16 to bridge group 1:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#bridge-group 1
```
crypto map <mapname>

Use the crypto map command to associate crypto maps with the interface.

**Note**  
When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the VPN Configuration Guide located on the ProCurve SROS Documentation CD provided with your unit.

**Syntax Description**

| <mapname> | Specifies the crypto map name that you wish to assign to the interface. |

**Default Values**

By default, no crypto maps are assigned to an interface.

**Functional Notes**

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the unencrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the diagram above, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only unencrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local side, unencrypted source of the data. The destination information will be the far end, unencrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy class is the far end. The destination information is the local side.

**Usage Examples**

The following example applies all crypto maps with the name **MyMap** to the Frame Relay interface:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#crypto map MyMap
```
**dynamic-dns**

Use the `dynamic-dns` command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the `no` versions of these commands to disable these features. Variations of this command include:

- `dynamic-dns custom <hostname> <minutes>`
- `dynamic-dns dyndns <hostname> <username> <password>`
- `dynamic-dns dyndns-custom <hostname> <username> <password>`
- `dynamic-dns dyndns-static <hostname> <username> <password>`

**Syntax Description**

- `<hostname>` Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).
- `<minutes>` Specifies the intervals in minutes to update the server with information (updates also occur when the interface’s IP address changes regardless of the update intervals).

Refer to **Functional Notes** below for additional argument descriptions.

**Default Values**

No default is necessary for this command.

**Functional Notes**

- **custom** - Constanttime.com’s Custom Dynamic DNS service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows to manage IP address mappings (A records), domain aliases (CNAME records) and mail servers (MX records).

- **dyndns** - The Dynamic DNS service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- **dyndns-custom** - DynDNS.org’s Custom DNS service provides a full DNS solution, giving you complete control over an entire domain name. A web-based interface provides two levels of control over your domain, catering to average or power users. Five globally redundant DNS servers ensure that your domain will always resolve.

A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background, and provides layout and functionality similar to a BIND zone file allowing for the creation of nearly any record type.

Custom DNS can be used with both static and dynamic IPs, and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
**dyndns-static** - The Static DNS service is similar to Dynamic DNS service, in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate through the DNS system. This service is provided for up to five hostnames.

If your IP address doesn't change often or at all, but you still want an easy name to remember it by (without having to purchase your own domain name) Static DNS service is ideal for you.

If you would like to use your own domain name (such as yourname.com) you need Custom DNS service which also provides full dynamic and static IP address support.

**Usage Examples**

The following example sets the Dynamic DNS to **dyndns custom** with hostname host, username user, and password **pass**:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#dynamic-dns dyndns-custom host user pass
```
**frame-relay bc** <committed burst value>

Use the **frame-relay bc** command to set the $b_c$ (committed burst) value for a Frame Relay sublink. The value is in bits. Use the **no** form of this command to return to default.

**Syntax Description**

<committed burst value> Enter the committed burst value (in bits) for the sublink.

**Default Values**

By default, the committed burst value is set to 0 (no limit).

**Functional Notes**

The time interval is always one second, so this can also be considered bits per second. Shaping is performed on a sliding one-second window to make maximum use of configured bandwidth. Note that when both $b_c$ and $b_e$ are non-zero, shaping is performed on the virtual circuit. The circuit is limited to the sum of $b_c$ and $b_e$, and it is recommended that the sum always be greater than 8000.

**Usage Examples**

The following example configures sublink fr 1.16 with a committed burst value of 128000 bits:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#frame-relay bc 128000
```
frame-relay be <excessive burst value>

Use the frame-relay be command to set the $b_e$ (excessive burst) value for a Frame Relay sublink. The value is in bits. Use the no form of this command to return to default.

Syntax Description

<excessive burst value> Specifies the excessive burst value (in bits) for the sublink.

Default Values

By default, the excessive burst value is set to 0 (no limit).

Functional Notes

The time interval is always one second, so this can also be considered bits per second. Shaping is performed on a sliding one-second window to make maximum use of configured bandwidth. Note that when both $b_c$ and $b_e$ are non-zero, shaping is performed on the virtual circuit. The circuit is limited to the sum of $b_c$ and $b_e$, and it is recommended that the sum always be greater than 8000.

Usage Examples

The following example configures the sublink fr 1.16 with an excessive burst value of 64000 bits:

```
ProCurve(config)#interface frame relay 1.16
ProCurve(config-fr 1.16)#frame-relay be 64000
```
**frame-relay fragment** <threshold>

Use the **frame-relay fragment** command to set the FRF.12 fragmentation threshold. Use the **no** form of this command to erase the configured threshold.

**Syntax Description**

| <threshold> | Specifies the fragmentation threshold. Valid fragmentation thresholds are greater than or equal to 64 and less than or equal to 1600. |

**Default Values**

No default value is necessary for this command.

**Functional Notes**

For Frame Relay fragmentation to take effect, rate-limiting must be enabled by setting the committed burst rate and excessive burst rate. See **frame-relay bc <committed burst value>** on page 840 and **frame-relay be <excessive burst value>** on page 841 for more information.

**Usage Examples**

The following example enables FRF.12 fragmentation on a sublink:

```plaintext
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#frame-relay bc 64000
ProCurve(config-fr 1.16)#frame-relay be 16
ProCurve(config-fr 1.16)#frame-relay fragmentation 100
```

The following example disables FRF.12 fragmentation on a sublink:

```plaintext
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#no frame-relay fragment
```
frame-relay interface-dlci <dlci>

Use the frame-relay interface-dlci command to configure the Data Link Connection Identifier (DLCI) for the Frame Relay sub-interface. This setting should match the DLCI supplied by your Frame Relay service provider. Use the no form of this command to remove the configured DLCI.

Syntax Description

<dlci>  Specifies numeric value supplied by your provider.

Default Values

By default, the DLCI is populated with the sub-interface identifier. For example, if configuring the virtual Frame Relay sub-interface labeled fr 1.20, the default DLCI is 20.

Usage Examples

The following example configures a DLCI of 72 for this Frame Relay endpoint:

ProCurve(config)#interface frame relay 1.16
ProCurve(config-fr 1.16)#frame-relay interface-dlci 72
ip access-group <listname> [in | out]

Use the `ip access-group` command to create an access list to be used for packets transmitted on or received from the specified interface. Use the `no` form of this command to disable this type of control.

Syntax Description

- `<listname>` Specifies the IP access list name.
- `in` Enables access control on packets received on the specified interface.
- `out` Enables access control on packets transmitted on the specified interface.

Default Values

By default, these commands are disabled.

Functional Notes

When this command is enabled, the IP destination address of each packet must be processed by the assigned access-list parameters before being passed through to the router system. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the router to only allow Telnet traffic into the Frame Relay sub-interface:

```
ProCurve(config)#interface frame relay 1.16
ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#int frame-relay 1.16
ProCurve(config-fr 1.16)#ip access-group TelnetOnly in
```
ip address dhcp

Use the `ip address dhcp` command to use Dynamic Host Configuration Protocol (DHCP) to obtain an address on the interface. Use the `no` form of this command to remove a configured IP address (using DHCP) and disable DHCP operation on the interface.

```
ip address dhcp [client-id [<interface> | <identifier>] hostname <"string"> ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-id</td>
<td>Optional. Specifies the client identifier used when obtaining an IP address from a DHCP server.</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>Specifying an interface defines the client identifier as the hexadecimal MAC address of the specified interface (including a hexadecimal number added to the front of the MAC address to identify the media type).</td>
</tr>
<tr>
<td>For example, specifying the <code>client-id ethernet 0/1</code> (where the Ethernet interface has a MAC address of 0012.7991.1150) defines the client identifier as 01:00:12:79:91:11:50 (where 01 defines the media type as Ethernet). Refer to <code>hardware-address &lt;hardware-address&gt; &lt;type&gt;</code> on page 1287 for a detailed listing of media types.</td>
<td></td>
</tr>
<tr>
<td>&lt;identifier&gt;</td>
<td>Specifies a custom client-identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters).</td>
</tr>
<tr>
<td>hostname</td>
<td>Specifies a text string (to override the global router name) to use as the name in the DHCP option 12 field.</td>
</tr>
<tr>
<td>&lt;&quot;string&quot;&gt;</td>
<td>String (encased in quotation marks) of up to 35 characters to use as the name of the host for DHCP operation.</td>
</tr>
<tr>
<td>no-default-route</td>
<td>Keyword used to specify that the SROS not install the default-route obtained via DHCP.</td>
</tr>
<tr>
<td>no-domain-name</td>
<td>Keyword used to specify that the SROS not install the domain-name obtained via DHCP.</td>
</tr>
<tr>
<td>no-nameservers</td>
<td>Keyword used to specify that the SROS not install the DNS servers obtained via DHCP.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-id</td>
<td>Optional. By default, the client identifier is populated using the following formula: TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS</td>
</tr>
<tr>
<td>Where TYPE specifies the media type in the form of one hexadecimal byte (refer to <code>hardware-address &lt;hardware-address&gt; &lt;type&gt;</code> on page 1812 for a detailed listing of media types), and the MAC ADDRESS is the Media Access Control (MAC) address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to Ethernet 0/1 is used in this field).</td>
<td></td>
</tr>
</tbody>
</table>
INTERFACE SPECIFIC INFO is only used for Frame Relay interfaces and can be determined using the following:

FR_PORT#: Q.922 ADDRESS

Where the FR_PORT# specifies the label assigned to the virtual Frame Relay interface using four hexadecimal bytes. For example, a virtual Frame Relay interface labeled 1 would have a FR_PORT# of 00:00:00:01.

The Q.922 ADDRESS field is populated using the following:

```
  8  7  6  5  4  3  2  1
```

<table>
<thead>
<tr>
<th>DLCI (high order)</th>
<th>C/R</th>
<th>EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLCI (lower)</td>
<td>FECN</td>
<td>BECN</td>
</tr>
</tbody>
</table>

Where the FECN, BECN, C/R, DE, and high order extended address (EA) bits are assumed to be 0 and the lower order EA bit is set to 1.

The following list provides a few example DLCIs and associated Q.922 address:

- DLCI (decimal) / Q.922 address (hex)
  - 16 / 0x0401
  - 50 / 0x0C21
  - 60 / 0x0CC1
  - 70 / 0x1061
  - 80 / 0x1401

hostname <"string"> Optional. By default, the host name is the name configured using the Global Configuration hostname command.

Functional Notes

DHCP allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the client-id and hostname fields.

Usage Examples

The following example enables DHCP operation on the virtual Frame Relay interface (labeled 1.16):

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip address dhcp
```
### ip address <address> <mask> [secondary]

Use the `ip address` command to define an IP address on the specified interface. Use the optional `secondary` keyword to define a secondary IP address. Use the `no` form of this command to remove a configured IP address.

#### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).</td>
</tr>
<tr>
<td><code>&lt;mask&gt;</code></td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
<tr>
<td><code>secondary</code></td>
<td>Optional keyword used to configure a secondary IP address for the specified interface.</td>
</tr>
</tbody>
</table>

#### Default Values

By default, there are no assigned IP addresses.

#### Functional Notes

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

#### Usage Examples

The following example configures a secondary IP address of 192.168.72.101/30:

```plaintext
(ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip address 192.168.72.101 255.255.255.252 secondary
```

To view configured secondary IP address(es), use the `show running-config` command. Secondary IP address information is not available through other `show` commands.
ip dhcp [release | renew]

Use the ip dhcp command to release or renew the DHCP IP address. This command is only applicable when using DHCP for IP address assignment.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>release</td>
<td>Releases DHCP IP address.</td>
</tr>
<tr>
<td>renew</td>
<td>Renews DHCP IP address.</td>
</tr>
</tbody>
</table>

Default Values

No default values required for this command.

Usage Examples

The following example releases the IP DHCP address for the virtual interface:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip dhcp release
**ip helper-address <address>**

Use the **ip helper-address** command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the **no** form of this command to disable forwarding packets.

**Note**  
The **ip helper** command must be used in conjunction with the **ip forward-protocol** command to configure the SROS to forward UDP broadcast packets. Refer to **ip forward-protocol udp <port number>** on page 387 for more information.

<table>
<thead>
<tr>
<th>Syntax Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
</tr>
</tbody>
</table>

**Default Values**

By default, broadcast UDP packets are not forwarded.

**Functional Notes**

When used in conjunction with the **ip forward-protocol** command, the **ip helper-address** feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign a helper-address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the **ip forward-protocol** command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the **ip helper-address** feature:

1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the **ip forward-protocol** command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

**Usage Examples**

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.33.5.99:

```bash
ProCurve(config)#interface frame-relay 1.16
ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip helper-address 192.168.5.99
```
ip igmp

Use the **ip igmp** command to configure multicasting-related functions for the interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>immediate-leave</td>
<td>Specifies that if only one host (or IGMP snooping switch) is connected to the interface, when a leave is received, multicast of that group is immediately terminated as opposed to sending a group query and timing out the group if no device responds. Works in conjunction with <strong>ip igmp last-member-query-interval</strong>. Applies to all groups when configured.</td>
</tr>
<tr>
<td>last-member-query-interval &lt;milliseconds&gt;</td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group’s multicast packets to the interface. Range: 100 to 65535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td>querier-timeout &lt;seconds&gt;</td>
<td>Specifies the number of seconds that the router waits after the current querier’s last query before it takes over as querier (IGMP V2). Range: 60-300 seconds. Default: 2x the <strong>query-interval</strong> value.</td>
</tr>
<tr>
<td>query-interval &lt;seconds &gt;</td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td>query-max-response-time &lt;seconds&gt;</td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td>static-group &lt;group-address&gt;</td>
<td>Configures the router’s interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>version [1</td>
<td>2]</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example sets the query message interval on the interface to 200 milliseconds:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip igmp last-member-query-interval 200
```
**ip mcast-stub downstream**

Use the `ip mcast-stub downstream` command to enable multicast forwarding and IGMP (router mode) on an interface and place it in multicast stub downstream mode. Use the no form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub upstream` commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub downstream` on page 851 for more information.

**Usage Examples**

The following example enables multicast forwarding and IGMP on the interface:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip mcast-stub downstream
ip mcast-stub fixed

Use the `ip mcast-stub fixed` command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the `no` form of this command to disable this mode.

Syntax Description

No subcommands.

Default Values

By default, this command is disabled.

Functional Notes

Multicast routing must be enabled prior to setting `ip mcast-stub fixed` on the selected interface. Also, use the `ip igmp static-group <A.B.C.D>` command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

Usage Examples

The following example enables multicast traffic forwarding and IGMP on the interface:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip mcast-stub fixed
```
ip mcast-stub helper-enable

Use the ip mcast-stub helper-enable command to assign the ip mcast-stub helper-address as the IGMP proxy. Use the no form of this command to disable.

Syntax Description
No subcommands.

Default Values
By default, this command is disabled.

Functional Notes
This command is used in IP multicast stub applications in conjunction with the ip mcast-stub helper-address, ip mcast-stub upstream, and ip mcast-stub downstream commands. When enabled, the interface becomes a helper forwarding interface. The IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the unit to perform as an IGMP proxy. Refer to ip mcast-stub helper-address <ip address> on page 397 and ip mcast-stub downstream on page 851 for more information.

Usage Examples
The following example sets the helper address as the IGMP proxy:

ProCurve(config)#interface frame relay 1.16
ProCurve(config-fr 1.16)#ip mcast-stub helper-enable
**ip mcast-stub upstream**

Use the **ip mcast-stub upstream** command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the **no** form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the **ip mcast-stub helper-address** and **ip mcast-stub downstream** commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router's unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface. Refer to **ip mcast-stub helper-address <ip address>** on page 397 and **ip mcast-stub downstream** on page 851 for more information.

**Usage Examples**

The following example enables multicast forwarding on the interface:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip mcast-stub upstream
```
**ip ospf**

Use the `ip ospf` command to customize OSPF settings (if needed).

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication-key</td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td>cost <code>&lt;value&gt;</code></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32767.</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td>message-digest-key <code>&lt;keyid&gt; md5 </code>&lt;key&gt;`</td>
<td>Configures OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td>priority <code>&lt;value&gt;</code></td>
<td>Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32,767.</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32,767.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>5 seconds</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>1 second</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>10 seconds</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>40 seconds</td>
</tr>
</tbody>
</table>

**Usage Example**

The following example sets the maximum number of seconds allowed between hello packets to 25,000:

```bash
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip ospf dead-interval 25000
```
ip ospf authentication [message-digest | null]

Use the `ip ospf authentication` command to authenticate an interface that is performing OSPF authentication.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-digest</td>
<td>Optional. Selects message-digest authentication type.</td>
</tr>
<tr>
<td>null</td>
<td>Optional. Specifies that no authentication be used.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this is set to null (meaning no authentication is used).

**Usage Examples**

The following example specifies that no authentication will be used on the Frame Relay interface:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip ospf authentication null
```
ip ospf network [broadcast | point-to-point]

Use the `ip ospf network` command to specify the type of network on this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcast</td>
<td>Sets the network type for broadcast.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Sets the network type for point-to-point.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

**Functional Notes**

A point-to-point network will not elect designated routers.

**Usage Examples**

The following example designates a broadcast network type:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip ospf network broadcast
```
ip pim sparse-mode

Use the **ip pim sparse-mode** command to enable PIM Sparse Mode on the interface. Use the **no** form of this command to disable PIM Sparse Mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, PIM Sparse Mode is disabled for all interfaces.

**Functional Notes**

PIM Sparse Mode is a multicast routing protocol that makes use of the unicast forwarding table. PIM-systems builds unidirectional shared trees rooted at a Rendezvous Point (RP) for a multicast group or a shortest-path tree rooted at a specific source for a multicast group.

**Usage Examples**

The following example enables PIM sparse-mode on the Frame Relay interface 1:

```plaintext
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#ip pim sparse-mode
```
**ip pim-sparse dr-priority <value>**

Use the **ip pim-sparse dr-priority** command to specify the priority of this PIM interface for use when selecting the designated router (DR). Use the **no** form of this command to return to the default value.

**Syntax Description**

| <value> | Specifies the priority of this interface (to be used when determining the DR). Valid range is 1 to 4,294,967,295. |

**Default Values**

By default, the priority of all PIM interfaces is 1.

**Functional Notes**

Interfaces advertise their configured priority values in the hello messages transmitted on the interface. Routers use the priority values to determine the appropriate DR. The router on the network segment with the highest priority is selected as the DR. If a hello message is received on the interface from a router on the network segment and it does not contain a priority, the entire network segment defaults to DR selection based on IP addresses instead of priority. In this instance, the DR is selected as the router on the network segment that has the highest IP address. ProCurve Secure Routers will always include a priority in all transmitted hello messages. If no priority is specifically designated by the user, the priority is set as the default of 1.

**Usage Examples**

The following example specifies a priority of 100 on the Frame Relay interface 1.1:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#ip pim-sparse dr-priority 100
ip pim-sparse hello-timer <time>

Use the ip pim-sparse hello-timer command to set the time interval at which periodic hello messages are transmitted out the interface. Each PIM interface has an independent hello-timer. Use the no form of this command to return to the default value.

Syntax Description

<time> Specifies the interval (in seconds) at which periodic hellos are sent out the interface. Valid range is 10 to 3600 seconds.

Default Values

By default, hellos are transmitted on PIM interfaces every 60 seconds.

Functional Notes

Hello messages are used to inform neighbors of a router’s presence. Hello messages normally generate a small amount of traffic on an interface. Setting the hello-timer to a small interval increases the amount of hellos sent (thus increasing the amount of traffic). Set the hello-timer to a reasonable value, taking into consideration the bandwidth available on the interface.

Usage Examples

The following example specifies hellos to be sent on the Frame Relay interface every 3600 seconds:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#ip pim-sparse hello-timer 3600
ip pim-sparse nbr-timeout <time>

Use the **ip pim-sparse nbr-timeout** command to specify the interval the PIM interface waits before declaring that the neighbor is not present. Use the **no** form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th><strong>&lt;time&gt;</strong></th>
<th>Specifies the time interval (in seconds) the PIM interface waits before a neighbor is considered not present. Valid range is 30 to 10,800 seconds.</th>
</tr>
</thead>
</table>

**Default Values**

By default, the **nbr-timeout** is set to 105 seconds.

**Usage Examples**

The following example specifies a wait interval of 360 seconds on the Frame Relay interface 1.1:

```
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#ip pim-sparse nbr-timeout 360
```
**ip pim-sparse override-interval** *<time>*

Use the `ip pim-sparse override-interval` command to specify the delay interval after a join/prune in which another router on the LAN may override the join/prune. Use the `no` form of this command to return to the default value.

**Syntax Description**

| *<time>* | Specifies the delay interval (in milliseconds) after a join/prune in which another router on the LAN may override the join/prune. Valid range is 0 to 65,535 milliseconds. |

**Default Values**

By default, the `override-interval` is set to 2500 milliseconds.

**Usage Examples**

The following example sets the delay interval for join/prune overrides to 3000 milliseconds on the Frame Relay interface 1.1:

```
ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#ip pim-sparse override-interval 3000
```
ip pim-sparse propagation-delay <time>

Use the `ip pim-sparse propagation-delay` command to specify the expected propagation delay for join/prune messages. Set the propagation delay (in milliseconds) to estimate the amount of delay found in the local link. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time>   | Specifies the expected propagation delay in the local link in milliseconds. Valid range is 0 to 32,767 milliseconds. |

**Default Values**

By default, the `propagation-delay` is set to 500 milliseconds.

**Usage Examples**

The following example sets the expected propagation delay to 300 milliseconds on the Frame Relay interface 1.1:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#ip pim-sparse propagation-delay 300
ip policy route-map <mapname>

Use the ip policy route-map command to assign a policy route map to this interface. Use the no form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

Syntax Description

<mapname> Specifies the route map to associate with this interface.

Default Values

By default, policy-based routing is disabled for all interfaces.

Usage Examples

The following example associates the route map named MyMap with Frame Relay interface 1.1:

ProCurve(config)#interface frame-relay 1.1
ProCurve(config-fr 1.1)#ip policy route-map MyMap
**ip proxy-arp** `<address> <subnet mask>`

Use the `ip proxy-arp` to enable proxy Address Resolution Protocol (ARP) on the interface. Use the `no` form of this command to disable this feature.

**Syntax Description**

| `<address>` | Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101). |
| `<subnet mask>` | Specifies the subnet mask that corresponds to the listed IP address. |

**Default Values**

By default, proxy ARP is enabled.

**Functional Notes**

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all proxy ARP requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

**Usage Examples**

The following enables proxy ARP on the Frame Relay sub-interface:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip proxy-arp
```
ip rip receive version [1 | 2]

Use the `ip rip receive version` command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accepts only received RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Accepts only received RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip receive version` command to specify a RIP version that will override the `version` (in the Router RIP) configuration.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures a Frame Relay sub-interface to accept only RIP version 2 packets:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip rip receive version 2
```
**ip rip send version [1 | 2]**

Use the `ip rip send version` command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transmits only RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Transmits only RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces transmit RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip send version` to specify a RIP version that will override the `version` (in the Router RIP) configuration.

The SROS only transmits one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures a Frame Relay sub-interface to transmit only RIP version 2 packets:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip rip send version 2
```
**ip rip summary-address** `<network address> <network mask>`

Use the `ip rip summary-address` command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the `no` form of this command to disable this mode.

**Syntax Description**

<table>
<thead>
<tr>
<th><strong>&lt;network address&gt;</strong></th>
<th>Specifies the IP address of the network to be summarized.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>&lt;network mask&gt;</strong></td>
<td>Specifies the network mask to be applied to the specific network to be summarized.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no manual summarization is applied by RIP.

**Functional Notes**

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the `ip rip summary-address` command. This command is only effective if RIP version 2 is configured.

**Usage Examples**

The following example enables manual summarization on the specified IP address:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip rip summary-address 10.10.123.0 255.255.255.0
```
**ip route-cache**

Use the `ip route-cache` command to enable route caching on the interface. Use the `no` form of this command to disable route caching and return to process switching mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, route caching is enabled on all interfaces.

**Functional Notes**

Route caching allows an IP interface to provide optimum performance when processing IP traffic.

**Usage Examples**

The following example enables route caching on a Frame Relay sub-interface:

```plaintext
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip route-cache
```
ip unnumbered <interface>

Use the **ip unnumbered** command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the **no** form of this command to remove the unnumbered configuration.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt;</td>
<td>Specifies the interface (in the format type slot/port) that contains the IP address to use as the source address for all packets transmitted on this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces are configured to use a specified IP address (using the **ip address** command).

**Functional Notes**

If **ip unnumbered** is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying **ip unnumbered eth 0/1** while in the Frame Relay Sub-Interface Configuration mode configures the Frame Relay sub-interface to use the IP address assigned to the Ethernet interface for all IP processing. In addition, the SROS uses the specified interface information when sending route updates over the unnumbered interface.

**Usage Examples**

The following example configures the Frame Relay interface (labeled **frame-relay 1.16**) to use the IP address assigned to the Ethernet interface (**eth 0/1**):

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#ip unnumbered eth 0/1
```
lldp receive

Use the lldp receive command to allow LLDP packets to be received on this interface.

Syntax Description

No subcommands.

Default Values

By default, all interfaces are configured to send and receive LLDP packets.

Usage Examples

The following example configures the Frame Relay sub-interface to receive LLDP packets:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#lldp receive
lldp send [management-address \ port-description \ system-capabilities \ system-description \ system-name]

Use the \lldp\ send command to configure this interface to transmit LLDP packets or to control the types of information contained in the LLDP packets transmitted by this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>management-address</td>
<td>Enables transmission of management address information on this interface.</td>
</tr>
<tr>
<td>port-description</td>
<td>Enables transmission of port description information on this interface.</td>
</tr>
<tr>
<td>system-capabilities</td>
<td>Enables transmission of this device’s system capabilities on this interface.</td>
</tr>
<tr>
<td>system-description</td>
<td>Enables transmission of this device’s system description on this interface.</td>
</tr>
<tr>
<td>system-name</td>
<td>Enables transmission of this device’s system name on this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces are configured to transmit and receive LLDP packets of all types.

**Functional Notes**

Individual LLDP information can be enabled or disabled using the various forms of the \lldp\ send command. For example, use the \lldp\ send and-receive command to enable transmit and receive of all LLDP information. Then use the no \lldp\ send port-description command to prevent LLDP from transmitting port description information.

**Usage Examples**

The following example configures the Frame Relay sub-interface to transmit LLDP packets containing all enabled information types:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#lldp send
```
lldp send-and-receive

Use the lldp send-and-receive command to configure this interface to transmit and receive LLDP packets.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces are configured to transmit and receive LLDP packets of all types.

**Functional Notes**

Individual LLDP information can be enabled or disabled using the various forms of the lldp send command. For example, use the lldp send-and-receive command to enable transmit and receive of all LLDP information. Then use the no lldp send port-description command to prevent LLDP from transmitting port description information.

**Usage Examples**

The following example configures the Frame Relay sub-interface (fr 1.16) to transmit and receive LLDP packets containing all information types:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#lldp send-and-receive
**mtu <size>**

Use the `mtu` command to configure the maximum transmit unit (MTU) size for the active interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

Configure the window size for transmitted packets. The valid ranges for the various interfaces are listed below:

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>64 to 256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>64 to 18,190</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>64 to 1500</td>
</tr>
</tbody>
</table>

**Default Values**

The default values for the various interfaces are listed below:

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>1500</td>
</tr>
</tbody>
</table>

**Functional Notes**

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

**Usage Examples**

The following example specifies an MTU of 1200 on the Frame Relay interface:

```
ProCurve(config)#interface frame relay 1.16
ProCurve(config-fr 1.16)#mtu 1200
```
spanning-tree bpdufilter [enable | disable]

Use the spanning-tree bpdufilter command to block BPDUs from being transmitted and received on this interface. To return to the default value, use the no form of this command.

Syntax Description

<table>
<thead>
<tr>
<th>enable</th>
<th>Enables the BPDU filter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>disable</td>
<td>Disables the BPDU filter.</td>
</tr>
</tbody>
</table>

Default Values

By default, this command is set to disable.

Functional Notes

The purpose of this command is to remove a port from participation in the spanning tree. This might be beneficial while debugging a network setup. It normally should not be used in a live network.

Usage Examples

The following example enables the BPDU filter on the interface:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#spanning-tree bpdufilter enable
spanning-tree bpduguard [enable | disable]

Use the `spanning-tree bpduguard` command to block BPDUs from being received on this interface. To return to the default value, use the `no` form of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables the BPDU block.</td>
</tr>
<tr>
<td>disable</td>
<td>Disables the BPDU block.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is set to disable.

**Usage Examples**

The following example enables the BPDU guard on the interface:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#spanning-tree bpduguard enable
```
spanning-tree edgeport

Use the spanning-tree edgeport command to set this interface to be an edgeport. This command overrides the Global setting. Use the no form of this command to return the interface to normal operation (non-edgeport).

Syntax Description

No subcommands.

Default Values

By default, this command is set to disable.

Usage Examples

The following example configures the interface to be an edgeport:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#spanning-tree edgeport

An individual interface can be configured to not be considered an edgeport. For example:
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#no spanning-tree edgeport
spanning-tree link-type [auto | point-to-point | shared]

Use the `spanning-tree link-type` command to configure the spanning-tree protocol link type for an interface. To return to the default value, use the `no` form of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto</td>
<td>Link type is determined by the port's duplex settings.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Link type is manually set to point-to-point, regardless of duplex settings.</td>
</tr>
<tr>
<td>shared</td>
<td>Link type is manually set to shared, regardless of duplex settings.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, a port is set to auto.

**Functional Notes**

This command overrides the default link-type setting determined by the duplex of the individual port. By default a port configured for half-duplex is set to `shared` link type, and a port configured for full-duplex is set to `point-to-point` link type. Setting the link type manually overrides the default and forces the port to use the specified link type. Using the `link-type auto` command, restores the convention of determining link type based on duplex settings.

**Usage Examples**

The following example forces the link type to point-to-point, even if the port is configured to be half-duplex:

```
ProCurve(config)# bridge 1 protocol ieee
ProCurve(config)# interface frame-relay 1.16
ProCurve(config-fr 1.16)# spanning-tree link-type point-to-point
```

**Technology Review**

Rapid transitions are possible in rapid spanning-tree protocol (RSTP) by taking advantage of point-to-point links (a port is connected to exactly one other bridge) and edge-port connections (a port is not connected to any additional bridges). Setting the link type to `auto` allows the spanning tree to automatically configure the link type based on the duplex of the link. Setting the link type to `point-to-point` allows a half-duplex link to act as if it were a point-to-point link.
spanning-tree path-cost <value>

Use the spanning-tree path-cost command to assign a cost to a bridge group that is used when computing the spanning-tree root path. To return to the default path-cost value, use the no form of this command.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;value&gt;</td>
<td>Assigns the bridge interface a path cost value to be used in spanning calculations (valid range: 0 to 65,535).</td>
</tr>
</tbody>
</table>

Default Values

By default, the path-cost value is set at 19.

Functional Notes

The specified value is inversely proportional to the likelihood the bridge interface will be chosen as the root path. Set the path-cost value lower to increase the chance the interface will be the root. To obtain the most accurate spanning-tree calculations, develop a system for determining path costs for links and apply it to all bridged interfaces.

Usage Examples

The following example assigns a path cost of 100 for bridge group 17 on a Frame Relay sub-interface:

ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#spanning-tree path-cost 100

Technology Review

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.
**spanning-tree port-priority** `<value>`

Use the `spanning-tree port-priority` command to select the priority level of a port associated with a bridge. To return to the default bridge-group priority value, use the `no` version of this command.

**Syntax Description**

| `<value>` | Priority value for the bridge group; the lower the value, the higher the priority (valid range: 0 to 255). |

**Default Values**

By default, the bridge-group priority value is set at 128.

**Functional Notes**

The only time that this priority level is used is when two interfaces with a path to the root have equal cost. At that point, the level set in this command will determine which port the bridge will use. Set the priority value lower to increase the chance the interface will be used.

**Usage Examples**

The following example sets the maximum priority on the Frame Relay sub-interface labeled 1.16 in bridge group 17:

```
ProCurve(config)#interface frame-relay 1.16
ProCurve(config-fr 1.16)#spanning-tree port-priority 0
```
HDLC COMMAND SET

To activate the HDLC mode, enter the `interface hdlc` command at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `access-policy <policyname>` on page 883
- `alias link<"text">` on page 884
- `backup commands` begin on page 885
- `bandwidth <value>` on page 902
- `bridge-group <group#>` on page 903
- `crypto map <mapname>` on page 904
- `dynamic-dns` on page 906
- `fair-queue <threshold>` on page 908
- `hold-queue <queue size> out` on page 909
- `ip commands` begin on page 910
- `keepalive <seconds>` on page 934
- `lldp receive` on page 935
- `lldp send [management-address l port-description l system-capabilities l system-description l system-name]` on page 936
- `lldp send-and-receive` on page 937
max-reserved-bandwidth <percent> on page 938
mtu <size> on page 939
qos-policy out <mapname> on page 940
snmp trap link-status on page 941
access-policy <policyname>

Use the access-policy command to assign a specified access policy for the inbound traffic on an interface. Use the no form of this command to remove an access policy association.

**Syntax Description**

<policyname> Identifies the configured access policy by alphanumeric descriptor (all access policy descriptors are case-sensitive).

**Default Values**

By default, there are no configured access policies associated with an interface.

**Functional Notes**

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter access policy <policy name>. For more details on creating and using access policies, refer to ip policy-class <policyname> on page 400.

**Usage Examples**

The following example associates the access policy UnTrusted (to allow inbound traffic to the Web server) to the HDLC interface labeled 1:

Enable the SROS security features:
ProCurve(config)#ip firewall

Create the access list (this is the packet selector):
ProCurve(config)#ip access-list extended InWeb
ProCurve(config-ext-nacl)#permit tcp any host 10.12.5.253 eq 80

Create the access policy that contains the access list InWeb:
ProCurve(config)#ip policy-class UnTrusted
ProCurve(config-policy-class)#allow list InWeb

Associate the access list with the interface:
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#access-policy UnTrusted

**Note** Configured access policies will only be active if the ip firewall command has been entered at the Global Configuration mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.
Each configured HDLC interface (when referenced using SNMP) contains a link (physical port) and a bundle (group of links). RFC 1471 (for Link Connection Protocol) provides an interface table to manage lists of bundles and associated links. Use the `alias link` command to provide the management station an identifying description for each link (HDLC physical).

**Syntax Description**

| `<text>` | Describes the interface (for SNMP) by alphanumeric character string (must be encased in quotation marks). |

**Default Values**

By default, the HDLC identification string appears as empty quotes (" ").

**Functional Notes**

The `alias link` string should be used to uniquely identify an HDLC link. Enter a string that clearly identifies the link.

**Usage Examples**

The following example defines a unique character string for the virtual HDLC interface (1):

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#alias link "HDLC_link_1"
```
backup auto-backup

Use the backup auto-backup command to configure the interface to automatically attempt a backup upon failure. Use the no form of this command to disable auto-backup functionality.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically attempt backup upon a failure.

Usage Examples

The following enables automatic backup on the endpoint:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup auto-backup
backup auto-restore

Use the `backup auto-restore` command to configure the interface to automatically discontinue backup when all network conditions are operational. Use the `no` form of this command to disable the auto-restore feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, all backup endpoints will automatically restore the primary connection when the failure condition clears.

**Usage Examples**

The following configures the SROS to automatically restore the primary connection when the failure condition clears:

```plaintext
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup auto-restore
```
**backup backup-delay <seconds>**

Use the `backup backup-delay` command to configure the amount of time the router will wait after the failure condition is recognized before attempting to backup the link. Use the `no` form of this command to return to the default value.

**Syntax Description**

<seconds>  Specifies the delay period (in seconds) a failure must be active before the SROS will enter backup operation on the interface (valid range: 10 to 86400 seconds).

**Default Values**

By default, the `backup backup-delay` is set to 10 seconds.

**Usage Examples**

The following configures the SROS to wait 60 seconds (on an endpoint with an active alarm condition) before attempting backup operation:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup backup-delay 60
```
backup call-mode <role>

Use the backup call-mode command to specify whether the configured backup interfaces answers or originates (or a combination of both) backup calls. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;role&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer</td>
<td>Selects the role the router will take in backup of this sub-interface.</td>
</tr>
<tr>
<td>answer-always</td>
<td>Answers and backs up primary link on failure.</td>
</tr>
<tr>
<td>originate</td>
<td>Originates backup call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer</td>
<td>Originates or answers call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer-always</td>
<td>Originates on failure; answers and backs up always.</td>
</tr>
</tbody>
</table>

Default Values

By default, the backup call-mode is set to originate-answer.

Functional Notes

The majority of the configuration for the SROS backup implementation is configured via the backup PPP interface configuration commands. However, the numbers dialed are configured in the primary interface. Full sample configurations follow:

Sample config for remote router (dialing out)

hostname "Remote7203dl"
enable password password
!
interface eth 0/1
  ip address 192.168.1.254 255.255.255.0
  no shutdown
!
interface modem 1/3
  no shutdown
!
interface t1 1/1
  coding b8zs
  framing esf
  clock source line
  tdm-group 1 timeslots 1-24
  no shutdown
!
interface fr 1 point-to-point
    frame-relay lmi-type ansi
    no shutdown
    bind 1 t1 1/1 1 fr 1

interface fr 1.16 point-to-point
    frame-relay interface-dlci 16
    ip address 10.1.1.2 255.255.255.252
    backup call-mode originate
    backup number 5551111 analog ppp 1
    backup number 5552222 analog ppp 1
    no shutdown

interface ppp 1
    ip address 172.22.56.1 255.255.255.252
    ppp authentication chap
    username remoterouter password remotepass
    ppp chap hostname localrouter
    ppp chap password procurve
    no shutdown

ip route 192.168.2.0 255.255.255.0 172.22.56.2 255.255.255.252

line telnet 0 4
    password password

Sample config for central router (dialing in)

hostname "Central7203dl"
enable password password

interface eth 0/1
    ip address 192.168.100.254 255.255.255.0
    no shutdown

interface modem 1/3
    no shutdown

interface t1 1/1
    coding b8zs
    framing esf
    clock source line
tdm-group 1 timeslots 1-24
no shutdown
!
interface fr 1 point-to-point
   frame-relay lmi-type ansi
   no shutdown
   bind 1 t1 1/1 1 fr 1
!
interface fr 1.100 point-to-point
   frame-relay interface-dlci 100
   ip address 10.1.1.1 255.255.255.252
   backup call-mode answer
   backup number 555-8888 analog ppp 1
!
interface ppp 1
   ip address 172.22.56.2 255.255.255.252
   ppp authentication chap
   username localrouter password procurve
   ppp chap hostname remoterruter
   ppp chap password remotepass
   no shutdown
!
ip route 192.168.1.0 255.255.255.0 172.22.56.1 255.255.255.252

line telnet 0 4
   password password

Usage Examples
The following configures the SROS to generate backup calls for this endpoint using an analog modem interface (to phone number 555 1111) but never answer calls and specifies ppp 2 as the backup interface:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup call-mode originate
ProCurve(config-hdlc 1)#backup number 555 1111 analog ppp 2

Technology Review
This technology review provides information regarding specific backup router behavior (i.e., when the router will perform backup, where in the configuration the SROS accesses specific routing information, etc.):

Dialing Out
1. The SROS determines to place an outbound call when either the Layer 1 or Layer 2 has a failure.
2. When placing outbound calls, the SROS matches the number dialed to a PPP interface.
3. When placing the call, the SROS uses the configuration of the related PPP interface for authentication and IP negotiation.
4. If the call fails to connect on the first number dialed, the SROS places a call to the second number if configured. The second number to be dialed references a separate PPP interface.

Dialing In
1. The SROS receives an inbound call on a physical interface.
2. Caller ID is used to match the backup number command to the configured PPP interface.
3. If a match is found, the call connects and the SROS pulls down the primary connection if it is not already in a down state.
4. If no match is found from Caller ID, the call is terminated.
backup connect-timeout <seconds>

Use the backup connect-timeout command to specify the number of seconds to wait for a connection after a call is attempted before trying to call again or dialing a different number. Recommended value is greater than 60. For more detailed on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 888.

Syntax Description

| <seconds> | Selects the amount of time in seconds that the router will wait for a connection before attempting another call (valid range: 10 to 300). |

Default Values

By default, backup connect-timeout is set to 60 seconds.

Usage Examples

The following configures the SROS to wait 120 seconds before retrying a failed backup call:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup connect-timeout 120
backup force [backup | primary]

Use the `backup force` command to manually override the automatic backup feature. This can be used to force a link into backup to allow maintenance to be performed on the primary link without disrupting data. Use the `no` form of this command to return to the normal backup operation state. For more detailed on backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 888.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backup</td>
<td>Forces backup regardless of primary link state.</td>
</tr>
<tr>
<td>primary</td>
<td>Forces primary link regardless of its state.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this feature is disabled.

**Usage Examples**

The following configures the SROS to force this endpoint into backup:

```plaintext
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup force backup
```
backup maximum-retry <attempts>

Use the `backup maximum-retry` command to select the number of calls the router will make when attempting to backup a link. Use the `no` form of this command to return to the default state. For more detailed information on backup functionality, refer to the Functional Notes section of the command `backup call-mode <role>` on page 888.

Syntax Description

| <attempts> | Selects the number of call retries that will be made after a link failure (valid range is 0 to 15). Setting this value to 0 will allow unlimited retries during the time the network is failed. |

Default Values

By default, `backup maximum-retry` is set to 0 attempts.

Usage Examples

The following example configures the SROS to retry a dial-backup call four times before considering backup operation not available:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup maximum-retry 4
```
backup number <digits> [analog | digital-56k | digital 64k] <isdn min chan> <isdn max chan> ppp <interface>

Use the `backup number` command to configure the phone number and the call type the router will dial upon network failure. Multiple entries can be made for an interface to allow alternate sites to be dialed. For more detailed information on backup functionality, refer to the `Functional Notes` section of the command `backup call-mode <role>` on page 888.

**Syntax Description**

- `<digits>` Specifies the phone numbers to call when the backup is initiated.
- `analog` Indicates number connects to an analog modem.
- `digital-56k` Indicates number belongs to a digital 56 kbps per DS0 connection.
- `digital-64k` Indicates number belongs to a digital 64 kbps per DS0 connection.
- `<isdn min chan>` Specifies the minimum number of DS0s required for the backup link (Range: 1 to 24).
- `<isdn max chan>` Specifies the maximum number of DS0s desired for the backup link (Range: 1 to 24).
- `ppp <interface>` Specifies the PPP interface to use as the backup for this interface.

**Default Values**

By default, there are no configured `backup numbers`.

**Usage Examples**

The following example configures the SROS to dial 704-555-1212 (digital 64 kbps connection) to initiate backup operation for this endpoint using the configured `ppp 1` backup interface:

```plaintext
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup number 7045551212 digital-64k 1 1 ppp 1
```
backup priority <value>

Use the backup priority command to select the backup priority for the primary interface. This allows the user to establish the highest priority backup link and ensure that link will override backups attempted by lower priority links. Use the no form of this command to return to the default value. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 888.

Syntax Description

<value>  Sets the relative priority of this link (valid range: 0 to 100). A value of 100 designates the highest priority.

Default Values

By default, backup priority is set to 50.

Usage Examples

The following example assigns the highest priority to this endpoint:

ProCurve(config)#interface hdlc 16
ProCurve(config-hdlc 16)#backup priority 100
backup randomize-timers

Use the backup randomize-timers command to randomize the call timers to minimize potential contention for resources. Use the no form of this command to return to the default value. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 888.

Syntax Description

No subcommands.

Default Values

By default, the SROS does not randomize the backup call timers.

Usage Examples

The following example configures the SROS to randomize the backup timers associated with this endpoint:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup randomize-timers
backup redial-delay <seconds>

Use the backup redial-delay command to configure the delay after an unsuccessful call until the call will be re-tryed. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 888.

Syntax Description

<number> Specifies the delay in seconds between attempting to re-dial a failed backup attempt. Range: 10 to 3600.

Default Values

By default, backup redial-delay is set to 10 seconds.

Usage Examples

The following example configures a redial delay of 25 seconds on this endpoint:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup redial-delay 25
backup restore-delay <seconds>

Use the backup restore-delay command to configure the amount of time the router will wait after the network is restored before disconnecting the backup link and reverting to the primary. This setting is used to prevent disconnecting the backup link if the primary link is “bouncing” in and out of alarm. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 888.

Syntax Description

<seconds> Specifies the number of seconds the SROS will wait (after a primary link is restored) before disconnecting backup operation. Range: 10 to 86,400.

Default Values

By default, backup restore-delay is set to 10 seconds.

Usage Examples

The following example configures the SROS to wait 30 seconds before disconnecting backup operation and restoring the primary connection for this endpoint:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup restore-delay 30
backup schedule [day | enable-time | disable-time]

Use the `backup schedule` command to set the time of day that backup will be enabled. Use this command if backup is desired only during normal business hours and on specific days of the week. Use the `no` form of this command to disable backup (as specified). For more detailed information on backup functionality, refer to the Functional Notes section of the command `backup call-mode <role>` on page 888.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>Sets the days to allow backup (valid range: Monday through Sunday).</td>
</tr>
<tr>
<td>enable-time</td>
<td>Sets the time of day to enable backup. Time is entered in 24-hour format (00:00).</td>
</tr>
<tr>
<td>disable-time</td>
<td>Sets the time of day to disable backup.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, backup is enabled for all days and times if the `backup auto-backup` command has been issued and the backup schedule has not been entered.

**Usage Examples**

The following example enables backup Monday through Friday 8:00 am to 7:00 pm:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup schedule enable-time 08:00
ProCurve(config-hdlc 1)#backup schedule disable-time 19:00
ProCurve(config-hdlc 1)#no backup schedule day Saturday
ProCurve(config-hdlc 1)#no backup schedule day Sunday
```
**backup shutdown**

Use the `backup shutdown` command to deactivate all backup functionality in the unit. Backup configuration parameters are kept intact, but the unit will not initiate (or respond) to backup sequences in the event of a network outage. Use the `no` form of this command to reactivate the backup interface. For more detailed information on backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 888.

**Syntax Description**

No subcommands.

**Default Values**

By default, all SROS interfaces are disabled.

**Usage Examples**

The following example deactivates the configured backup interface:

```
ProCurve(config)#interface hdlc 16
ProCurve(config-hdlc 16)#backup shutdown
```
**bandwidth <value>**

Use the `bandwidth` command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the `no` form of this command to restore the default values.

**Syntax Description**

`<value>`
Enter bandwidth in kbps.

**Default Values**

To view default values use the `show interfaces` command.

**Functional Notes**

The `bandwidth` command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the HDLC interface to 10 Mbps:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#bandwidth 10000
```
**bridge-group** <group#>

Use the **bridge-group** command to assign an interface to the specified bridge group. Use the **no** form of this command to remove the interface from the bridge group.

**Syntax Description**

| <group#> | Specifies bridge group number (1 to 255) specified using the **bridge-group** command. |

**Default Values**

By default, there are no configured bridge groups.

**Functional Notes**

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (Ethernet to T1 bridge, Ethernet to Frame Relay sub-interface, etc.).

**Usage Examples**

The following example assigns the HDLC interface labeled 1 to bridge-group 1:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#bridge-group 1
```
crypto map <mapname>

Use the **crypto map** command to associate crypto maps with the interface.

**Note**

> When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

> For VPN configuration example scripts, refer to the **VPN Configuration Guide** located on the ProCurve SROS Documentation CD provided with your unit.

### Syntax Description

- **<mapname>**
  - Enter the crypto map name that you wish to assign to the interface.

### Default Values

By default, no crypto maps are assigned to an interface.

### Functional Notes

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the un-encrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the previous diagram, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access-group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only un-encrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, un-encrypted source of the data. The destination information will be the far-end, un-encrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy-class is the far-end. The destination information is the local-side.

**Usage Examples**

The following example applies all crypto maps with the name MyMap to the HDLC 1 interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#crypto map MyMap
```
dynamic-dns

Use the dynamic-dns command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the no versions of these commands to disable these features. Variations of this command include:

- `dynamic-dns custom <hostname> <minutes>`
- `dynamic-dns dyndns <hostname> <username> <password>`
- `dynamic-dns dyndns-custom <hostname> <username> <password>`
- `dynamic-dns dyndns-static <hostname> <username> <password>`

Syntax Description

- `<hostname>` Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).
- `<minutes>` Specifies the intervals in minutes to update the server with information (updates also occur when the interface's IP address changes regardless of the update intervals).
  
  Refer to Functional Notes below for additional argument descriptions.

Default Values

No default is necessary for this command.

Functional Notes

- **custom** - Constanttime.com’s Custom Dynamic DNS\textsuperscript{SM} service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows to manage IP address mappings (A records), domain aliases (CNAME records) and mail servers (MX records).

- **dyndns** - The Dynamic DNS\textsuperscript{SM} service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- **dyndns-custom** - DynDNS.org's Custom DNS\textsuperscript{SM} service provides a full DNS solution, giving you complete control over an entire domain name. A web-based interface provides two levels of control over your domain, catering to average or power users. Five globally redundant DNS servers ensure that your domain will always resolve.

  A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background, and provides layout and functionality similar to a BIND zone file allowing for the creation of nearly any record type.

  Custom DNS\textsuperscript{SM} can be used with both static and dynamic IPs, and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
**dyndns-static** - The Static DNS service is similar to Dynamic DNS service, in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate through the DNS system. This service is provided for up to five hostnames.

If your IP address doesn't change often or at all, but you still want an easy name to remember it by (without having to purchase your own domain name) Static DNS service is ideal for you.

If you would like to use your own domain name (such as yourname.com) you need Custom DNS service which also provides full dynamic and static IP address support.

**Usage Examples**

The following example sets the dynamic-dns to **dyndns-custom** with hostname **host**, username **user**, and password **pass**:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#dynamic-dns dyndns-custom host user pass
```
**fair-queue <threshold>**

Use the `fair-queue` command to enable weighted fair queuing (WFQ) on an interface. Use the `no` form of this command to disable WFQ and enable FIFO (first-in-first-out) queuing for an interface. WFQ is enabled by default for WAN interfaces.

**Syntax Description**

| `<threshold>`         | Optional. Value that specifies the maximum number of packets that can be present in each conversation sub-queue. Packets received for a conversation after this limit is reached are discarded. Valid range is 16 to 512. |

**Default Values**

By default, fair-queue is enabled with a threshold of 64 packets.

**Usage Examples**

The following example enables WFQ on the interface with a threshold set at 100 packets:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#fair-queue 100
```
hold-queue <queue size> out

Use the hold-queue command to change the overall size of an interface's WAN output queue. Use the no form of this command to return to the default setting.

Syntax Description

| <queue size> | The total number of packets the output queue can contain before packets are dropped. Valid range is 16-1000. |

Default Values

The default queue size for WFQ is 400. The default queue size for PPP FIFO and Frame Relay round-robin is 200.

Usage Examples

The following example sets the overall output queue size to 700:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#hold-queue 700
ip access-group <listname> [in | out]

Use the ip access-group command to create an access list to be used for packets transmitted on or received from the specified interface. Use the no form of this command to disable this type of control.

Syntax Description

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listname&gt;</td>
<td>Assigned IP access list name.</td>
</tr>
<tr>
<td>in</td>
<td>Enables access control on packets received on the specified interface.</td>
</tr>
<tr>
<td>out</td>
<td>Enables access control on packets transmitted on the specified interface.</td>
</tr>
</tbody>
</table>

Default Values

By default, these commands are disabled.

Functional Notes

When this command is enabled, the IP destination address of each packet must be validated before being passed through to the router system. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the unit to only allow Telnet traffic (as defined in the user-configured TelnetOnly IP access list) into the HDLC interface:

```
ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#int hdlc 1
ProCurve(config-hdlc 1)#ip access-group TelnetOnly in
```
ip address <address> <mask> [secondary]

Use the `ip address` command to define an IP address on the specified interface. Use the optional `secondary` keyword to define a secondary IP address. Use the `no` form of this command to remove a configured IP address.

**Syntax Description**

<address>  Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).

<mask>  Specifies the subnet mask that corresponds to the listed IP address.

secondary  Optional. Keyword used to configure a secondary IP address for the specified interface.

**Default Values**

By default, there are no assigned IP addresses.

**Functional Notes**

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

**Usage Examples**

The following example configures a secondary IP address of 192.168.72.101/30:

```
ProCurve(config)#hdlc 1
ProCurve(config-hdlc 1)#ip address 192.168.72.101 255.255.255.252 secondary
```

To view configured secondary IP address(es), use the `show running-config` command. Secondary IP address information is not available through other `show` commands.
ip helper-address <address>

Use the `ip helper-address` command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the `no` form of this command to disable forwarding packets.

**Note**  
The `ip helper` command must be used in conjunction with the `ip forward-protocol` command to configure the SROS to forward UDP broadcast packets. Refer to `ip forward-protocol udp <port number>` on page 387 for more information.

**Syntax Description**

| <address> | Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets. |

**Default Values**

By default, broadcast UDP packets are not forwarded.

**Functional Notes**

When used in conjunction with the `ip forward-protocol` command, the `ip helper-address` feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign a helper-address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the `ip forward-protocol` command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the `ip helper-address` feature:

1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the `ip forward-protocol` command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

**Usage Examples**

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.33.5.99:

```
ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip helper-address 192.168.5.99
```
ip igmp

Use the **ip igmp** command to configure multicasting-related functions for the interface.

### Syntax Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>immediate-leave</td>
<td>Specifies that if only one host (or IGMP snooping switch) is connected to the interface, when a leave is received, multicast of that group is immediately terminated as opposed to sending a group query and timing out the group if no device responds. Applies to all groups when configured.</td>
</tr>
<tr>
<td>last-member-query-interval</td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group’s multicast packets to the interface. Range: 100 to 65535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td>querier-timeout</td>
<td>Specifies the number of seconds that the router waits after the current querier’s last query before it takes over as querier (IGMP V2). Range: 60-300 seconds. Default: 2x the <strong>query-interval</strong> value.</td>
</tr>
<tr>
<td>query-interval</td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td>query-max-response-time</td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td>static-group</td>
<td>Configures the router’s interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>version [1</td>
<td>2]</td>
</tr>
</tbody>
</table>

### Usage Examples

The following example sets the query message interval on the interface to 200 milliseconds:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip igmp last-member-query-interval 200
```
**ip mcast-stub downstream**

Use the `ip mcast-stub downstream` command to enable multicast forwarding and IGMP (router mode) on an interface and place it in multicast stub downstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub upstream` commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub downstream` on page 914 for more information.

**Usage Examples**

The following example enables multicast forwarding and IGMP on the HDLC interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip mcast-stub downstream
```
ip mcast-stub fixed

Use the `ip mcast-stub fixed` command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the `no` form of this command to disable this mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Multicast routing must be enabled prior to setting `ip mcast-stub fixed` on the selected interface. Also, use the `ip igmp static-group <A.B.C.D>` command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

**Usage Examples**

The following example enables multicast traffic forwarding and IGMP on the interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip mcast-stub fixed
```
ip mcast-stub helper-enable

Use the `ip mcast-stub helper-enable` command to assign the `ip mcast-stub helper-address` as the IGMP proxy. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address`, `ip mcast-stub upstream`, and `ip mcast-stub downstream` commands. When enabled, the interface becomes a helper forwarding interface. The IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the unit to perform as an IGMP proxy. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub downstream` on page 914 for more information.

**Usage Examples**

The following example sets the helper address as the IGMP proxy:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip mcast-stub helper-enable
```
ip mcast-stub upstream

Use the `ip mcast-stub upstream` command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub downstream` commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router's unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub downstream` on page 914 for more information.

**Usage Examples**

The following example enables multicast forwarding on the HDLC interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip mcast-stub upstream
```
**ip ospf**

Use the `ip ospf` command to customize OSPF settings (if needed).

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>authentication-key</strong></td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td><strong>cost &lt;value&gt;</strong></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td><strong>dead-interval &lt;seconds&gt;</strong></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>hello-interval &lt;seconds&gt;</strong></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>message-digest-key &lt;keyid&gt; md5 &lt;key&gt;</strong></td>
<td>Configures OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td><strong>priority &lt;value&gt;</strong></td>
<td>Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td><strong>retransmit-interval &lt;seconds&gt;</strong></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>transmit-delay &lt;seconds&gt;</strong></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32,767.</td>
</tr>
</tbody>
</table>

### Default Values

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>retransmit-interval &lt;seconds&gt;</td>
<td>5 seconds</td>
</tr>
<tr>
<td>transmit-delay &lt;seconds&gt;</td>
<td>1 second</td>
</tr>
<tr>
<td>hello-interval &lt;seconds&gt;</td>
<td>10 seconds</td>
</tr>
<tr>
<td>dead-interval &lt;seconds&gt;</td>
<td>40 seconds</td>
</tr>
</tbody>
</table>

### Usage Example

The following example sets the maximum number of seconds allowed between hello packets to 25,000:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip ospf dead-interval 25000
```
**ip ospf authentication [message-digest | null]**

Use the `ip ospf authentication` command to authenticate an interface that is performing OSPF authentication.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-digest</td>
<td>Optional. Select message-digest authentication type.</td>
</tr>
<tr>
<td>null</td>
<td>Optional. Select for no authentication to be used.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this is set to null (meaning no authentication is used).

**Usage Examples**

The following example specifies that no authentication will be used on the HDLC interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip ospf authentication null
```
ip ospf network [broadcast | point-to-point]

Use the ip ospf network command to specify the type of network on this interface.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcast</td>
<td>Set the network type for broadcast.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Set the network type for point-to-point.</td>
</tr>
</tbody>
</table>

Default Values

By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

Functional Notes

A point-to-point network will not elect designated routers.

Usage Examples

The following example designates a broadcast network type:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip ospf network broadcast
**ip pim sparse-mode**

Use the `ip pim sparse-mode` command to enable PIM Sparse Mode on the interface. Use the `no` form of this command to disable PIM Sparse Mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, PIM Sparse Mode is disabled for all interfaces.

**Functional Notes**

PIM Sparse Mode is a multicast routing protocol that makes use of the unicast forwarding table. PIM-systems builds unidirectional shared trees rooted at a Rendezvous Point (RP) for a multicast group or a shortest-path tree rooted at a specific source for a multicast group.

**Usage Examples**

The following example enables PIM sparse-mode on the HDLC interface 1:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip pim sparse-mode
```
ip pim-sparse dr-priority <value>

Use the ip pim-sparse dr-priority command to specify the priority of this PIM interface for use when selecting the designated router (DR). Use the no form of this command to return to the default value.

Syntax Description

| <value> | Specifies the priority of this interface (to be used when determining the DR). Valid range is 1 to 4,294,967,295. |

Default Values

By default, the priority of all PIM interfaces is 1.

Functional Notes

Interfaces advertise their configured priority values in the hello messages transmitted on the interface. Routers use the priority values to determine the appropriate DR. The router on the network segment with the highest priority is selected as the DR. If a hello message is received on the interface from a router on the network segment and it does not contain a priority, the entire network segment defaults to DR selection based on IP addresses instead of priority. In this instance, the DR is selected as the router on the network segment that has the highest IP address. ProCurve Secure Routers will always include a priority in all transmitted hello messages. If no priority is specifically designated by the user, the priority is set as the default of 1.

Usage Examples

The following example specifies a priority of 100 on the HDLC interface 1:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip pim-sparse dr-priority 100
The following example specifies hellos to be sent on the HDLC interface every 3600 seconds:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip pim-sparse hello.timer 3600
ip pim-sparse nbr-timeout <time>

Use the `ip pim-sparse nbr-timeout` command to specify the interval the PIM interface waits before declaring that the neighbor is not present. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;time&gt;</code></td>
<td>Specifies the time interval (in seconds) the PIM interface waits before a neighbor is considered not present. Valid range is 30 to 10,800 seconds.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the `nbr-timeout` is set to 105 seconds.

**Usage Examples**

The following example specifies a wait interval of 360 seconds on the HDLC interface 1:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip pim-sparse nbr-timeout 360
```
**ip pim-sparse override-interval** `<time>`

Use the `ip pim-sparse override-interval` command to specify the delay interval after a join/prune in which another router on the LAN may override the join/prune. Use the `no` form of this command to return to the default value.

**Syntax Description**

| `<time>` | Specifies the delay interval (in milliseconds) after a join/prune in which another router on the LAN may override the join/prune. Valid range is 0 to 65,535 milliseconds. |

**Default Values**

By default, the `override-interval` is set to 2500 milliseconds.

**Usage Examples**

The following example sets the delay interval for join/prune overrides to 3000 milliseconds on the HDLC interface 1:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip pim-sparse override-interval 3000
```
ip pim-sparse propagation-delay <time>

Use the ip pim-sparse propagation-delay command to specify the expected propagation delay for join/prune messages. Set the propagation delay (in milliseconds) to estimate the amount of delay found in the local link. Use the no form of this command to return to the default value.

Syntax Description

<time> Specifies the expected propagation delay in the local link in milliseconds. Valid range is 0 to 32,767 milliseconds.

Default Values

By default, the propagation-delay is set to 500 milliseconds.

Usage Examples

The following example sets the expected propagation delay to 300 milliseconds on the HDLC interface 1:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip pim-sparse propagation-delay 300
ip policy route-map <mapname>

Use the ip policy route-map command to assign a policy route map to this interface. Use the no form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

Syntax Description

<mapname> Specifies the route map to associate with this interface.

Default Values

By default, policy-based routing is disabled for all interfaces.

Usage Examples

The following example associates the route map named MyMap with HDLC interface 1:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip policy route-map MyMap
ip proxy-arp <ip address> <subnet mask>

Use the ip proxy-arp command to enable proxy Address Resolution Protocol (ARP) on the interface. Use the no form of this command to disable this feature.

Syntax Description

<ip address>  Defines the proxy ARP IP address in dotted decimal notation (for example: 192.168.73.101).

<subnet mask>  Specifies the subnet mask that corresponds to the listed IP address.

Default Values

By default, proxy-arp is enabled.

Functional Notes

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all proxy ARP requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

Usage Examples

The following enables proxy ARP on the HDLC interface:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip proxy-arp
ip rip receive version [1 | 2]

Use the `ip rip receive version` command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Only accept received RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Only accept received RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1 (the default value for the version command).

**Functional Notes**

Use the `ip rip receive version` command to specify a RIP version that overrides the `version` (in the Router RIP) configuration. See `version [1 l 2]` on page 1264 for more information.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the HDLC interface to accept only RIP version 2 packets:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip rip receive version 2
```
ip rip send version [1 | 2]

Use the `ip rip send version` command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the `no` form of this command to restore the default value.

Syntax Description

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Only transmits RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Only transmits RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

Default Values

By default, all interfaces transmit RIP version 1 (the default value for the `version` command).

Functional Notes

Use the `ip rip send version` command to specify a RIP version that overrides the `version` (in the Router RIP) configuration. See `version [1 l 2]` on page 1264 for more information.

The SROS only transmits one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the HDLC interface to transmit only RIP version 2 packets:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip rip send version 2
```
ip rip summary-address <network address> <network mask>

Use the `ip rip summary-address` command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the `no` form of this command to disable this mode.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network address&gt;</td>
<td>Specifies the IP address of the network to be summarized.</td>
</tr>
<tr>
<td>&lt;network mask&gt;</td>
<td>Specifies the network mask to be applied to the specific network to be summarized.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no manual summarization is applied by RIP.

**Functional Notes**

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the `ip rip summary-address` command. This command is only effective if RIP version 2 is configured.

**Usage Examples**

The following example enables manual summarization on the specified IP address:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip rip summary-address 10.10.123.0 255.255.255.0
**ip route-cache**

Use the `ip route-cache` command to enable route caching on the interface. Use the `no` form of this command to disable route caching and return to process switching mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, route caching is enabled on all interfaces.

**Functional Notes**

Route caching allows an IP interface to provide optimum performance when processing IP traffic.

**Usage Examples**

The following example enables route caching on the HDLC interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip route-cache
```
ip unnumbered <interface>

Use the **ip unnumbered** command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the **no** form of this command to remove the unnumbered configuration.

**Syntax Description**

| **<interface>** | Specifies the interface in the format **type slot/port** (e.g., **ppp 1**) that contains the IP address to be used as the source address for all packets transmitted on this interface. |

**Default Values**

By default, all interfaces are configured to use a specified IP address (using the **ip address** command).

**Functional Notes**

If **ip unnumbered** is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying **ip unnumbered eth 0/1** while in the Frame Relay Sub-Interface Configuration mode configures the Frame Relay sub-interface to use the IP address assigned to the Ethernet interface for all IP processing. In addition, the SROS uses the specified interface information when sending route updates over the unnumbered interface.

**Usage Examples**

The following example configures the HDLC interface to use the IP address assigned to the Ethernet interface (**eth 0/1**):

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#ip unnumbered eth 0/1
**keepalive <seconds>**

Use the `keepalive` command to enable the transmission of keepalive packets on the interface and specify the time interval in seconds between transmitted packets.

**Syntax Description**

| <seconds> | Defines the time interval (in seconds) between transmitted keepalive packets (valid range: 0 to 32,767 seconds). |

**Default Values**

By default, the time interval between transmitted keepalive packets is 10 seconds.

**Functional Notes**

If three keepalive packets are sent to an interface with no response, the interface is considered down. To detect interface failures quickly, specify a smaller keepalive time.

**Usage Examples**

The following example specifies a keepalive time of 5 seconds on the HDLC interface:

```plaintext
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#keepalive 5
```
Ildp receive

Use the `Ildp receive` command to allow LLDP packets to be received on this interface.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces are configured to send and receive LLDP packets.

**Usage Examples**

The following example configures the HDLC interface to receive LLDP packets:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#Ildp receive
```
lldp send [management-address | port-description | system-capabilities | system-description | system-name]

Use the lldp send command to configure this interface to transmit LLDP packets or to control the types of information contained in the LLDP packets transmitted by this interface.

Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>management-address</td>
<td>Enables transmission of management address information on this interface.</td>
</tr>
<tr>
<td>port-description</td>
<td>Enables transmission of port description information on this interface.</td>
</tr>
<tr>
<td>system-capabilities</td>
<td>Enables transmission of this device's system capabilities on this interface.</td>
</tr>
<tr>
<td>system-description</td>
<td>Enables transmission of this device's system description on this interface.</td>
</tr>
<tr>
<td>system-name</td>
<td>Enables transmission of this device's system name on this interface.</td>
</tr>
</tbody>
</table>

Default Values

Be default, all interfaces are configured to transmit and receive LLDP packets of all types.

Functional Notes

Individual LLDP information can be enabled or disabled using the various forms of the lldp send command. For example, use the lldp send-and-receive command to enable transmit and receive of all LLDP information. Then use the no lldp send port-description command to prevent LLDP from transmitting port description information.

Usage Examples

The following example configures the HDLC interface to transmit LLDP packets containing all enabled information types:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#lldp send
lldp send-and-receive

Use the `lldp send-and-receive` command to configure this interface to transmit and receive LLDP packets.

Syntax Description

No subcommands.

Default Values

By default, all interfaces are configured to transmit and receive LLDP packets of all types.

Functional Notes

Individual LLDP information can be enabled or disabled using the various forms of the `lldp send` command. For example, use the `lldp send-and-receive` command to enable transmit and receive of all LLDP information. Then use the `no lldp send port-description` command to prevent LLDP from transmitting port description information.

Usage Examples

The following example configures the HDLC interface (hdlc 16) to transmit and receive LLDP packets containing all information types:

```
ProCurve(config)#interface hdlc 16
ProCurve(config-hdlc 16)#lldp send-and-receive
```
max-reserved-bandwidth <percent>

Use the `max-reserved-bandwidth` command to specify the percentage of interface bandwidth reserved for use in user-defined (priority or class-based) queues. The remainder of the interface bandwidth is reserved for system critical traffic and is not available to user-defined queues. Use the `no` form of this command to restore the default value.

**Caution** Reserving a portion of the interface bandwidth for system critical traffic is necessary for proper operation. Specifying the entire interface bandwidth for use in user-defined queues can cause undesirable operation.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;percent&gt;</th>
<th>Specifies the percentage of interface bandwidth to make available for user-defined (priority or class-based) queues. Enter an integer 1 to 100.</th>
</tr>
</thead>
</table>

**Default Values**

By default, `max-reserved-bandwidth` is set to **75** which reserves 25 percent of the interface bandwidth for system critical traffic.

**Usage Examples**

The following example specifies 85 percent of the bandwidth on the **hdlc 1** interface to be available for use in user-defined queues:

```
(config)#interface hdlc 1
(config-hdlc 1)#max-reserved-bandwidth 85
```
**mtu <size>**

Use the **mtu** command to configure the maximum transmit unit (MTU) size for the active interface. Use the **no** form of this command to return to the default value.

**Syntax Description**

<size>  
Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>64 to 256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>64 to 18,190</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>64 to 1500</td>
</tr>
</tbody>
</table>

**Default Values**

The default values for the various interfaces are listed below:

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>1500</td>
</tr>
</tbody>
</table>

**Functional Notes**

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

**Usage Examples**

The following example specifies an MTU of 1200 on the HDLC interface:

```bash
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#mtu 1200
```
**qos-policy out <mapname>**

Use the **qos-policy out** command to apply a previously-configured QoS map to an interface. Use the **no** form of this command to remove the map from the interface. The **out** keyword specifies that this policy will be applied to outgoing packets.

**Syntax Description**

| <mapname> | Specifies the name of a previously-created QoS map (see **qos map <mapname> <sequence number>** on page 451 for more information). |

**Default Values**

No default value is necessary for this command.

**Usage Examples**

The following example applies the QoS map **VOICEMAP** to the HDLC interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#qos-policy out VOICEMAP
```
**snmp trap link-status**

Use the **snmp trap link-status** command to control the SNMP variable `ifLinkUpDownTrapEnable` (RFC2863), which enables (or disables) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the `ifLinkUpDownTrapEnable` OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The **snmp trap link-status** command is used to control the RFC2863 `ifLinkUpDownTrapEnable` OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the HDLC interface:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#no snmp trap link-status
```
LOOPBACK INTERFACE CONFIGURATION COMMAND SET

To activate the Loopback Interface Configuration mode, enter the interface loopback command at the Global Configuration mode prompt. For example:

ProCurve>enable  
ProCurve#configure terminal  
ProCurve(config)#interface loopback 1  
ProCurve(config-loop 1)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- alias <"text"> on page 1303
- bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port> on page 1304
- description <text> on page 1307
- do on page 1308
- end on page 1309
- exit on page 1310
- ping <address> on page 1311
- shutdown on page 1315

All other commands for this command set are described in this section in alphabetical order.

- access-policy <policyname> on page 943
- bandwidth <value> on page 944
- crypto map <mapname> on page 945
- dynamic-dns on page 947
- ip commands begin on page 949
- mtu <size> on page 973
- snmp trap on page 974
- snmp trap link-status on page 975
access-policy <policyname>

Use the **access-policy** command to assign a specified access policy for the inbound traffic to an interface. Use the **no** form of this command to remove an access policy association.

### Syntax Description

| **<policyname>** | Identifies the configured access policy by alphanumeric descriptor (all access policy descriptors are case-sensitive). |

### Default Values

By default, there are no configured access policies associated with an interface.

### Functional Notes

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy <policy name>**. For more details on creating and using access policies, refer to **ip policy-class <policyname>** on page 400.

### Usage Examples

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the loopback interface:

Enable the SROS security features:
ProCurve(config)#ip firewall

Create the access list (this is the packet selector):
ProCurve(config)#ip access-list extended InWeb
ProCurve(config-ext-nacl)#permit tcp any host 10.12.5.253 eq 80

Create the access policy that contains the access list **InWeb**:
ProCurve(config)#ip policy-class UnTrusted
ProCurve(config-policy-class)#allow list InWeb

Associate the access policy with the loopback interface:
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#access-policy UnTrusted
**bandwidth <value>**

Use the *bandwidth* command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the *no* form of this command to restore the default values.

**Syntax Description**

<value> Specifies bandwidth in kbps.

**Default Values**

To view default values, use the *show interfaces* command.

**Functional Notes**

The *bandwidth* command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the loopback interface to 10 Mbps:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#bandwidth 10000
```
crypto map <mapname>

Use the crypto map command to associate crypto maps with the interface.

**Note**  
When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the **VPN Configuration Guide** located on the ProCurve SROS Documentation CD provided with your unit.

**Syntax Description**

<mapname>  
Specifies the crypto map name that you wish to assign to the interface.

**Default Values**

By default, no crypto maps are assigned to an interface.

**Functional Notes**

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following information in mind:

When defining the policy class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the unencrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the previous diagram, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only unencrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, unencrypted source of the data. The destination information will be the far end, unencrypted destination of the data. However, ACLs for a policy class work in reverse. The source information for the ACL in a policy class is the far end. The destination information is the local side.

Usage Examples

The following example applies all crypto maps with the name MyMap to the loopback interface:

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#crypto map MyMap
dynamic-dns

Use the dynamic-dns command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the no versions of these commands to disable these features. Variations of this command include:

- `dynamic-dns custom <hostname> <minutes>`
- `dynamic-dns dyndns <hostname> <username> <password>`
- `dynamic-dns dyndns-custom <hostname> <username> <password>`
- `dynamic-dns dyndns-static <hostname> <username> <password>`

**Syntax Description**

- `<hostname>` Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).
- `<minutes>` Specifies the intervals in minutes to update the server with information (updates also occur when the interface's IP address changes regardless of the update intervals).

Refer to **Functional Notes** below for additional argument descriptions.

**Default Values**

No default is necessary for this command.

**Functional Notes**

- **custom** - Constanttime.com’s Custom Dynamic DNSSM service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows to manage IP address mappings (A records), domain aliases (CNAME records) and mail servers (MX records).

- **dyndns** - The Dynamic DNSSM service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- **dyndns-custom** - DynDNS.org's Custom DNSSM service provides a full DNS solution, giving you complete control over an entire domain name. A web-based interface provides two levels of control over your domain, catering to average or power users. Five globally redundant DNS servers ensure that your domain will always resolve.

A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background, and provides layout and functionality similar to a BIND zone file allowing for the creation of nearly any record type.

Custom DNSSM can be used with both static and dynamic IPs, and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
**dyndns-static** - The Static DNS service is similar to Dynamic DNS service, in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate though the DNS system. This service is provided for up to five hostnames.

If your IP address doesn't change often or at all, but you still want an easy name to remember it by (without having to purchase your own domain name) Static DNS service is ideal for you.

If you would like to use your own domain name (such as yourname.com) you need Custom DNS service which also provides full dynamic and static IP address support.

**Usage Examples**

The following example sets the **dynamic-dns** to **dyndns-custom** with hostname **host**, username **user**, and password **pass**.

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#dynamic-dns dyndns-custom host user pass
ip access-group <listname> [in | out]

Use the ip access-group command to create an access list to be used for packets transmitted on or received from the specified interface. Use the no form of this command to disable this type of control.

Syntax Description

- **<listname>**  Specifies IP access list name.
- **in**  Enables access control on packets received on the specified interface.
- **out**  Enables access control on packets transmitted on the specified interface.

Default Values

By default, these commands are disabled.

Functional Notes

When this command is enabled, the IP destination address of each packet must be validated before being passed through to the router system. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the router to allow only Telnet traffic into the loopback interface:

```
ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#interface loopback 1
ProCurve(config-loop 1)#ip access-group TelnetOnly in
```
ip address <address> <mask> [secondary]

Use the **ip address** command to define an IP address on the specified interface. Use the optional keyword **secondary** to define a secondary IP address. Use the **no** form of this command to remove a configured IP address.

**Syntax Description**

- `<address>`: Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).
- `<mask>`: Specifies the subnet mask that corresponds to the listed IP address.
- **secondary**: Optional. Configures a secondary IP address for the specified interface.

**Default Values**

By default, there are no assigned IP addresses.

**Functional Notes**

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

**Usage Examples**

The following example configures a secondary IP address of **192.22.72.101/30**:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip address 192.168.72.101 255.255.255.252 secondary
```

To view configured secondary IP address(es), use the **show running-config** command. Secondary IP address information is not available through other **show** commands.
ip helper-address <address>

Use the ip helper-address command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the no form of this command to disable forwarding packets.

Note  The ip helper command must be used in conjunction with the ip forward-protocol command to configure the SROS to forward UDP broadcast packets.

Syntax Description

<address> Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

Default Values

By default, broadcast UDP packets are not forwarded.

Functional Notes

When used in conjunction with the ip forward-protocol command, the ip helper-address feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign helper address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the ip forward-protocol command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the ip helper-address feature:
1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the ip forward-protocol command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

Usage Examples

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.168.5.99:

ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip helper-address 192.168.5.99
ip igmp

Use the `ip igmp` command to configure multicasting-related functions for the interface.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>immediate-leave</code></td>
<td>Specifies that if only one host (or IGMP snooping switch) is connected to the interface, when a leave is received, multicast of that group is immediately terminated as opposed to sending a group query and timing out the group if no device responds. Works in conjunction with <code>ip igmp last-member-query-interval</code>. Applies to all groups when configured.</td>
</tr>
<tr>
<td><code>last-member-query-interval &lt;milliseconds&gt;</code></td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group’s multicast packets to the interface. Range: 100 to 65535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td><code>querier-timeout &lt;seconds&gt;</code></td>
<td>Specifies the number of seconds that the router waits after the current querier’s last query before it takes over as querier (IGMP V2). Range: 60-300 seconds. Default: 2x the <code>query-interval</code> value.</td>
</tr>
<tr>
<td><code>query-interval &lt;seconds&gt;</code></td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td><code>query-max-response-time &lt;seconds&gt;</code></td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td><code>static-group &lt;group-address&gt;</code></td>
<td>Configures the router’s interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>`version [1</td>
<td>2]`</td>
</tr>
</tbody>
</table>

### Usage Examples

The following example sets the query message interval on the interface to 200 milliseconds:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip igmp last-member-query-interval 200
```
ip mcast-stub downstream

Use the `ip mcast-stub downstream` command to enable multicast forwarding and IGMP (router mode) on an interface and place it in multicast stub downstream mode. Use the `no` form of this command to disable.

Syntax Description

No subcommands.

Default Values

By default, this command is disabled.

Functional Notes

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub upstream` commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub upstream` on page 956 for more information.

Usage Examples

The following example enables multicast forwarding and IGMP on the interface:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip mcast-stub downstream
```
ip mcast-stub fixed

Use the **ip mcast-stub fixed** command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the **no** form of this command to disable this mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Multicast routing must be enabled prior to setting **ip mcast-stub fixed** on the selected interface. Also, use the **ip igmp static-group <A.B.C.D>** command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

**Usage Examples**

The following example enables multicast traffic forwarding and IGMP on the interface:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip mcast-stub fixed
```
ip mcast-stub helper-enable

Use the `ip mcast-stub helper-enable` command to assign the `ip mcast-stub helper-address` as the IGMP proxy. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address`, `ip mcast-stub upstream`, and `ip mcast-stub downstream` commands. When enabled, the interface becomes a helper forwarding interface. The IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the unit to perform as an IGMP proxy. Refer to `ip mcast-stub helper-enable` on page 955, `ip mcast-stub downstream` on page 953, and `ip mcast-stub upstream` on page 956 for more information.

**Usage Examples**

The following example sets the helper address as the IGMP proxy:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip mcast-stub helper-enable
```
ip mcast-stub upstream

Use the ip mcast-stub upstream command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the no form of this command to disable.

Syntax Description
No subcommands.

Default Values
By default, this command is disabled.

Functional Notes
This command is used in IP multicast stub applications in conjunction with the ip mcast-stub helper-address and ip mcast-stub downstream commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router’s unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface. Refer to ip mcast-stub helper-address <ip address> on page 397 and ip mcast-stub downstream on page 953 for more information.

Usage Examples
The following example enables multicast forwarding on the interface:

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip mcast-stub upstream
ip ospf

Use the `ip ospf` command to customize OSPF settings (if needed).

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication-key</td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td>cost <code>&lt;value&gt;</code></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32,767.</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td>message-digest-key <code>&lt;keyid&gt;</code> md5 <code>&lt;key&gt;</code></td>
<td>Configure OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td>priority <code>&lt;value&gt;</code></td>
<td>Sets the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32767.</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32767.</td>
</tr>
</tbody>
</table>

Default Values

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>5 seconds</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>1 second</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>10 seconds</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>40 seconds</td>
</tr>
</tbody>
</table>

Usage Examples

The following example sets the maximum number of seconds allowed between hello packets to 25K:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip ospf dead-interval 25000
```
ip ospf authentication [message-digest | null]

Use the ip ospf authentication command to authenticate an interface that is performing OSPF authentication.

Syntax Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-digest</td>
<td>Optional. Specifies message-digest authentication type.</td>
</tr>
<tr>
<td>null</td>
<td>Optional. Specifies that no authentication be used.</td>
</tr>
</tbody>
</table>

Default Values

By default, this is set to null (meaning no authentication is used).

Usage Examples

The following example specifies that no authentication will be used on the loopback interface:

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip ospf authentication null
The ip ospf network command is used to specify the type of network on this interface.

**Syntax Description**
- **broadcast**: Sets the network type for broadcast.
- **point-to-point**: Sets the network type for point-to-point.

**Default Values**
By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

**Functional Notes**
A point-to-point network will not elect designated routers.

**Usage Examples**
The following example designates a broadcast network type:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip ospf network broadcast
```
ip pim sparse-mode

Use the `ip pim sparse-mode` command to enable PIM Sparse Mode on the interface. Use the `no` form of this command to disable PIM Sparse Mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, PIM Sparse Mode is disabled for all interfaces.

**Functional Notes**

PIM Sparse Mode is a multicast routing protocol that makes use of the unicast forwarding table. PIM-systems builds unidirectional shared trees rooted at a Rendezvous Point (RP) for a multicast group or a shortest-path tree rooted at a specific source for a multicast group.

**Usage Examples**

The following example enables PIM sparse-mode on the loopback interface 1:

```
ProCurve(config)#interface loop 1
ProCurve(config-loop 1)#ip pim sparse-mode
```
ip pim-sparse dr-priority <value>

Use the `ip pim-sparse dr-priority` command to specify the priority of this PIM interface for use when selecting the designated router (DR). Use the `no` form of this command to return to the default value.

**Syntax Description**

| <value> | Specifies the priority of this interface (to be used when determining the DR). Valid range is 1 to 4,294,967,295. |

**Default Values**

By default, the priority of all PIM interfaces is 1.

**Functional Notes**

Interfaces advertise their configured priority values in the hello messages transmitted on the interface. Routers use the priority values to determine the appropriate DR. The router on the network segment with the highest priority is selected as the DR. If a hello message is received on the interface from a router on the network segment and it does not contain a priority, the entire network segment defaults to DR selection based on IP addresses instead of priority. In this instance, the DR is selected as the router on the network segment that has the highest IP address. ProCurve Secure Routers will always include a priority in all transmitted hello messages. If no priority is specifically designated by the user, the priority is set as the default of 1.

**Usage Examples**

The following example specifies a priority of 100 on the loopback interface 1:

```
ProCurve(config)#interface loop 1
ProCurve(config-loop 1)#ip pim-sparse dr-priority 100
```
ip pim-sparse hello-timer <time>

Use the ip pim-sparse hello-timer command to set the time interval at which periodic hello messages are transmitted out the interface. Each PIM interface has an independent hello-timer. Use the no form of this command to return to the default value.

Syntax Description

| <time> | Specifies the interval (in seconds) at which periodic hellos are sent out the interface. Valid range is 10 to 3600 seconds. |

Default Values

By default, hellos are transmitted on PIM interfaces every 60 seconds.

Functional Notes

Hello messages are used to inform neighbors of a router's presence. Hello messages normally generate a small amount of traffic on an interface. Setting the hello-timer to a small interval increases the amount of hellos sent (thus increasing the amount of traffic). Set the hello-timer to a reasonable value, taking into consideration the bandwidth available on the interface.

Usage Examples

The following example specifies hellos to be sent on the loopback interface every 3600 seconds:

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip pim-sparse hello-timer 3600
ip pim-sparse nbr-timeout <time>

Use the ip pim-sparse nbr-timeout command to specify the interval the PIM interface waits before declaring that the neighbor is not present. Use the no form of this command to return to the default value.

Syntax Description

<time> Specifies the time interval (in seconds) the PIM interface waits before a neighbor is considered not present. Valid range is 30 to 10,800 seconds.

Default Values

By default, the nbr-timeout is set to 105 seconds.

Usage Examples

The following example specifies a wait interval of 360 seconds on the loopback interface 1:

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip pim-sparse nbr-timeout 360
ip pim-sparse override-interval <time>

Use the `ip pim-sparse override-interval` command to specify the delay interval after a join/prune in which another router on the LAN may override the join/prune. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the delay interval (in milliseconds) after a join/prune in which another router on the LAN may override the join/prune. Valid range is 0 to 65,535 milliseconds. |

**Default Values**

By default, the `override-interval` is set to 2500 milliseconds.

**Usage Examples**

The following example sets the delay interval for join/prune overrides to 3000 milliseconds on the loopback interface 1:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip pim-sparse override-interval 3000
```
ip pim-sparse propagation-delay <time>

Use the `ip pim-sparse propagation-delay` command to specify the expected propagation delay for join/prune messages. Set the propagation delay (in milliseconds) to estimate the amount of delay found in the local link. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the expected propagation delay in the local link in milliseconds. Valid range is 0 to 32,767 milliseconds. |

**Default Values**

By default, the `propagation-delay` is set to 500 milliseconds.

**Usage Examples**

The following example sets the expected propagation delay to 300 milliseconds on the loopback interface 1:

```
ProCurve(config)#interface hdlc 1
ProCurve(config-loop 1)#ip pim-sparse propagation-delay 300
```
ip policy route-map <mapname>

Use the `ip policy route-map` command to assign a policy route map to this interface. Use the `no` form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

**Syntax Description**

- `<mapname>` Specifies the route map to associate with this interface.

**Default Values**

By default, policy-based routing is disabled for all interfaces.

**Usage Examples**

The following example associates the route map named `MyMap` with loopback interface 1:

```plaintext
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip policy route-map MyMap
```
ip proxy-arp <address> <subnet mask>

Use the `ip proxy-arp` command to enable proxy Address Resolution Protocol (ARP) on the interface. Use the `no` form of this command to disable this feature.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).</td>
</tr>
<tr>
<td>&lt;subnet mask&gt;</td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, proxy ARP is enabled.

**Functional Notes**

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all proxy ARP requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

**Usage Examples**

The following enables proxy ARP on the loopback interface:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip proxy-arp
```
ip rip receive version [1 | 2]

Use the **ip rip receive version** command to configure the RIP version the unit accepts in all RIP packets received on the interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accepts only received RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Accepts only received RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1 (the default value for the **version** command).

**Functional Notes**

Use the **ip rip receive version** to specify a RIP version that overrides the **version** (in the Router RIP) configuration.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the loopback interface to accept only RIP version 2 packets:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip rip receive version 2
```
**ip rip send version [1 | 2]**

Use the `ip rip send version` command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transmits only RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Transmits only RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces transmit RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip send version` to specify a RIP version that overrides the `version` (in the Router RIP) configuration.

The SROS only transmits one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the loopback interface to transmit only RIP version 2 packets:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip rip send version 2
```
ip rip summary-address <network address> <network mask>

Use the `ip rip summary-address` command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the `no` form of this command to disable this mode.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;network address&gt;</code></th>
<th>Specifies the IP address of the network to be summarized.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network mask&gt;</code></td>
<td>Specifies the network mask to be applied to the specific network to be summarized.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no manual summarization is applied by RIP.

**Functional Notes**

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the `ip rip summary-address` command. This command is only effective if RIP version 2 is configured.

**Usage Examples**

The following example enables manual summarization on the specified IP address:

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip rip summary-address 10.10.123.0 255.255.255.0
ip route-cache

Use the `ip route-cache` command to enable route caching on the interface. Use the `no` form of this command to disable route caching and return to process switching mode.

**Note**
Using Network Address Translation (NAT) or the SROS firewall capabilities on an interface requires process switching mode (using the `no ip route-cache` command).

**Syntax Description**
No subcommands.

**Default Values**
By default, route caching is enabled on all interfaces.

**Functional Notes**
Route caching allows an IP interface to provide optimum performance when processing IP traffic.

**Usage Examples**
The following example enables route caching on the loopback interface:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip route-cache
```
ip unnumbered <interface>

Use the ip unnumbered command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the no form of this command to remove the unnumbered configuration.

Syntax Description

| <interface> | Specifies the interface in the format type slot/port (e.g., ppp 1) that contains the IP address to be used as the source address for all packets transmitted on this interface. |

Default Values

By default, all interfaces are configured to use a specified IP address (using the ip address command).

Functional Notes

If ip unnumbered is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying ip unnumbered ppp 1 while in the Loopback Interface Configuration mode configures the Loopback interface to use the IP address assigned to the PPP interface for all IP processing. In addition, the SROS uses the specified interface information when sending route updates over the unnumbered interface.

Usage Examples

The following example configures the loopback interface (labeled loop 1) to use the IP address assigned to the PPP interface (PPP 1):

ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#ip unnumbered ppp 1
**mtu <size>**

Use the **mtu** command to configure the maximum transmit unit size for the active interface. Use the **no** form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ATM interfaces 64 to 1520&lt;br&gt;Demand interfaces 64 to 1520&lt;br&gt;Ethernet interfaces 64 to 1500&lt;br&gt;FDL interfaces 64 to 256&lt;br&gt;HDLC interfaces 64 to 1520&lt;br&gt;Loopback interfaces 64 to 1500&lt;br&gt;Tunnel interfaces 64 to 18,190&lt;br&gt;Virtual Frame Relay sub-interfaces 64 to 1520&lt;br&gt;Virtual PPP interfaces 64 to 1500</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>The default values for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ATM interfaces 1500&lt;br&gt;Demand interfaces 1500&lt;br&gt;Ethernet interfaces 1500&lt;br&gt;FDL interfaces 256&lt;br&gt;HDLC interfaces 1500&lt;br&gt;Loopback interfaces 1500&lt;br&gt;Tunnel interfaces 1500&lt;br&gt;Virtual Frame Relay sub-interfaces 1500&lt;br&gt;Virtual PPP interfaces 1500</td>
</tr>
</tbody>
</table>

**Functional Notes**

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

**Usage Examples**

The following example specifies an MTU of 1200 on the loopback interface:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#mtu 1200
```
snmp trap

Use the **snmp trap** command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces (except virtual Frame Relay interfaces and sub-interfaces) have SNMP traps enabled.

**Usage Examples**

The following example enables SNMP capability on the Ethernet interface:

```
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#snmp trap
```
snmp trap link-status

Use the **snmp trap link-status** command to control the Simple Network Management Protocol (SNMP) variable *ifLinkUpDownTrapEnable* (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the *ifLinkUpDownTrapEnable* OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The **snmp trap link-status** command is used to control the RFC2863 *ifLinkUpDownTrapEnable* OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the loopback interface:

```plaintext
ProCurve(config)#interface loopback 1
ProCurve(config-loop 1)#no snmp trap link-status
```
PPP INTERFACE CONFIGURATION COMMAND SET

To activate the PPP Interface Configuration mode, enter the `interface ppp` command at the Global Configuration mode prompt. For example:

ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# interface ppp 1
ProCurve(config-ppp 1)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `access-policy <policyname>` on page 978
- `alias link"<"text"">` on page 979
- `backup commands` begin on page 980
- `bandwidth <value>` on page 997
- `bridge-group <group#>` on page 998
- `crypto map <mapname>` on page 1000
- `dynamic-dns` on page 1002
- `fair-queue <threshold>` on page 1004
- `hold-queue <queue size> out` on page 1005
- `ip commands` begin on page 1006
- `keepalive <seconds>` on page 1033
- `lldp receive` on page 1034
- `lldp send [management-address l port-description l system-capabilities l system-description l system-name]` on page 1035
- `lldp send-and-receive` on page 1036
max-reserved-bandwidth <percent> on page 1037
mtu <size> on page 1038
peer default ip address <address> on page 1039
ppp commands begin on page 1040
pppoe ac-name <name> on page 1048
pppoe service-name <name> on page 1049
qos-policy out <mapname> on page 1050
snmp trap link-status on page 1051
username <username> password <password> on page 1052
**access-policy <policyname>**

Use the `access-policy` command to assign a specified access policy for the inbound traffic to an interface. Use the `no` form of this command to remove an access policy association.

**Note** Configured access policies will only be active if the `ip firewall` command has been entered at the Global Configuration mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.

**Syntax Description**

- `<policyname>` Identifies the configured access policy by alphanumeric descriptor. (All access policies are case-sensitive.)

**Note** All access policy descriptors are case-sensitive.

**Default Values**

By default, there are no configured access policies associated with an interface.

**Functional Notes**

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter `access policy <policy name>`. For more details on creating and using access policies, refer to `ip policy-class <policyname>` on page 400.

**Usage Examples**

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the virtual PPP interface:

1. Enable the SROS security features:
   ```plaintext```
   ProCurve(config)#ip firewall
   ```plaintext```

2. Create the access list (this is the packet selector):
   ```plaintext```
   ProCurve(config)#ip access-list extended InWeb
   ProCurve(config-ext-nacl)#permit tcp any host 10.12.5.253 eq 80
   ```plaintext```

3. Create the access policy that contains the access list **InWeb**:
   ```plaintext```
   ProCurve(config)#ip policy-class UnTrusted
   ProCurve(config-policy-class)#allow list InWeb
   ```plaintext```

4. Associate the access list with the PPP virtual interface (labeled 1):
   ```plaintext```
   ProCurve(config)#interface ppp 1
   ProCurve(config-ppp 1)#access-policy UnTrusted
   ```plaintext```
**alias link<“text”>**

Each configured PPP interface (when referenced using SNMP) contains a link (physical port) and a bundle (group of links). RFC 1471 (for Link Connection Protocol) provides an interface table to manage lists of bundles and associated links. The **alias link** command provides the management station an identifying description for each link (PPP physical).

**Syntax Description**

| `<“text”>` | Describes the interface (for SNMP) by alphanumeric character string (must be encased in quotation marks). |

**Default Values**

By default, the PPP identification string appears as empty quotes (“ ”).

**Functional Notes**

The **alias link** string should be used to uniquely identify a PPP link. Enter a string that clearly identifies the link.

**Usage Examples**

The following example defines a unique character string for the virtual PPP interface (1):

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#alias link “PPP_link_1”
```

**Technology Review**

Please refer to RFC 1990 for a more detailed discussion on PPP links and bundles.
backup auto-backup

Use the backup auto-backup command to configure the PPP interface to automatically attempt a backup upon failure. For more detailed information on PPP backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description
No subcommands.

Default Values
By default, all backup endpoints will automatically attempt backup upon a failure.

Usage Examples
The following example enables automatic backup on the endpoint:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup auto-backup
backup auto-restore

Use the backup auto-restore command to configure the interface to automatically discontinue backup when all network conditions are operational. Use the no form of this command to disable the auto-restore feature. For more detailed information on PPP backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically restore the primary connection when the failure condition clears.

Usage Examples

The following example configures the SROS to automatically restore the primary connection when the failure condition clears:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup auto-restore
backup backup-delay <seconds>

Use the backup backup-delay command to configure the amount of time the router will wait after the failure condition is recognized before attempting to backup the link. Use the no form of this command to return to the default value. For more detailed information on PPP backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

**Syntax Description**

| <seconds> | Specifies the delay period (in seconds) a failure must be active before the SROS will enter backup operation on the interface (valid range: 10 to 86400 seconds). |

**Default Values**

By default, the backup-delay period is set to 10 seconds.

**Usage Examples**

The following example configures the SROS to wait 60 seconds (on an endpoint with an active alarm condition) before attempting backup operation:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup backup-delay 60
backup call-mode <role>

Use the backup call-mode command to specify whether the configured backup interfaces answers or originates (or a combination of both) backup calls. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;role&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer</td>
<td>Answers and backs up primary link on failure.</td>
</tr>
<tr>
<td>answer-always</td>
<td>Answers and backs up regardless of primary link state.</td>
</tr>
<tr>
<td>originate</td>
<td>Originates backup call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer</td>
<td>Originates or answers call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer-always</td>
<td>Originates on failure; answers and backs up always.</td>
</tr>
</tbody>
</table>

Default Values

By default, backup call-mode is set to originate-answer.

Functional Notes

The majority of the configuration for the SROS backup implementation is configured via the backup PPP interface configuration commands. However, the numbers dialed are configured in the primary interface. Full sample configurations follow:

Sample config for remote router (dialing out)

hostname "Remote7203dl"
enable password password

interface eth 0/1
  ip address 192.168.1.254 255.255.255.0
  no shutdown

interface modem 1/3
  no shutdown

interface t1 1/1
  coding b8zs
  framing esf
  clock source line
  tdm-group 1 timeslots 1-24
  no shutdown

interface fr 1 point-to-point
   frame-relay lmi-type ansi
   no shutdown
   bind 1 t1 1/1 1 fr 1

interface fr 1.16 point-to-point
   frame-relay interface-dlci 16
   ip address 10.1.1.2 255.255.255.252
   backup call-mode originate
   backup number 5551111 analog ppp 1
   backup number 5552222 analog ppp 1
   no shutdown

interface ppp 1
   ip address  172.22.56.1  255.255.255.252
   ppp authentication chap
   username remoterouter password remotepass
   ppp chap hostname localrouter
   ppp chap password procurve
   no shutdown

ip route 192.168.2.0 255.255.255.0 172.22.56.2 255.255.255.252

line telnet 0 4
   password password

Sample config for central router (dialing in)

hostname "Central7203dl"
enable password password

interface eth 0/1
   ip address  192.168.100.254  255.255.255.0
   no shutdown

interface modem 1/3
   no shutdown

interface t1 1/1
   coding b8zs
   framing esf
   clock source line
tdm-group 1 timeslots 1-24
no shutdown
!
interface fr 1 point-to-point
frame-relay lmi-type ansi
no shutdown
bind 1 t1 1/1 1 fr 1
!
interface fr 1.100 point-to-point
frame-relay interface-dlci 100
ip address 10.1.1.1 255.255.255.252
backup call-mode answer
backup number 555-8888 analog ppp 1
!
interface ppp 1
ip address 172.22.56.2 255.255.255.252
ppp authentication chap
username localrouter password procurve
ppp chap hostname remoterrouter
ppp chap password remotepass
no shutdown
!
ip route 192.168.1.0 255.255.255.0 172.22.56.1 255.255.255.252

line telnet 0 4
password password

Usage Examples
The following configures the SROS to generate backup calls for this endpoint using an analog modem interface (to phone number 555 1111) but never answer calls and specifies ppp 2 as the backup interface:

ProCurve(config)#interface hdlc 1
ProCurve(config-hdlc 1)#backup call-mode originate
ProCurve(config-hdlc 1)#backup number 555 1111 analog ppp 2

Technology Review
This technology review provides information regarding specific backup router behavior (i.e., when the router will perform backup, where in the configuration the SROS accesses specific routing information, etc.):
Dialing Out
1. The SROS determines to place an outbound call when either the Layer 1 or Layer 2 has a failure.
2. When placing outbound calls, the SROS matches the number dialed to a PPP interface.
3. When placing the call, the SROS uses the configuration of the related PPP interface for authentication and IP negotiation.
4. If the call fails to connect on the first number dialed, the SROS places a call to the second number if configured. The second number to be dialed references a separate PPP interface.

Dialing In
1. The SROS receives an inbound call on a physical interface.
2. Caller ID is used to match the backup number command to the configured PPP interface.
3. If a match is found, the call connects and the SROS pulls down the primary connection if it is not already in a down state.
4. If no match is found from Caller ID, the call is terminated.
backup connect-timeout <seconds>

Use the backup connect-timeout command to specify the number of seconds to wait for a connection after a call is attempted before trying to call again or dialing a different number. It is recommended this number be greater than 60. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description

| <seconds> | Selects the amount of time (in seconds) that the router will wait for a connection before attempting another call (valid range: 10 to 300). |

Default Values

By default, the backup connect-timeout period is set to 60 seconds.

Usage Examples

The following example configures the SROS to wait 120 seconds before retrying a failed backup call:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup connect-timeout 120
backup force [backup | primary]

Use the backup force command to manually override the automatic backup feature. This can be used to force a link into backup to allow maintenance to be performed on the primary link without disrupting data. Use the no form of this command to return to the normal backup operation state. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description

<table>
<thead>
<tr>
<th>backup</th>
<th>Forces backup regardless of primary link state.</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>Forces primary link regardless of its state.</td>
</tr>
</tbody>
</table>

Default Values

By default, this feature is disabled.

Usage Examples

The following example configures the SROS to force this interface into backup:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup force backup
backup maximum-retry <attempts>

Use the `backup maximum-retry` command to select the number of calls the router will make when attempting to backup a link. Use the `no` form of this command to return to the default state. For more detailed information on backup functionality, refer to the `Functional Notes` section of the command `backup call-mode <role>` on page 983.

**Syntax Description**

| `<attempts>` | Selects the number of call retries that will be made after a link failure (valid range is 0 to 15). Setting this value to 0 will allow unlimited retries during the time the network is failed. |

**Default Values**

By default, `backup maximum-retry` is set to 0 attempts.

**Usage Examples**

The following example configures the SROS to retry a backup call four times before considering backup operation not available:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup maximum-retry 4
```
backup number <digits> [analog | digital-56k | digital 64k] <isdn min chan> <isdn max chan> ppp <interface>

Use the backup number command to configure the phone number and the call type the router will dial upon network failure. Multiple entries can be made for an interface to allow alternate sites to be dialed. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description

- `<digits>` Specifies the phone numbers to call when the backup is initiated.
- `analog` Indicates number connects to an analog modem.
- `digital-56k` Indicates number belongs to a digital 56 kbps per DS0 connection.
- `digital-64k` Indicates number belongs to a digital 64 kbps per DS0 connection.
- `<isdn min chan>` Specifies the minimum number of DS0s required for a digital 56 or 64 kbps connection (Range: 1 to 24).
- `<isdn max chan>` Specifies the maximum number of DS0s desired for a digital 56 or 64 kbps connection (Range: 1 to 24).
- `ppp <interface>` Specifies the PPP interface to use as the backup for this interface.

Default Values

By default, there are no configured backup numbers.

Usage Examples

The following example configures the SROS to dial 704-555-1212 (digital 64 kbps connection) to initiate backup operation on this endpoint using the configured PPP interface ppp 1:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup number 7045551212 digital-64k 1 1 ppp 1
**backup priority <value>**

Use the `backup priority` command to select the backup priority for this interface. This allows the user to establish the highest priority backup link and ensure that link will override backups attempted by lower priority links. Use the `no` form of this command to return to the default value. For more detailed information on backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 983.

**Syntax Description**

| <value> | Sets the relative priority of this link (valid range: 0 to 100). A value of 100 designates the highest priority. |

**Default Values**

By default, `backup priority` is set to 50.

**Usage Examples**

The following example assigns the highest priority to this endpoint:

```
ProCurve(config)#interface ppp 16
ProCurve(config-ppp 16)#backup priority 100
```
backup randomize-timers

Use the **backup randomize-timers** command to randomize the call timers to minimize potential contention for resources. Use the **no** form of this command to return to the default value. For more detailed information on backup functionality, refer to the *Functional Notes* section of the command *backup call-mode <role>* on page 983.

**Syntax Description**

No subcommands.

**Default Values**

By default, the SROS does not randomize the backup call timers.

**Usage Examples**

The following example configures the SROS to randomize the backup timers associated with this endpoint:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup randomize-timers
backup redial-delay <seconds>

Use the backup redial-delay command to configure the delay after an unsuccessful call until the call will be re-tried. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description

<seconds>  Specifies the delay in seconds between attempting to re-dial a failed backup attempt. Range: 10 to 3600.

Default Values

By default, backup redial-delay is set to 10 seconds.

Usage Examples

The following example configures a redial delay of 25 seconds on this endpoint:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup redial-delay 25
backup restore-delay <seconds>

Use the backup restore-delay command to configure the amount of time the router will wait after the network is restored before disconnecting the backup link and reverting to the primary. This setting is used to prevent disconnecting the backup link if the primary link is “bouncing” in and out of alarm. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description

| <seconds> | Specifies the number of seconds the SROS will wait (after a primary link is restored) before disconnecting backup operation. Range: 10 to 86,400. |

Default Values

By default, backup restore-delay is set to 10 seconds.

Usage Examples

The following example configures the SROS to wait 30 seconds before disconnecting backup operation and restoring the primary connection for this endpoint:

ProCurve(config)#interface ppp 1  
ProCurve(config-ppp 1)#backup restore-delay 30
backup schedule [day | enable-time | disable-time]

Use the **backup schedule** command to set the time of day that backup will be enabled. Use this command if backup is desired only during normal business hours and on specific days of the week. Use the **no** form of this command to disable backup (as specified). For more detailed information on backup functionality, refer to the *Functional Notes* section of the command *backup call-mode <role>* on page 983.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>day</strong></td>
<td>Sets the days to allow backup (valid range: Monday through Sunday).</td>
</tr>
<tr>
<td><strong>enable-time</strong></td>
<td>Sets the time of day to enable backup. Time is entered in 24-hour format (00:00).</td>
</tr>
<tr>
<td><strong>disable-time</strong></td>
<td>Sets the time of day to disable backup.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, backup is enabled for all days and times if the **backup auto-backup** command has been issued and the backup schedule has not been entered.

**Usage Examples**

The following example enables backup Monday through Friday 8:00 am to 7:00 pm:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#backup schedule enable-time 08:00
ProCurve(config-ppp 1)#backup schedule disable-time 19:00
ProCurve(config-ppp 1)#no backup schedule day Saturday
ProCurve(config-ppp 1)#no backup schedule day Sunday
```
backup shutdown

Use the backup shutdown command to deactivate all backup functionality in the unit. Backup configuration parameters are kept intact, but the unit will not initiate (or respond) to backup sequences in the event of a network outage. Use the no form of this command to reactivate the backup interface. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 983.

Syntax Description

No subcommands.

Default Values

By default, all SROS interfaces are disabled.

Usage Examples

The following example deactivates the configured backup interface:

ProCurve(config)#interface ppp 16
ProCurve(config-ppp 16)#backup shutdown
**bandwidth <value>**

Use the `bandwidth` command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the `no` form of this command to restore the default values.

**Syntax Description**

| <value> | Specifies the bandwidth value in kbps. |

**Default Values**

To view default values, use the `show interfaces` command.

**Functional Notes**

The `bandwidth` command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the PPP interface to 10 Mbps:

ProCurve(config)#`interface ppp 1`
ProCurve(config-ppp 1)#`bandwidth 10000`
**bridge-group <group#>**

Use the `bridge-group` command to assign an interface to the specified bridge group. This command is supported on all Ethernet interfaces, PPP virtual interfaces, and Frame Relay virtual sub-interfaces.

**Syntax Description**

<group#> Assigns a bridge group number (range: 1 to 255).

**Default Values**

By default, there are no configured bridge groups.

**Functional Notes**

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (Ethernet to T1 bridge, Ethernet to Frame Relay sub-interface, etc.).

**Usage Examples**

The following example assigns the PPP interface to bridge-group 1:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#bridge-group 1
bridge-group <group#> vlan-transparent

Use the bridge-group vlan-transparent command to prevent an interface from removing the VLAN tag. Use the no form of this command to allow the interface to remove the VLAN tag from the packet.

**Note** The bridge-group vlan-transparent command is not a global command. The command must be applied on all interfaces of the bridge group.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group#&gt;</td>
<td>Specifies the bridge group number. Valid range is 1 to 255.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, VLAN tags are removed from the data.

**Usage Examples**

The following example removes the VLAN tags from the packets on the PPP interface labeled 1:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#no bridge-group 1 vlan-transparent
crypto map <mapname>

Use the crypto map command to associate crypto maps with the interface.

**Note**

When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the VPN Configuration Guide located on the ProCurve SROS Documentation CD provided with your unit.

**Syntax Description**

<mapname> Assigns a crypto map name to the interface.

**Default Values**

By default, no crypto maps are assigned to an interface.

**Functional Notes**

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the unencrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the previous diagram, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only unencrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, unencrypted source of the data. The destination information will be the far end, unencrypted destination of the data. However, ACLs for a policy class work in reverse. The source information for the ACL in a policy class is the far end. The destination information is the local side.

**Usage Examples**

The following example applies all crypto maps with the name **MyMap** to the PPP interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#crypto map MyMap
```
dynamic-dns

Use the dynamic-dns command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the no versions of these commands to disable these features. Variations of this command include:

- **dynamic-dns custom** `<hostname> <minutes>`
- **dynamic-dns dyndns** `<hostname> <username> <password>`
- **dynamic-dns dyndns-custom** `<hostname> <username> <password>`
- **dynamic-dns dyndns-static** `<hostname> <username> <password>`

### Syntax Description

- `<hostname>`
  - Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).
- `<minutes>`
  - Specifies the intervals in minutes to update the server with information (updates also occur when the interface's IP address changes regardless of the update intervals).

Refer to **Functional Notes** below for additional argument descriptions.

### Default Values

No default is necessary for this command.

### Functional Notes

- **custom** - Constanttime.com’s Custom Dynamic DNS℠ service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows to manage IP address mappings (A records), domain aliases (CNAME records) and mail servers (MX records).

- **dyndns** - The Dynamic DNS℠ service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- **dyndns-custom** - DynDNS.org’s Custom DNS℠ service provides a full DNS solution, giving you complete control over an entire domain name. A web-based interface provides two levels of control over your domain, catering to average or power users. Five globally redundant DNS servers ensure that your domain will always resolve.

A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background, and provides layout and functionality similar to a BIND zone file allowing for the creation of nearly any record type.

Custom DNS℠ can be used with both static and dynamic IPs, and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
**dyndns-static** - The Static DNS service is similar to Dynamic DNS service, in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate though the DNS system. This service is provided for up to five hostnames.

If your IP address doesn't change often or at all, but you still want an easy name to remember it by (without having to purchase your own domain name) Static DNS service is ideal for you.

If you would like to use your own domain name (such as yourname.com) you need Custom DNS service which also provides full dynamic and static IP address support.

**Usage Examples**

The following example sets the **dynamic-dns** to **dyndns-custom** with hostname **host**, username **user**, and password **pass**:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#dynamic-dns dyndns-custom host user pass
**fair-queue <threshold>**

Use the `fair-queue` command to enable weighted fair queuing (WFQ) on an interface. Use the `no` form of this command to disable WFQ and enable FIFO queueing for an interface. WFQ is enabled by default for WAN interfaces.

**Syntax Description**

| <threshold> | Optional. Specifies the maximum number of packets that can be present in each conversation sub-queue. Packets received for a conversation after this limit is reached are discarded. Range: 16 to 512 packets. |

**Default Values**

By default, `fair-queue` is enabled with a threshold of 64 packets.

**Usage Examples**

The following example enables WFQ on the interface with a threshold set at 100 packets:

```bash
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#fair-queue 100
```
hold-queue <queue size> out

Use the hold-queue out command to change the overall size of an interface's WAN output queue. Use the no form of this command to return to the default setting.

Syntax Description

<queue size> specifies the total number of packets the output queue can contain before packets are dropped. Range 16 to 1000.

Default Values

The default queue size for WFQ is 400. The default queue size for PPP FIFO and Frame Relay round-robin is 200.

Usage Examples

The following example sets the overall output queue size to 700:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#hold-queue 700 out
**ip access-group** `<listname>` [in | out]

Use the `ip access-group` command to create an access list to be used for packets transmitted on or received from the specified interface. Use the `no` form of this command to disable this type of control.

**Syntax Description**

- `<listname>` Indicates assigned IP access list name.
- `in` Enables access control on packets received on the specified interface.
- `out` Enables access control on packets transmitted on the specified interface.

**Default Values**

By default, these commands are disabled.

**Functional Notes**

When this command is enabled, the IP destination address of each packet must be validated before being passed through to the router system. If the packet is not acceptable per these settings, it is dropped.

**Usage Examples**

The following example sets up the router to only allow Telnet traffic into the PPP interface:

```
ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#interface ppp 1
ProCurve(config-ppp 1)#ip access-group TelnetOnly in
```
**ip address dhcp**

Use the `ip address dhcp` command to use Dynamic Host Configuration Protocol (DHCP) to obtain an address on the Ethernet interface. Use the `no` form of this command to remove a configured IP address (using DHCP) and disable DHCP operation on the interface.

```
ip address dhcp [client-id [<interface> | <identifier>] hostname <"string"> ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>client-id</strong></td>
<td>Optional. Specifies the client identifier used when obtaining an IP address from a DHCP server.</td>
</tr>
<tr>
<td><strong>&lt;interface&gt;</strong></td>
<td>Specifying an interface defines the client identifier as the hexadecimal MAC address of the specified interface (including a hexadecimal number added to the front of the MAC address to identify the media type).</td>
</tr>
<tr>
<td></td>
<td>For example, specifying the <code>client-id ethernet 0/1</code> (where the Ethernet interface has a MAC address of 0012.7991.1150) defines the client identifier as <code>01:00:12:79:91:11:50</code> (where 01 defines the media type as Ethernet). Refer to <code>hardware-address &lt;hardware-address&gt; &lt;type&gt;</code> on page 1287 for a detailed listing of media types.</td>
</tr>
<tr>
<td><strong>&lt;identifier&gt;</strong></td>
<td>Specifies a custom client-identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters).</td>
</tr>
<tr>
<td></td>
<td>For example, a custom client identifier of <code>0f:ff:ff:ff:ff:51:04:99:a1</code> may be entered using the <code>&lt;identifier&gt;</code> option.</td>
</tr>
<tr>
<td><strong>hostname</strong></td>
<td>Specifies a text string (to override the global router name) to use as the name in the DHCP option 12 field.</td>
</tr>
<tr>
<td><strong>&lt;&quot;string&quot;&gt;</strong></td>
<td>String (encased in quotation marks) of up to 35 characters to use as the name of the host for DHCP operation.</td>
</tr>
<tr>
<td><strong>no-default-route</strong></td>
<td>Keyword used to specify that the SROS not install the default-route obtained via DHCP.</td>
</tr>
<tr>
<td><strong>no-domain-name</strong></td>
<td>Keyword used to specify that the SROS not install the domain-name obtained via DHCP.</td>
</tr>
<tr>
<td><strong>no-nameservers</strong></td>
<td>Keyword used to specify that the SROS not install the DNS servers obtained via DHCP.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>client-id</strong></td>
<td>Optional. By default, the client identifier is populated using the following formula: TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS</td>
</tr>
<tr>
<td></td>
<td>Where TYPE specifies the media type in the form of one hexadecimal byte (refer to <code>hardware-address &lt;hardware-address&gt; &lt;type&gt;</code> on page 1287 for a detailed listing of media types) and the MAC ADDRESS is the Media Access Control (MAC) address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to Ethernet 0/1 is used in this field).</td>
</tr>
</tbody>
</table>
hostname
<"string"/>

By default, the hostname is the name configured using the Global Configuration hostname command.

Functional Notes
Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the client-id and hostname fields.

Usage Examples
The following example enables DHCP operation on the PPP sub-interface 1.1:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip address dhcp
ip address negotiated [no-default]

Use the `ip address negotiated` command to allow the interface to negotiate (i.e., be assigned) an IP address from the far end PPP connection. Use the `no` form of this command to disable the negotiation for an IP address.

**Syntax Description**

| no-default | Optional. Prevents the insertion of a default route. Some systems already have a default route configured and need a static route to the PPP interface to function correctly. |

**Default Values**

By default, the interface is assigned an address with the `ip address <address><mask>` command.

**Usage Examples**

The following example enables the PPP interface to negotiate an IP address from the far end connection:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip address negotiated
```

The following example enables the PPP interface to negotiate an IP address from the far end connection without inserting a default route:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip address negotiated no-default
```
ip address <address> <mask> [secondary]

Use the `ip address` command to define an IP address on the specified interface. Use the optional `secondary` keyword to define a secondary IP address. Use the `no` form of this command to remove a configured IP address.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).</td>
</tr>
<tr>
<td><code>&lt;mask&gt;</code></td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
<tr>
<td><code>secondary</code></td>
<td>Optional. Configures a secondary IP address for the specified interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no assigned IP addresses.

**Functional Notes**

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

**Usage Examples**

The following example configures a secondary IP address of 192.168.72.101/30:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip address 192.168.72.101 255.255.255.252 secondary
```

To view configured secondary IP address(es), use the `show running-config` command. Secondary IP address information is not available through other `show` commands.
ip helper-address <address>

Use the ip helper-address command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the no form of this command to disable forwarding packets.

Syntax Description

<address>  Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

Default Values

By default, broadcast UDP packets are not forwarded.

Functional Notes

When used in conjunction with the ip forward-protocol command, the ip helper-address feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign helper address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the ip forward-protocol command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the ip helper-address feature:
1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the ip forward-protocol command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

Usage Examples

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.168.5.99:

ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip helper-address 192.168.5.99

Note

The ip helper command must be used in conjunction with the ip forward-protocol command to configure the SROS forward UDP broadcast packets. Refer to ip forward-protocol udp <port number> on page 387 for more information.
**ip igmp**

Use the `ip igmp` command to configure multicasting-related functions for the interface.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>immediate-leave</td>
<td>Specifies that if only one host (or IGMP snooping switch) is connected to the interface, when a leave is received, multicast of that group is immediately terminated as opposed to sending a group query and timing out the group if no device responds. Applies to all groups when configured.</td>
</tr>
<tr>
<td>last-member-query-interval <code>&lt;milliseconds&gt;</code></td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group's multicast packets to the interface. Range: 100 to 65535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td>querier-timeout <code>&lt;seconds&gt;</code></td>
<td>Specifies the number of seconds that the router waits after the current querier's last query before it takes over as querier (IGMP V2). Range: 60-300 seconds. Default: 2x the <strong>query-interval</strong> value.</td>
</tr>
<tr>
<td>query-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td>query-max-response-time <code>&lt;seconds&gt;</code></td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td>static-group <code>&lt;group-address&gt;</code></td>
<td>Configures the router's interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>version `[1</td>
<td>2]`</td>
</tr>
</tbody>
</table>

### Usage Examples

The following example sets the query message interval on the interface to 200 milliseconds:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip igmp last-member-query-interval 200
```
ip mcast-stub downstream

Use the **ip mcast-stub downstream** command to enable multicast forwarding and IGMP (router mode) on an interface and place it in multicast stub downstream mode. Use the **no** form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the **ip mcast-stub helper-address** and **ip mcast-stub upstream** commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. Refer to **ip mcast-stub helper-address <ip address>** on page 397 and **ip mcast-stub upstream** on page 1016 for more information.

**Usage Examples**

The following example enables multicast forwarding and IGMP on the interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip mcast-stub downstream
```
**ip mcast-stub fixed**

Use the `ip mcast-stub fixed` command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the `no` form of this command to disable this mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Multicast routing must be enabled prior to setting `ip mcast-stub fixed` on the selected interface. Also, use the `ip igmp static-group <A.B.C.D>` command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

**Usage Examples**

The following example enables multicast traffic forwarding and IGMP on the interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip mcast-stub fixed
```
ip mcast-stub helper-enable

Use the ip mcast-stub helper-enable command to assign the ip mcast-stub helper-address as the IGMP proxy. Use the no form of this command to disable.

Syntax Description
No subcommands.

Default Values
By default, this command is disabled.

Functional Notes
This command is used in IP multicast stub applications in conjunction with the ip mcast-stub helper-address, ip mcast-stub upstream, and ip mcast-stub downstream commands. When enabled, the interface becomes a helper forwarding interface. The IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the unit to perform as an IGMP proxy.

Usage Examples
The following example sets the helper address as the IGMP proxy:

    ProCurve(config)#interface ppp 1
    ProCurve(config-ppp 1)#ip mcast-stub helper-enable
ip mcast-stub upstream

Use the `ip mcast-stub upstream` command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the `no` form of this command to disable.

Syntax Description

No subcommands.

Default Values

By default, this command is disabled.

Functional Notes

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub downstream` commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router's unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface.

Usage Examples

The following example enables multicast forwarding on the interface:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip mcast-stub upstream
ip ospf

Use the `ip ospf` command to customize OSPF settings (if needed).

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication-key</td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td>cost <code>&lt;value&gt;</code></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32767.</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td>message-digest-key <code>&lt;keyid&gt;</code> md5 <code>&lt;key&gt;</code></td>
<td>Configures OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td>priority <code>&lt;value&gt;</code></td>
<td>Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32,767.</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32,767.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>5 seconds</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>1 second</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>10 seconds: Ethernet, point-to-point, Frame Relay, and PPP</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>40 seconds</td>
</tr>
</tbody>
</table>

**Usage Example**

The following example sets the maximum number of seconds allowed between hello packets to 25,000:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip ospf dead-interval 25000
```
ip ospf authentication [message-digest | null]

Use the ip ospf authentication command to authenticate an interface that is performing OSPF authentication.

Syntax Description

message-digest  Optional. Selects message-digest authentication type.
null           Optional. Specifies that no authentication be used.

Default Values

By default, ip ospf authentication is set to null (meaning no authentication is used).

Usage Examples

The following example specifies that no authentication will be used on the PPP interface:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip ospf authentication null
ip ospf network [broadcast | point-to-point]

Use the `ip ospf network` command to specify the type of network on this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcast</td>
<td>Sets the network type for broadcast.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Sets the network type for point-to-point.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

**Functional Notes**

A point-to-point network will not elect designated routers.

**Usage Examples**

The following example designates a broadcast network type:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip ospf network broadcast
```
**ip pim sparse-mode**

Use the `ip pim sparse-mode` command to enable PIM Sparse Mode on the interface. Use the `no` form of this command to disable PIM Sparse Mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, PIM Sparse Mode is disabled for all interfaces.

**Functional Notes**

PIM Sparse Mode is a multicast routing protocol that makes use of the unicast forwarding table. PIM-systems builds unidirectional shared trees rooted at a Rendezvous Point (RP) for a multicast group or a shortest-path tree rooted at a specific source for a multicast group.

**Usage Examples**

The following example enables PIM sparse-mode on the ppp 1 interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip pim sparse-mode
```
**ip pim-sparse dr-priority** `<value>`

Use the `ip pim-sparse dr-priority` command to specify the priority of this PIM interface for use when selecting the designated router (DR). Use the `no` form of this command to return to the default value.

**Syntax Description**

| `<value>` | Specifies the priority of this interface (to be used when determining the DR). Valid range is 1 to 4,294,967,295. |

**Default Values**

By default, the priority of all PIM interfaces is 1.

**Functional Notes**

Interfaces advertise their configured priority values in the hello messages transmitted on the interface. Routers use the priority values to determine the appropriate DR. The router on the network segment with the highest priority is selected as the DR. If a hello message is received on the interface from a router on the network segment and it does not contain a priority, the entire network segment defaults to DR selection based on IP addresses instead of priority. In this instance, the DR is selected as the router on the network segment that has the highest IP address. ProCurve Secure Routers will always include a priority in all transmitted hello messages. If no priority is specifically designated by the user, the priority is set as the default of 1.

**Usage Examples**

The following example specifies a priority of 100 on the ppp 1 interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip pim-sparse dr-priority 100
```
ip pim-sparse hello-timer <time>

Use the ip pim-sparse hello-timer command to set the time interval at which periodic hello messages are transmitted out the interface. Each PIM interface has an independent hello-timer. Use the no form of this command to return to the default value.

Syntax Description

<time> Specifies the interval (in seconds) at which periodic hellos are sent out the interface. Valid range is 10 to 3600 seconds.

Default Values

By default, hellos are transmitted on PIM interfaces every 60 seconds.

Functional Notes

Hello messages are used to inform neighbors of a router's presence. Hello messages normally generate a small amount of traffic on an interface. Setting the hello-timer to a small interval increases the amount of hellos sent (thus increasing the amount of traffic). Set the hello-timer to a reasonable value, taking into consideration the bandwidth available on the interface.

Usage Examples

The following example specifies hellos to be sent on the PPP interface every 3600 seconds:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip pim-sparse hello-timer 3600
ip pim-sparse nbr-timeout <time>

Use the ip pim-sparse nbr-timeout command to specify the interval the PIM interface waits before declaring that the neighbor is not present. Use the no form of this command to return to the default value.

Syntax Description

| <time> | Specifies the time interval (in seconds) the PIM interface waits before a neighbor is considered not present. Valid range is 30 to 10,800 seconds. |

Default Values

By default, the nbr-timeout is set to 105 seconds.

Usage Examples

The following example specifies a wait interval of 360 seconds on the ppp 1 interface:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip pim-sparse nbr-timeout 360
ip pim-sparse override-interval <time>

Use the ip pim-sparse override-interval command to specify the delay interval after a join/prune in which another router on the LAN may override the join/prune. Use the no form of this command to return to the default value.

Syntax Description

| <time> | Specifies the delay interval (in milliseconds) after a join/prune in which another router on the LAN may override the join/prune. Valid range is 0 to 65,535 milliseconds. |

Default Values

By default, the override-interval is set to 2500 milliseconds.

Usage Examples

The following example sets the delay interval for join/prune overrides to 3000 milliseconds on the ppp 1 interface:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip pim-sparse override-interval 3000
**ip pim-sparse propagation-delay <time>**

Use the `ip pim-sparse propagation-delay` command to specify the expected propagation delay for join/prune messages. Set the propagation delay (in milliseconds) to estimate the amount of delay found in the local link. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;time&gt;</th>
<th>Specifies the expected propagation delay in the local link in milliseconds. Valid range is 0 to 32,767 milliseconds.</th>
</tr>
</thead>
</table>

**Default Values**

By default, the `propagation-delay` is set to 500 milliseconds.

**Usage Examples**

The following example sets the expected propagation delay to 300 milliseconds on the ppp 1 interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip pim-sparse propagation-delay 300
```
ip policy route-map <mapname>

Use the ip policy route-map command to assign a policy route map to this interface. Use the no form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

Syntax Description

<mapname> Specifies the route map to associate with this interface.

Default Values

By default, policy-based routing is disabled for all interfaces.

Usage Examples

The following example associates the route map named MyMap with ppp 1 interface:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip policy route-map MyMap
**ip proxy-arp <address> <subnet mask>**

Use the **ip proxy-arp** to enable proxy Address Resolution Protocol (ARP) on the interface. Use the **no** form of this command to disable this feature.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Defines the IP address for the interface in dotted decimal notation (for example, 192.168.73.101).</td>
</tr>
<tr>
<td>&lt;subnet mask&gt;</td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, proxy ARP is enabled.

**Functional Notes**

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all proxy ARP requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

**Usage Examples**

The following enables proxy ARP on the virtual PPP interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip proxy-arp
```
ip rip receive version [1 | 2]

Use the `ip rip receive version` command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th></th>
<th>Accepts only received RIP version 1 packets on the interface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Accepts only received RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip receive version` to specify a RIP version that overrides the `version` (in the Router RIP) configuration.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the virtual PPP interface to accept only RIP version 2 packets:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip rip receive version 2
```
**ip rip send version [1 | 2]**

Use the `ip rip send version` command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transmits only RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Transmits only RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces transmit RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip send version` to specify a RIP version that overrides the `version` (in the Router RIP) configuration.

The SROS only transmits one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the virtual PPP interface to transmit only RIP version 2 packets:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip rip send version 2
```
**ip rip summary-address** `<network address> <network mask>`

Use the `ip rip summary-address` command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the `no` form of this command to disable this mode.

**Syntax Description**

- `<network address>` Specifies the IP address of the network to be summarized.
- `<network mask>` Specifies the network mask to be applied to the specific network to be summarized.

**Default Values**

By default, no manual summarization is applied by RIP.

**Functional Notes**

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the `ip rip summary-address` command. This command is only effective if RIP version 2 is configured.

**Usage Examples**

The following example enables manual summarization on the specified IP address:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip rip summary-address 10.10.123.0 255.255.255.0
```
**ip route-cache**

Use the `ip route-cache` command to enable route caching on the interface. Use the `no` form of this command to disable route caching and return to process switching mode.

**Note**

Using Network Address Translation (NAT) or the SROS firewall capabilities on an interface requires process switching mode (using the `no ip route-cache` command).

**Syntax Description**

No subcommands.

**Default Values**

By default, route caching is enabled on all interfaces.

**Functional Notes**

Route caching allows an IP interface to provide optimum performance when processing IP traffic.

**Usage Examples**

The following example enables route caching on the virtual PPP interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip route-cache
```
**ip unnumbered <interface>**

Use the `ip unnumbered` command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the `no` form of this command to remove the unnumbered configuration.

**Syntax Description**

| <interface> | Specifies the interface (in the format type slot/port) that contains the IP address to use as the source address for all packets transmitted on this interface. |

**Default Values**

By default, all interfaces are configured to use a specified IP address (using the `ip address` command).

**Functional Notes**

If `ip unnumbered` is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying `ip unnumbered eth 0/1` while in the PPP Interface Configuration mode configures the PPP interface to use the IP address assigned to the Ethernet interface for all IP processing. In addition, the SROS uses the specified interface information when sending route updates over the unnumbered interface. Static routes may either use the interface name (ppp 1) or the far-end address (if it will be discovered).

**Usage Examples**

The following example configures the PPP interface (labeled **ppp 1**) to use the IP address assigned to the Ethernet interface (**eth 0/1**):

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip unnumbered eth 0/1
```
keepalive <seconds>

Use the keepalive command to enable the transmission of keepalive packets on the interface and specify the time interval in seconds between transmitted packets.

Syntax Description

| <seconds> | Defines the time interval (in seconds) between transmitted keepalive packets (valid range: 0 to 32,767 seconds). |

Default Values

By default, the time interval between transmitted keepalive packets is 10 seconds.

Functional Notes

If three keepalive packets are sent to an interface with no response, the interface is considered down. To detect interface failures quickly, specify a smaller keepalive time.

Usage Examples

The following example specifies a keepalive time of 5 seconds on the virtual PPP interface:

ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#keepalive 5
**lldp receive**

Use the **lldp receive** command to allow LLDP packets to be received on this interface.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces are configured to send and receive LLDP packets.

**Usage Examples**

The following example configures the PPP interface to receive LLDP packets:

```plaintext
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#lldp receive
```
lldp send [management-address | port-description | system-capabilities | system-description | system-name]

Use the `lldp send` command to configure this interface to transmit LLDP packets or to control the types of information contained in the LLDP packets transmitted by this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>management-address</td>
<td>Enables transmission of management address information on this interface.</td>
</tr>
<tr>
<td>port-description</td>
<td>Enables transmission of port description information on this interface.</td>
</tr>
<tr>
<td>system-capabilities</td>
<td>Enables transmission of this device’s system capabilities on this interface.</td>
</tr>
<tr>
<td>system-description</td>
<td>Enables transmission of this device’s system description on this interface.</td>
</tr>
<tr>
<td>system-name</td>
<td>Enables transmission of this device’s system name on this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

Be default, all interfaces are configured to transmit and receive LLDP packets of all types.

**Functional Notes**

Individual LLDP information can be enabled or disabled using the various forms of the `lldp send` command. For example, use the `lldp send-and-receive` command to enable transmit and receive of all LLDP information. Then use the `no lldp send port-description` command to prevent LLDP from transmitting port description information.

**Usage Examples**

The following example configures the PPP interface to transmit LLDP packets containing all enabled information types:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#lldp send
```
**lldp send-and-receive**

Use the **lldp send-and-receive** command to configure this interface to transmit and receive LLDP packets.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces are configured to transmit and receive LLDP packets of all types.

**Functional Notes**

Individual LLDP information can be enabled or disabled using the various forms of the **lldp send** command. For example, use the **lldp send-and-receive** command to enable transmit and receive of all LLDP information. Then use the **no lldp send port-description** command to prevent LLDP from transmitting port description information.

**Usage Examples**

The following example configures the PPP interface (ppp 16) to transmit and receive LLDP packets containing all information types:

```
ProCurve(config)#interface ppp 16
ProCurve(config-ppp 16)#lldp send-and-receive
```
max-reserved-bandwidth <percent>

Use the max-reserved-bandwidth command to specify the percentage of interface bandwidth reserved for use in user-defined (priority or class-based) queues. The remainder of the interface bandwidth is reserved for system critical traffic and is not available to user-defined queues. Use the no form of this command to restore the default value.

Caution  Reserving a portion of the interface bandwidth for system critical traffic is necessary for proper operation. Specifying the entire interface bandwidth for use in user-defined queues can cause undesirable operation.

Syntax Description

<percent>Specifies the percentage of interface bandwidth to make available for user-defined (priority or class-based) queues. Enter an integer 1 to 100.

Default Values

By default, max-reserved-bandwidth is set to 75 which reserves 25 percent of the interface bandwidth for system critical traffic.

Usage Examples

The following example specifies 85 percent of the bandwidth on the ppp 1 interface to be available for use in user-defined queues:

(config)#interface ppp 1
(config-ppp 1)#max-reserved-bandwidth 85
**mtu <size>**

Use the `mtu` command to configure the maximum transmit unit (MTU) size for the active interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>64 to 256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>64 to 18,190</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>64 to 1500</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>The default values for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>1500</td>
</tr>
</tbody>
</table>

**Functional Notes**

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

**Usage Examples**

The following example specifies an MTU of 1200 on the virtual PPP interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#mtu 1200
```
peer default ip address <address>

Use the peer default ip address command to specify the default IP address of the remote end of this interface.

**Syntax Description**

| <address> | Specifies the default IP address for the remote end (A.B.C.D). |

**Default Values**

By default, there is no assigned peer default IP address.

**Functional Notes**

This command is useful if the peer does not send the IP address option during PPP negotiations.

**Usage Examples**

The following example sets the default peer IP address to 192.168.71.50:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#peer default ip address 192.168.71.50
```
ppp authentication [chap | pap]

Use the **ppp authentication** command to specify the authentication protocol on the PPP virtual interface that the peer should use to authenticate itself.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chap</td>
<td>Configures CHAP authentication on the interface.</td>
</tr>
<tr>
<td>pap</td>
<td>Configures PAP authentication on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, PPP endpoints have no authentication configured.

**Technology Review**

CHAP and PAP are two authentication methods that enjoy widespread support. Both methods are included in the SROS and are easily configured.

**Note**

The authentication method set up on the local router can be different from that on the peer. Also, just because one router requires authentication from its peer does not mean it also has to authenticate itself to the peer.

**Defining PAP**

The Password Authentication Protocol (PAP) is used to verify that the PPP peer is a permitted device by checking a username and password configured on the peer. The username and password are both sent unencrypted across the connecting private circuit.

PAP requires two-way message passing. First, the router that is required to be authenticated (say the peer) sends an authentication request with its username and password to the router requiring authentication (say the local router). The local router then looks up the username and password in the username database within the PPP interface, and if they match sends an authentication acknowledge back to the peer.

**Note**

The PPP username and password database is separate and distinct from the global username password database. For PAP and CHAP, use the database under the PPP interface configuration.

Several example scenarios are given below for clarity.

**Configuring PAP Example 1: Only the local router requires the peer to authenticate itself.**

On the local router (hostname Local):

Local(config-ppp 1)#ppp authentication pap
Local(config-ppp 1)#username farend password same

On the peer (hostname Peer):

Peer(config-ppp 1)#ppp pap sent-username farend password same
The first line of the configuration sets the authentication mode as PAP. This means the peer is required to authenticate itself to the local router via PAP. The second line is the username and password expected to be sent from the peer. On the peer, the `ppp pap sent-username` command is used to specify the appropriate matching username and password.

**Configuring PAP Example 2: Both routers require the peer to authenticate itself.**

On the local router (hostname Local):

```
Local(config-ppp 1)#ppp authentication pap
Local(config-ppp 1)#username farend password far
Local(config-ppp 1)#ppp pap sent-username nearend password near
```

On the peer (hostname Peer):

```
Peer(config-ppp 1)#ppp authentication pap
Peer(config-ppp 1)#username nearend password near
Peer(config-ppp 1)#ppp pap sent-username farend password far
```

Now both routers send the authentication request, verify that the username and password sent match what is expected in the database, and send an authentication acknowledge.

**Defining CHAP**

The Challenge-Handshake Authentication Protocol (CHAP) is a three-way authentication protocol composed of a challenge response and success or failure. The MD5 protocol is used to protect usernames and passwords in the response.

First, the local router (requiring its peer to be authenticated) sends a "challenge" containing only its own unencrypted username to the peer. The peer then looks up the username in the username database within the PPP interface, and if found takes the corresponding password and its own hostname and sends a "response" back to the local router. This data is encrypted. The local router verifies that the username and password are in its own username database within the PPP interface, and if so sends a "success" back to the peer.

```
Note: The PPP username and password database is separate and distinct from the global username password database. For PAP and CHAP, use the database under the PPP interface configuration.
```

Several example scenarios are given below for clarity.

**Configuring CHAP Example 1: Only the local router requires the peer to authenticate itself.**

On the local router (hostname Local):

```
Local(config-ppp 1)#ppp authentication chap
Local(config-ppp 1)#username Peer password same
```

On the peer (hostname Peer):

```
Peer(config-ppp 1)#username Local password same
```

The first line of this configuration sets the authentication mode to CHAP. This means the peer is required to authenticate itself to the local router via CHAP. The second line is the username and password
expected to be sent from the peer. The peer must also have the username up both to verify the incoming username from the local router and to use the password (along with its hostname) in the response to the local router.

**Note** Both ends must have identical passwords.

**Configuring CHAP Example 2: Both routers require the peer to authenticate itself.**

On the local router (hostname Local):

Local(config-ppp 1)#ppp authentication chap
Local(config-ppp 1)#username Peer password same

On the peer (hostname Peer):

Peer(config-ppp 1)#ppp authentication chap
Peer(config-ppp 1)#username Local password same

This is basically identical to Example 1 except that both routers will now challenge each other and respond.

**Configuring CHAP Example 3: Using the ppp chap hostname command as an alternate solution.**

On the local router (hostname Local):

Local(config-ppp 1)#ppp authentication chap
Local(config-ppp 1)#username Peer password same
Local(config-ppp 1)#ppp chap hostname nearend

On the peer (hostname Peer):

Peer(config-ppp 1)#username nearend password same

Notice the peer is expecting username “nearend” even though the local router's hostname is “Local.” Therefore the local router can use the ppp chap hostname command to send the correct name on the challenge.

**Configuring CHAP Example 4: Using the ppp chap password command as an alternate solution.**

On the local router (hostname Local):

Local(config-ppp 1)#ppp authentication chap
Local(config-ppp 1)#username Peer password different

On the peer (hostname Peer):

Peer(config-ppp 1)#username Local password same
Peer(config-ppp 1)#ppp chap password different

Here the local router challenges with hostname "Local." The peer verifies the name in the username database, but instead of sending the password "same" in the response, it uses the one in the ppp chap password command. The local router then verifies that user "Peer" with password "different" is valid and sends a “success.”
**ppp chap hostname <hostname>**

Use the `ppp chap hostname` command to configure an alternate hostname for CHAP PPP authentication. Use the `no` form of this command to remove a configured hostname. For more information on PAP and CHAP functionality, refer to the *Technology Review* section for the command `ppp authentication [chap l pap]` on page 1040.

**Syntax Description**

| <hostname> | Specifies a hostname by alphanumeric string up to 80 characters in length. |

**Default Values**

By default, there are no configured PPP CHAP hostnames.

**Usage Examples**

The following example specifies a PPP CHAP hostname of `my_host`:

```
ProCurve(config)#ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ppp chap hostname my_host
```
**ppp chap password** `<password>`

Use the `ppp chap password` command to configure an alternate password when the peer requires CHAP PPP authentication. Use the `no` form of this command to remove a configured password. For more information on PAP and CHAP functionality, refer to the *Technology Review* section for the command `ppp authentication [chap | pap]` on page 1040.

**Syntax Description**

`<password>` Specifies a password by alphanumeric string up to 80 characters in length.

**Default Values**

By default, there is no defined PPP CHAP password.

**Usage Examples**

The following example specifies a PPP CHAP password of `my_password`:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ppp chap password my_password
```
**ppp multilink**

Use the `ppp multilink` command to enable multilink PPP (MPPP) operation. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, MPPP is disabled.

**Functional Notes**

When enabled, this interface is capable of the following:

- Combining multiple physical links into one logical link.
- Receiving upper layer protocol data units (PDU), fragmenting and transmitting over the physical links based upon the physical link MTU.
- Receiving fragments over the physical links and reassembling them into PDUs.

**Usage Examples**

The following example enables MPPP:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ppp multilink
```
**ppp multilink [fragmentation | interleave]**

Use the `ppp multilink` command to enable multilink PPP (MPPP) operation on an existing PPP interface. Use the `no` form of this command to disable.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fragmentation</td>
<td>Enables multilink fragmentation operation.</td>
</tr>
<tr>
<td>interleave</td>
<td>Enables multilink interleave operation.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, MPPP is disabled.

**Functional Notes**

When enabled, this interface is capable of the following:

- Combining multiple physical links into one logical link.
- Receiving upper layer protocol data units (PDU), fragmenting and transmitting over the physical links.
- Receiving fragments over the physical links and reassembling them into PDUs.

The fragmentation and interleave options can be used to enhance the multilink operation. Fragmentation is used to reduce serialization delays of large packets. The fragmentation process evenly divides the data among all links in the bundle with a minimum packet size of 96 bytes. The interleave operation is used with streaming protocols to reduce delay by giving priority to packets identified as high priority. In order delivery is guaranteed with multilink fragmentation, but is not guaranteed with multilink interleave operation.

The multilink bundle will remain active with a minimum of one physical link. Physical links may be dynamically added or removed from the multilink bundle with minor interruption to traffic flow.

**Usage Examples**

The following example enables MLPPP:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ppp multilink
```
**ppp pap sent-username** `<username>` **password** `<password>`

Use the `ppp pap sent-username`/`password` command to configure a username and password when the peer requires PAP PPP authentication. Use the `no` form of this command to remove a configured password. For more information on PAP and CHAP functionality, refer to the *Technology Review* section for the command `ppp authentication [chap l pap]` on page 1040.

**Syntax Description**

- `<username>`: Specifies a username by alphanumeric string up to 80 characters in length (the username is case-sensitive).
- `<password>`: Specifies a password by alphanumeric string up to 80 characters in length (the password is case-sensitive).

**Default Values**

By default, there is no defined `ppp pap sent-username` and `password`.

**Usage Examples**

The following example specifies a PPP PAP sent-username of `local` and a password of `my_password`:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ppp pap sent-username local password my_password
```
**pppoe ac-name <name>**

Use the `pppoe ac-name` command to identify the Access Concentrator (AC) with which the SROS expects to establish a PPPoE session. Use the `no` form of this command to return to the default setting.

**Syntax Description**

| <name> | Specifies an AC by text string (up to 255 characters) corresponding to the AC-Name Tag under RFC 2516. If this field is not specified, any access concentrator is acceptable. The AC value may be a combination of trademark, model, and serial ID information (or simply the MAC address of the unit). |

**Default Values**

By default, no AC is specified.

**Usage Examples**

The following example identifies the AC with which the SROS expects to establish a PPPoE session:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#pppoe acc-name Access_Concentrator_Name
```
**pppoe service-name** <name>

Use the **pppoe service-name** command to use this tag value to filter PPPoE session offers from PPPoE servers. Use the **no** form of this command to return to the default setting.

**Syntax Description**

| <name> | Specifies a service name by text string (up to 255 characters) corresponding to the Service-Name Tags under RFC 2516. This string indicates an ISP name (or a class or quality of service). If this field is not specified, any service is acceptable. |

**Default Values**

By default, no names are specified.

**Usage Examples**

The following example defines a service type that is not to be accepted by the SROS:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#pppoe service-name Service_Name
```
**qos-policy out** `<mapname>`

Use the `qos-policy out` command to apply a previously-configured QoS map to an interface. Use the `no` form of this command to remove the map from the interface. The keyword `out` specifies that this policy will be applied to outgoing packets.

**Syntax Description**

`<mapname>`

Specifies the name of a previously-created QoS map (refer to `qos map` `<mapname>` `<sequence number>` on page 451 for more information).

**Default Values**

No default value is necessary for this command.

**Usage Examples**

The following example applies the QoS map `VOICEMAP` to the PPP 1 interface:

```bash
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#qos-policy out VOICEMAP
```
**snmp trap link-status**

Use the `snmp trap link-status` command to control the Simple Network Management Protocol (SNMP) variable `ifLinkUpDownTrapEnable` (RFC2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the `no` form of this command to disable this trap.

**Syntax Description**

No subcommands.

**Default Values**

By default, the `ifLinkUpDownTrapEnable` OID is enabled for all interfaces except virtual Frame Relay interfaces.

**Functional Notes**

The `snmp trap link-status` command is used to control the RFC2863 `ifLinkUpDownTrapEnable` OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

**Usage Examples**

The following example disables the link-status trap on the virtual PPP interface:

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#no snmp trap link-status
```
username <username> password <password>

Configures the username and password of the peer to use for PPP authentication.

**Syntax Description**

| `<username>` | Specifies a username by alphanumerical string up to 30 characters in length (the username is case-sensitive). |
| `<password>` | Specifies a password by alphanumerical string up to 30 characters in length (the password is case-sensitive). |

**Default Values**

By default, there is no established username and password.

**Functional Notes**

PAP uses this entry to check received information from the peer. CHAP uses this entry to check the received peer hostname and a common password.

**Usage Examples**

The following example creates a username of PROCURVE with password ROUTER for the PPP link labeled 5:

```plaintext
ProCurve(config)#interface ppp 5
ProCurve(config-ppp 5)#username PROCURVE password ROUTER
```
TUNNEL CONFIGURATION COMMAND SET

To activate the Tunnel Configuration mode, enter the **interface tunnel** command at the Global Configuration mode prompt. For example:

ProCurve>**enable**
ProCurve#**configure terminal**
ProCurve(config)#**interface tunnel 1**
ProCurve(config-tunnel 1)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- alias <"text"> on page 1303
- bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port> on page 1304
- description <text> on page 1307
- do on page 1308
- end on page 1309
- exit on page 1310
- ping <address> on page 1311
- shutdown on page 1315

All other commands for this command set are described in this section in alphabetical order.

- access-policy < policymap > on page 1055
- backup commands begin on page 1056
- bandwidth < value > on page 1072
- crypto map < mapname > on page 1073
- dynamic-dns on page 1075
- ip commands begin on page 1078
- keepalive < period > < retries > on page 1100
- lldp receive on page 1101
- lldp send [management-address l port-description l system-capabilities l system-description l system-name] on page 1102
- lldp send-and-receive on page 1103
- mtu < size > on page 1104
- spanning-tree path-cost < value > on page 1105
- tunnel checksum on page 1106
tunnel destination <ip address> on page 1107

Tunnel key <value> on page 1108

tunnel mode gre on page 1109

Tunnel sequence-datagrams on page 1110

Tunnel source [<ip address> | <interface>] on page 1111
access-policy <policyname>

Use the access-policy command to assign a specified access policy for the inbound traffic on an interface. Use the no form of this command to remove an access policy association.

**Note** Configured access policies will only be active if the ip firewall command has been entered at the Global Configuration mode prompt to enable the SROS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.

**Syntax Description**

| <policyname> | Identifies the configured access policy by alphanumeric descriptor (all access policy descriptors are case-sensitive). |

**Default Values**

By default, there are no configured access policies associated with an interface.

**Functional Notes**

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter access policy <policy name>. For more details on creating and using access policies, refer to ip policy-class <policyname> on page 400.

**Usage Examples**

The following example associates the access policy UnTrusted (to allow inbound traffic to the Web server) to the tunnel interface labeled 1:

Enable the SROS security features:
ProCurve(config)#ip firewall

Create the access list (this is the packet selector):
ProCurve(config)#ip access-list extended InWeb
ProCurve(config-ext-nacl)#permit tcp any host 10.12.5.253 eq 80

Create the access policy that contains the access list InWeb:
ProCurve(config)#ip policy-class UnTrusted
ProCurve(config-policy-class)#allow list InWeb

Associate the access list with the interface:
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#access-policy UnTrusted
backup auto-backup

Use the backup auto-backup command to configure the sub-interface to automatically attempt a backup upon failure.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically attempt backup upon a failure.

Usage Examples

The following enables automatic backup on the endpoint:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup auto-backup
backup auto-restore

Use the backup auto-restore command to configure the sub-interface to automatically discontinue backup when all network conditions are operational. Use the no form of this command to disable the auto-restore feature.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically restore the primary connection when the failure condition clears.

Usage Examples

The following configures the SROS to automatically restore the primary connection when the failure condition clears:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup auto-restore
backup backup-delay <seconds>

Use the backup backup-delay command to configure the amount of time the router will wait after the failure condition is recognized before attempting to backup the link. Use the no form of this command to return to the default value.

Syntax Description

| <seconds> | Specifies the delay period (in seconds) a failure must be active before the SROS will enter backup operation on the interface (valid range: 10 to 86400 seconds). |

Default Values

By default, the backup backup-delay is set to 10 seconds.

Usage Examples

The following configures the SROS to wait 60 seconds (on an endpoint with an active alarm condition) before attempting backup operation:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup backup-delay 60
backup call-mode <role>

Use the backup call-mode command to specify whether the configured backup interfaces answers or originates (or a combination of both) backup calls. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;role&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer</td>
<td>Selects the role the router will take in backup of this sub-interface.</td>
</tr>
<tr>
<td>answer-always</td>
<td>Answers and backs up primary link on failure.</td>
</tr>
<tr>
<td>originate</td>
<td>Originates backup call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer</td>
<td>Originates or answers call on primary link failure.</td>
</tr>
<tr>
<td>originate-answer-always</td>
<td>Originates on failure; answers and backs up always.</td>
</tr>
</tbody>
</table>

Default Values

By default, the backup call-mode is set to originate-answer.

Functional Notes

The majority of the configuration for the SROS backup implementation is configured via the backup PPP interface configuration commands. However, the numbers dialed are configured in the primary interface. Full sample configurations follow:

Sample config for remote router (dialing out)

hostname "Remote7203dl"
enable password password
!
interface eth 0/1
   ip address  192.168.1.254  255.255.255.0
   no shutdown
!
interface modem 1/3
   no shutdown
!
interface t1 1/1
   coding b8zs
   framing esf
   clock source line
   tdm-group 1 timeslots 1-24
   no shutdown
!
interface fr 1 point-to-point
  frame-relay lmi-type ansi
  no shutdown
  bind 1 t1 1/1 1 fr 1
!
interface fr 1.16 point-to-point
  frame-relay interface-dlci 16
  ip address 10.1.1.2 255.255.255.252
  backup call-mode originate
  backup number 5551111 analog ppp 1
  backup number 5552222 analog ppp 1
  no shutdown
!
interface ppp 1
  ip address 172.22.56.1 255.255.255.252
  ppp authentication chap
  username remoterouter password remotepass
  ppp chap hostname localrouter
  ppp chap password procurve
  no shutdown
!
ip route 192.168.2.0 255.255.255.0 172.22.56.2 255.255.255.252
!
line telnet 0 4
  password password

**Sample config for central router (dialing in)**

hostname "Central7203dl"
enable password password
!
interface eth 0/1
  ip address 192.168.100.254 255.255.255.0
  no shutdown
!
interface modem 1/3
  no shutdown
!
interface t1 1/1
  coding b8zs
  framing esf
  clock source line
tdm-group 1 timeslots 1-24
no shutdown

interface fr 1 point-to-point
  frame-relay lmi-type ansi
  no shutdown
  bind 1 t1 1/1 1 fr 1

! interface fr 1.100 point-to-point
  frame-relay interface-dlci 100
  ip address 10.1.1.1 255.255.255.252
  backup call-mode answer
  backup number 555-8888 analog ppp 1

! interface ppp 1
  ip address 172.22.56.2 255.255.255.252
  ppp authentication chap
  username localrouter password procurve
  ppp chap hostname remoterouter
  ppp chap password remotepass
  no shutdown

! ip route 192.168.1.0 255.255.255.0 172.22.56.1 255.255.255.252

line telnet 0 4
  password password

Usage Examples
The following configures the SROS to generate backup calls for this endpoint using an analog modem interface (to phone number 555 1111) but never answer calls and specifies ppp 2 as the backup interface:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup call-mode originate
ProCurve(config-tunnel 1)#backup number 555 1111 analog ppp 2
backup connect-timeout <seconds>

Use the `backup connect-timeout` command to specify the number of seconds to wait for a connection after a call is attempted before trying to call again or dialing a different number. Recommended value is greater than 60.

**Syntax Description**

| <seconds>         | Selects the amount of time in seconds that the router will wait for a connection before attempting another call (valid range: 10 to 300). |

**Default Values**

By default, `backup connect-timeout` is set to 60 seconds.

**Usage Examples**

The following configures the SROS to wait 120 seconds before retrying a failed backup call:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup connect-timeout 120
```
backup force [backup | primary]

Use the **backup force** command to manually override the automatic backup feature. This can be used to force a link into backup to allow maintenance to be performed on the primary link without disrupting data. Use the **no** form of this command to return to the normal backup operation state.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backup</td>
<td>Forces backup regardless of primary link state.</td>
</tr>
<tr>
<td>primary</td>
<td>Forces primary link regardless of its state.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this feature is disabled.

**Usage Examples**

The following configures the SROS to force this endpoint into backup:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup force backup
```
backup maximum-retry <attempts>

Use the backup maximum-retry command to select the number of calls the router will make when attempting to backup a link. Use the no form of this command to return to the default state. For more detailed information on tunnel backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 1059.

Syntax Description

| <attempts> | Selects the number of call retries that will be made after a link failure (valid range: 0 to 15). Setting this value to 0 will allow unlimited retries during the time the network is failed. |

Default Values

By default, backup maximum-retry is set to 0 attempts.

Usage Examples

The following example configures the SROS to retry a backup call four times before considering backup operation not available:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup maximum-retry 4
backup number <digits> [analog | digital-56k | digital 64k] <isdn min chan> <isdn max chan> ppp <interface>

Use the backup number command to configure the phone number and the call type the router will dial upon network failure. Multiple entries can be made for an interface to allow alternate sites to be dialed. For more detailed information on backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 1059.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;digits&gt;</td>
<td>Specifies the phone numbers to call when the backup is initiated.</td>
</tr>
<tr>
<td>analog</td>
<td>Indicates number connects to an analog modem.</td>
</tr>
<tr>
<td>digital-56k</td>
<td>Indicates number belongs to a digital 56 kbps per DS0 connection.</td>
</tr>
<tr>
<td>digital-64k</td>
<td>Indicates number belongs to a digital 64 kbps per DS0 connection.</td>
</tr>
<tr>
<td>&lt;isdn min chan&gt;</td>
<td>Specifies the minimum number of DS0s required for a digital 56 or 64 kbps</td>
</tr>
<tr>
<td></td>
<td>connection (Range: 1 to 24).</td>
</tr>
<tr>
<td>&lt;isdn max chan&gt;</td>
<td>Specifies the maximum number of DS0s desired for a digital 56 or 64 kbps</td>
</tr>
<tr>
<td></td>
<td>connection (Range: 1 to 24).</td>
</tr>
<tr>
<td>ppp &lt;interface&gt;</td>
<td>Specifies the PPP interface to use as the backup for this interface.</td>
</tr>
</tbody>
</table>

Default Values

By default, there are no configured backup numbers.

Usage Examples

The following example configures the SROS to dial 704-555-1212 (digital 64 kbps connection) to initiate backup operation for this endpoint using the configured ppp 1 backup interface:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup number 7045551212 digital-64k 1 1 ppp 1
```
backup priority <value>

Use the backup priority command to select the backup priority for this interface. This allows the user to establish the highest priority backup link and ensure that link will override backups attempted by lower priority links. Use the no form of this command to return to the default value. For more detailed information on tunnel backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 1059.

Syntax Description

| <value> | Sets the relative priority of this link (valid range: 0 to 100). A value of 100 designates the highest priority. |

Default Values

By default, backup priority is set to 50.

Usage Examples

The following example assigns the highest priority to this endpoint:

```
ProCurve(config)#interface tunnel 16
ProCurve(config-tunnel 16)#backup priority 100
```
backup randomize-timers

Use the backup randomize-timers command to randomize the call timers to minimize potential contention for resources. Use the no form of this command to return to the default value. For more detailed information on tunnel backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 1059.

Syntax Description

No subcommands.

Default Values

By default, the SROS does not randomize the backup call timers.

Usage Examples

The following example configures the SROS to randomize the backup timers associated with this endpoint:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup randomize-timers
**backup redial-delay <seconds>**

Use the **backup redial-delay** command to configure the delay after an unsuccessful call until the call will be re-tried. For more detailed information on tunnel backup functionality, refer to the *Functional Notes* section of the command **backup call-mode <role>** on page 1059.

**Syntax Description**

| <seconds> | Specifies the delay in seconds between attempting to re-dial a failed backup attempt. Range: 10 to 3600. |

**Default Values**

By default, **backup redial-delay** is set to 10 seconds.

**Usage Examples**

The following example configures a redial delay of 25 seconds on this endpoint:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup redial-delay 25
```
backup restore-delay <seconds>

Use the backup restore-delay command to configure the amount of time the router will wait after the network is restored before disconnecting the backup link and reverting to the primary. This setting is used to prevent disconnecting the backup link if the primary link is “bouncing” in and out of alarm. For more detailed information on tunnel backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 1059.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;seconds&gt;</th>
<th>Specifies the number of seconds the SROS will wait (after a primary link is restored) before disconnecting backup operation. Range: 10 to 86,400.</th>
</tr>
</thead>
</table>

**Default Values**

By default, backup restore-delay is set to 10 seconds.

**Usage Examples**

The following example configures the SROS to wait 30 seconds before disconnecting backup operation and restoring the primary connection for this endpoint:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup restore-delay 30
```
backup schedule [day | enable-time | disable-time]

Use the `backup schedule` command to set the time of day that backup will be enabled. Use this command if backup is desired only during normal business hours and on specific days of the week. Use the `no` form of this command to disable backup (as specified). For more detailed information on tunnel backup functionality, refer to the *Functional Notes* section of the command `backup call-mode <role>` on page 1059.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>Sets the days to allow backup (valid range: Monday through Sunday).</td>
</tr>
<tr>
<td>enable-time</td>
<td>Sets the time of day to enable backup. Time is entered in 24-hour format (00:00).</td>
</tr>
<tr>
<td>disable-time</td>
<td>Sets the time of day to disable backup.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, backup is enabled for all days and times if the `backup auto-backup` command has been issued and the backup schedule has not been entered.

**Usage Examples**

The following example enables backup Monday through Friday 8:00 am to 7:00 pm:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#backup schedule enable-time 08:00
ProCurve(config-tunnel 1)#backup schedule disable-time 19:00
ProCurve(config-tunnel 1)#no backup schedule day Saturday
ProCurve(config-tunnel 1)#no backup schedule day Sunday
```
backup shutdown

Use the backup shutdown command to deactivate all backup functionality in the unit. Backup configuration parameters are kept intact, but the unit will not initiate (or respond) to backup sequences in the event of a network outage. Use the no form of this command to reactivate the backup interface. For more detailed information on tunnel backup functionality, refer to the Functional Notes section of the command backup call-mode <role> on page 1059.

Syntax Description

No subcommands.

Default Values

By default, all SROS interfaces are disabled.

Usage Examples

The following example deactivates the configured backup interface:

ProCurve(config)#interface tunnel 16
ProCurve(config-tunnel 16)#backup shutdown
**bandwidth <value>**

Use the `bandwidth` command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the `no` form of this command to restore the default values.

**Syntax Description**

- `<value>` specifies bandwidth in kbps.

**Default Values**

To view default values, use the `show interfaces` command.

**Functional Notes**

The `bandwidth` command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

**Usage Examples**

The following example sets bandwidth of the tunnel 1 interface to 10 Mbps:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#bandwidth 10000
```
crypto map <mapname>

Use the crypto map command to associate crypto maps with the interface.

**Syntax Description**

<mapname> Enter the crypto map name that you wish to assign to the interface.

**Default Values**

By default, no crypto maps are assigned to an interface.

**Functional Notes**

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the unencrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical SROS data-flow logic.
As shown in the previous diagram, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access-group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only un-encrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, un-encrypted source of the data. The destination information will be the far-end, un-encrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy-class is the far-end. The destination information is the local-side.

Usage Examples
The following example applies all crypto maps with the name MyMap to the tunnel interface:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#crypto map MyMap
dynamic-dns

Use the `dynamic-dns` command to configure Dynamic DNS service provided by Dynamic Network Services, Inc. (www.dyndns.org). Use the `no` versions of these commands to disable these features. Variations of this command include:

- `dynamic-dns custom <hostname> <minutes>`
- `dynamic-dns dyndns <hostname> <username> <password>`
- `dynamic-dns dyndns-custom <hostname> <username> <password>`
- `dynamic-dns dyndns-static <hostname> <username> <password>`

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;hostname&gt;</code></td>
<td>Specifies the hostname for the server that updates the Dynamic Domain Name Server (DNS).</td>
</tr>
<tr>
<td><code>&lt;minutes&gt;</code></td>
<td>Specifies the intervals in minutes to update the server with information (updates also occur when the interface's IP address changes regardless of the update intervals). Refer to <em>Functional Notes</em> below for additional argument descriptions.</td>
</tr>
</tbody>
</table>

**Default Values**

No default is necessary for this command.

**Functional Notes**

- **custom** - Constanttime.com’s Custom Dynamic DNS<sup>SM</sup> service allows you complete access and management control over your domain name regardless of where you purchased/registered it. This allows to manage IP address mappings (A records), domain aliases (CNAME records) and mail servers (MX records).

- **dyndns** - The Dynamic DNS<sup>SM</sup> service allows you to alias a dynamic IP address to a static hostname in various domains. This allows your unit to be more easily accessed from various locations on the Internet. This service is provided for up to five hostnames.

- **dyndns-custom** - DynDNS.org’s Custom DNS<sup>SM</sup> service provides a full DNS solution, giving you complete control over an entire domain. A web-based interface provides two levels of control over your domain, catering to average or advanced users. Five globally redundant DNS servers ensure that your domain will always resolve.

A choice of two interfaces is available. The basic interface is designed for most users. It comes preconfigured for the most common configuration and allows for easy creation of most common record types. The advanced interface is designed for system administrators with a solid DNS background, and provides layout and functionality similar to a BIND zone file allowing for the creation of nearly any record type.

Custom DNS<sup>SM</sup> can be used with both static and dynamic IPs and has the same automatic update capability through Custom DNS-aware clients as Dynamic DNS.
dyndns-static - The Static DNS service is similar to Dynamic DNS service in that it allows a hostname such as yourname.dyndns.org to point to your IP address. Unlike a Dynamic DNS host, a Static DNS host does not expire after 35 days without updates, but updates take longer to propagate though the DNS system. This service is provided for up to five hostnames.

When to Use Dynamic DNS
If your IP address does not change often (or at all) but you still want an easy name to remember it by (without having to purchase your own domain name), Static DNS service is ideal for you.
If you would like to use your own domain name (such as yourname.com), you need Custom DNS service, which also provides full dynamic and static IP address support.

Dynamic DNS service can be extremely helpful for site-to-site VPN connections using dynamic IP addressing. Instead of creating a tunnel to the IP-address, a tunnel to the DNS-name is established.

Usage Examples
The following example sets the dynamic-dns to dyndns-custom with hostname host, username user, and password pass:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#dynamic-dns dyndns-custom host user pass
**ip access-group** `<listname>` [in | out]

Use the **ip access-group** command to create an access list to be used for packets transmitted on or received from the specified interface. Use the **no** form of this command to disable this type of control.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;listname&gt;</code></th>
<th>Assigns an IP access list name.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>in</strong></td>
<td>Enables access control on packets received on the specified interface.</td>
</tr>
<tr>
<td><strong>out</strong></td>
<td>Enables access control on packets transmitted on the specified interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, these commands are disabled.

**Functional Notes**

When this command is enabled, the IP destination address of each packet must be validated before being passed through to the router system. If the packet is not acceptable per these settings, it is dropped.

**Usage Examples**

The following example sets up the unit to only allow Telnet traffic (as defined in the user-configured TelnetOnly IP access list) into the tunnel interface:

```
ProCurve(config)#ip access-list extended TelnetOnly
ProCurve(config-ext-nacl)#permit tcp any any eq telnet
ProCurve(config-ext-nacl)#interface tunnel 1
ProCurve(config-tunnel 1)#ip access-group TelnetOnly in
```
ip address <address> <mask> [secondary]

Use the `ip address` command to define an IP address on the specified interface. Use the `no` form of this command to remove a configured IP address.

Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Defines the IP address for the interface in dotted decimal notation (for example: 192.168.73.101).</td>
</tr>
<tr>
<td>&lt;mask&gt;</td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
<tr>
<td>secondary</td>
<td>Optional. Configures a secondary IP address for the specified interface.</td>
</tr>
</tbody>
</table>

Default Values

By default, there are no assigned IP addresses.

Usage Examples

The following example configures an IP address of 192.168.72.101/30:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip address 192.168.72.101 255.255.255.252

To view configured secondary IP address(es), use the `show running-config` command. Secondary IP address information is not available through other `show` commands.
ip helper-address <address>

Use the `ip helper-address` command to configure the SROS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the `no` form of this command to disable forwarding packets.

**Syntax Description**

<address> Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

**Default Values**

By default, broadcast UDP packets are not forwarded.

**Functional Notes**

When used in conjunction with the `ip forward-protocol` command, the `ip helper-address` feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign helper address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the `ip forward-protocol` command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the `ip helper-address` feature:

1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the `ip forward-protocol` command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.168.4.251 for the 192.168.4.248/30 subnet).

**Usage Examples**

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.168.5.99:

```
ProCurve(config)#ip forward-protocol udp domain
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip helper-address 192.168.5.99
```
**ip igmp**

Use the `ip igmp` command to configure multicasting-related functions for the interface.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>immediate-leave</strong></td>
<td>Specifies that if only one host (or IGMP snooping switch) is connected to the interface, when a leave is received, multicast of that group is immediately terminated as opposed to sending a group query and timing out the group if no device responds. Works in conjunction with <code>ip igmp last-member-query-interval</code>. Applies to all groups when configured.</td>
</tr>
<tr>
<td><strong>last-member-query-interval</strong> <code>&lt;milliseconds&gt;</code></td>
<td>Controls the timeout used to detect whether any group receivers remain on an interface after a receiver leaves a group. If a receiver sends a leave-group message (IGMP Version 2), the router sends a group-specific query on that interface. After twice the time specified by this command plus as much as one second longer, if no receiver responds, the router removes that interface from the group and stops sending that group's multicast packets to the interface. Range: 100 to 65,535 ms. Default: 1000 ms.</td>
</tr>
<tr>
<td><strong>querier-timeout <code>&lt;seconds&gt;</code></strong></td>
<td>Specifies the number of seconds that the router waits after the current querier's last query before it takes over as querier (IGMP V2). Range: 60 to 300 seconds. Default: 2x the <code>query-interval</code> value.</td>
</tr>
<tr>
<td><strong>query-interval <code>&lt;seconds&gt;</code></strong></td>
<td>Specifies the interval at which IGMP queries are sent on an interface. Host query messages are addressed to the all-hosts multicast group with an IP TTL of 1. The router uses queries to detect whether multicast group members are on the interface and to select an IGMP designated router for the attached segment (if more than one multicast router exists). Only the designated router for the segment sends queries. For IGMP V2, the designated router is the router with the lowest IP address on the segment. Range: 0 to 65,535 seconds. Default: 60 seconds.</td>
</tr>
<tr>
<td><strong>query-max-response-time <code>&lt;seconds&gt;</code></strong></td>
<td>Specifies the maximum response time advertised by this interface in queries when using IGMP V2. Hosts are allowed a random time within this period to respond, reducing response bursts. Default: 10 seconds.</td>
</tr>
<tr>
<td><strong>static-group <code>&lt;group-address&gt;</code></strong></td>
<td>Configures the router's interface to be a statically-connected member of the specified group. Packets received on the correct RPF interface are forwarded to this interface regardless of whether any receivers have joined the specified group using IGMP.</td>
</tr>
<tr>
<td>**version [1</td>
<td>2]**</td>
</tr>
</tbody>
</table>

### Usage Examples

The following example sets the query message interval on the interface to 200 milliseconds:

```
config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip igmp last-member-query-interval 200
```
ip mcast-stub downstream

Use the ip mcast-stub downstream command to enable multicast forwarding and IGMP (router mode) on an interface, and to place it in multicast stub downstream mode. Use the no form of this command to disable.

Syntax Description
No subcommands.

Default Values
By default, this command is disabled.

Functional Notes
This command is used in IP multicast stub applications in conjunction with the ip mcast-stub helper-address and ip mcast-stub upstream commands. Downstream interfaces connect to segments with multicast hosts. Multiple interfaces may be configured in downstream mode; however, interfaces connecting to the multicast network (upstream) should not be configured in downstream mode. Interfaces configured as downstream should have the lowest IP address of all IGMP-capable routers on the connected segment in order to be selected as the designated router and ensure proper forwarding. Refer to ip mcast-stub helper-address <ip address> on page 397 and ip mcast-stub upstream on page 1084 for more information.

Usage Examples
The following example enables multicast forwarding and IGMP on the interface:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip mcast-stub downstream
ip mcast-stub fixed

Use the `ip mcast-stub fixed` command to allow forwarding of multicast traffic on a selected interface after enabling multicast routing. Use the `no` form of this command to disable this mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Multicast routing must be enabled prior to setting `ip mcast-stub fixed` on the selected interface. Also, use the `ip igmp static-group <A.B.C.D>` command to receive multicast traffic without host-initiated Internet Group Management Protocol (IGMP) activity on the selected interface. Otherwise, all host-initiated IGMP transactions will enter multicast routes on the router’s interface involved with IGMP activities.

**Usage Examples**

The following example enables multicast traffic forwarding and IGMP on the interface:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip mcast-stub fixed
ip mcast-stub helper-enable

Use the `ip mcast-stub helper-enable` command to assign the `ip mcast-stub helper-address` as the IGMP proxy. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address`, `ip mcast-stub upstream`, and `ip mcast-stub downstream` commands. When enabled, the interface becomes a helper forwarding interface. The IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the unit to perform as an IGMP proxy. Refer to `ip mcast-stub helper-address <ip address>` on page 397, `ip mcast-stub downstream` on page 1081, and `ip mcast-stub upstream` on page 1084 for more information.

**Usage Examples**

The following example sets the helper address as the IGMP proxy:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip mcast-stub helper-enable
**ip mcast-stub upstream**

Use the `ip mcast-stub upstream` command to enable multicast forwarding on an interface and place it in multicast stub upstream mode. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

This command is used in IP multicast stub applications in conjunction with the `ip mcast-stub helper-address` and `ip mcast-stub downstream` commands. When enabled, the interface becomes a candidate to be a helper forwarding interface. If chosen as the best path toward the helper address by the router's unicast route table, the IGMP host function is dynamically enabled and the interface becomes the active upstream interface, enabling the router to perform as an IGMP proxy. Though multiple interfaces may be candidates, no more than one interface will actively serve as the helper forwarding interface. Refer to `ip mcast-stub helper-address <ip address>` on page 397 and `ip mcast-stub downstream` on page 1081 for more information.

**Usage Examples**

The following example enables multicast forwarding on the interface:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip mcast-stub upstream
```
**ip ospf**

Use the **ip ospf** command to customize OSPF settings (if needed).

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>authentication-key</strong></td>
<td>Specifies a simple-text authentication password to be used by other routers using <code>&lt;password&gt;</code> the OSPF simple password authentication.</td>
</tr>
<tr>
<td><strong>cost <code>&lt;value&gt;</code></strong></td>
<td>Specifies the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1 to 65,535.</td>
</tr>
<tr>
<td><strong>dead-interval <code>&lt;seconds&gt;</code></strong></td>
<td>Sets the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>hello-interval <code>&lt;seconds&gt;</code></strong></td>
<td>Specifies the interval between hello packets sent on the interface. Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>message-digest-key <code>&lt;keyid&gt;</code> md5 <code>&lt;key&gt;</code></strong></td>
<td>Configures OSPF Message Digest 5 (MD5) authentication (16-byte max) keys. The SROS allows two keys (key ID 1 and key ID 2).</td>
</tr>
<tr>
<td><strong>priority <code>&lt;value&gt;</code></strong></td>
<td>Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0 to 255.</td>
</tr>
<tr>
<td><strong>retransmit-interval <code>&lt;seconds&gt;</code></strong></td>
<td>Specifies the time between link-state advertisements (LSAs). Range: 0 to 32,767.</td>
</tr>
<tr>
<td><strong>transmit-delay <code>&lt;seconds&gt;</code></strong></td>
<td>Sets the estimated time required to send an LSA on the interface. Range: 0 to 32,767.</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>retransmit-interval <code>&lt;seconds&gt;</code></td>
<td>5 seconds</td>
</tr>
<tr>
<td>transmit-delay <code>&lt;seconds&gt;</code></td>
<td>1 second</td>
</tr>
<tr>
<td>hello-interval <code>&lt;seconds&gt;</code></td>
<td>10 seconds</td>
</tr>
<tr>
<td>dead-interval <code>&lt;seconds&gt;</code></td>
<td>40 seconds</td>
</tr>
</tbody>
</table>

**Usage Example**

The following example sets the maximum number of seconds allowed between hello packets to 25,000:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip ospf dead-interval 25000
```
ip ospf authentication [message-digest | null]

Use the `ip ospf authentication` command to authenticate an interface that is performing OSPF authentication.

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-digest</td>
<td>Optional. Selects message-digest authentication type.</td>
</tr>
<tr>
<td>null</td>
<td>Optional. Specifies that no authentication is used.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this is set to null (meaning no authentication is used).

**Usage Examples**

The following example specifies that no authentication will be used on the tunnel interface:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip ospf authentication null
```
ip ospf network [broadcast | point-to-point]

Use the `ip ospf network` command to specify the type of network on this interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcast</td>
<td>Sets the network type for broadcast.</td>
</tr>
<tr>
<td>point-to-point</td>
<td>Sets the network type for point-to-point.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, Ethernet defaults to broadcast. All other interfaces default to point-to-point.

**Functional Notes**

A point-to-point network will not elect designated routers.

**Usage Examples**

The following example designates a broadcast network type:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip ospf network broadcast
```
**ip pim sparse-mode**

Use the `ip pim sparse-mode` command to enable PIM Sparse Mode on the interface. Use the `no` form of this command to disable PIM Sparse Mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, PIM Sparse Mode is disabled for all interfaces.

**Functional Notes**

PIM Sparse Mode is a multicast routing protocol that makes use of the unicast forwarding table. PIM-systems builds unidirectional shared trees rooted at a Rendezvous Point (RP) for a multicast group or a shortest-path tree rooted at a specific source for a multicast group.

**Usage Examples**

The following example enables PIM sparse-mode on the tunnel interface 1:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip pim sparse-mode
```
**ip pim-sparse dr-priority <value>**

Use the *ip pim-sparse dr-priority* command to specify the priority of this PIM interface for use when selecting the designated router (DR). Use the no form of this command to return to the default value.

**Syntax Description**

| **<value>** | Specifies the priority of this interface (to be used when determining the DR). Valid range is 1 to 4,294,967,295. |

**Default Values**

By default, the priority of all PIM interfaces is 1.

**Functional Notes**

Interfaces advertise their configured priority values in the hello messages transmitted on the interface. Routers use the priority values to determine the appropriate DR. The router on the network segment with the highest priority is selected as the DR. If a hello message is received on the interface from a router on the network segment and it does not contain a priority, the entire network segment defaults to DR selection based on IP addresses instead of priority. In this instance, the DR is selected as the router on the network segment that has the highest IP address. ProCurve Secure Routers will always include a priority in all transmitted hello messages. If no priority is specifically designated by the user, the priority is set as the default of 1.

**Usage Examples**

The following example specifies a priority of 100 on the tunnel interface 1:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip pim-sparse dr-priority 100
```
**ip pim-sparse hello-timer <time>**

Use the `ip pim-sparse hello-timer` command to set the time interval at which periodic hello messages are transmitted out the interface. Each PIM interface has an independent hello-timer. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the interval (in seconds) at which periodic hellos are sent out the interface. Valid range is 10 to 3600 seconds. |

**Default Values**

By default, hellos are transmitted on PIM interfaces every 60 seconds.

**Functional Notes**

Hello messages are used to inform neighbors of a router's presence. Hello messages normally generate a small amount of traffic on an interface. Setting the `hello-timer` to a small interval increases the amount of hellos sent (thus increasing the amount of traffic). Set the `hello-timer` to a reasonable value, taking into consideration the bandwidth available on the interface.

**Usage Examples**

The following example specifies hellos to be sent on the tunnel interface every 3600 seconds:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-hdlc 1)#ip pim-sparse hello-timer 3600
```
**ip pim-sparse nbr-timeout <time>**

Use the `ip pim-sparse nbr-timeout` command to specify the interval the PIM interface waits before declaring that the neighbor is not present. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <time> | Specifies the time interval (in seconds) the PIM interface waits before a neighbor is considered not present. Valid range is 30 to 10,800 seconds. |

**Default Values**

By default, the `nbr-timeout` is set to 105 seconds.

**Usage Examples**

The following example specifies a wait interval of 360 seconds on the tunnel interface 1:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-hdlc 1)#ip pim-sparse nbr-timeout 360
```
ip pim-sparse override-interval <time>

Use the ip pim-sparse override-interval command to specify the delay interval after a join/prune in which another router on the LAN may override the join/prune. Use the no form of this command to return to the default value.

Syntax Description

| <time> | Specifies the delay interval (in milliseconds) after a join/prune in which another router on the LAN may override the join/prune. Valid range is 0 to 65,535 milliseconds. |

Default Values

By default, the override-interval is set to 2500 milliseconds.

Usage Examples

The following example sets the delay interval for join/prune overrides to 3000 milliseconds on the tunnel interface 1:

ProCurve(config)#interface tunnel 1
ProCurve(config-hdlc 1)#ip pim-sparse override-interval 3000
**ip pim-sparse propagation-delay <time>**

Use the `ip pim-sparse propagation-delay` command to specify the expected propagation delay for join/prune messages. Set the propagation delay (in milliseconds) to estimate the amount of delay found in the local link. Use the `no` form of this command to return to the default value.

**Syntax Description**

| `<time>` | Specifies the expected propagation delay in the local link in milliseconds. Valid range is 0 to 32,767 milliseconds. |

**Default Values**

By default, the `propagation-delay` is set to 500 milliseconds.

**Usage Examples**

The following example sets the expected propagation delay to 300 milliseconds on the tunnel interface 1:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-hdlc 1)#ip pim-sparse propagation-delay 300
```
ip policy route-map <mapname>

Use the ip policy route-map command to assign a policy route map to this interface. Use the no form of this command to remove the assignment. Removing a route map from the interface does not remove the route map configuration parameters from the system.

Syntax Description

<mapname> Specifies the route map to associate with this interface.

Default Values

By default, policy-based routing is disabled for all interfaces.

Usage Examples

The following example associates the route map named MyMap with tunnel interface 1:

ProCurve(config)#interface tunnel 1
ProCurve(config-hdlc 1)#ip policy route-map MyMap
**ip proxy-arp** `<ip address> <subnet mask>`

Use the `ip proxy-arp` command to enable proxy Address Resolution Protocol (ARP) on the interface. Use the `no` form of this command to disable this feature.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;ip address&gt;</code></th>
<th>Defines the proxy ARP IP address in dotted decimal notation (for example: 192.168.73.101).</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;subnet mask&gt;</code></td>
<td>Specifies the subnet mask that corresponds to the listed IP address.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, proxy-arp is enabled.

**Functional Notes**

In general, the principle of proxy ARP allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy ARP is enabled, the SROS will respond to all proxy ARP requests with its specified MAC address and forward packets accordingly.

Enabling proxy ARP on an interface may introduce unnecessary ARP traffic on the network.

**Usage Examples**

The following enables proxy ARP on the tunnel interface:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip proxy-arp
```
**ip rip receive version [1 | 2]**

Use the `ip rip receive version` command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Only accept received RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Only accept received RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces implement RIP version 1 (the default value for the version command).

**Functional Notes**

Use the `ip rip receive version` command to specify a RIP version that overrides the `version` (in the Router RIP) configuration. See `version [1 l 2]` on page 1264 for more information.

The SROS only accepts one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the tunnel interface to accept only RIP version 2 packets:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip rip receive version 2
```
ip rip send version [1 | 2]

Use the `ip rip send version` command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the `no` form of this command to restore the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Only transmits RIP version 1 packets on the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Only transmits RIP version 2 packets on the interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, all interfaces transmit RIP version 1 (the default value for the `version` command).

**Functional Notes**

Use the `ip rip send version` command to specify a RIP version that overrides the `version` (in the Router RIP) configuration. See `version [1 l 2]` on page 1264 for more information.

The SROS only transmits one version (either 1 or 2) on a given interface.

**Usage Examples**

The following example configures the tunnel interface to transmit only RIP version 2 packets:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip rip send version 2
```
ip rip summary-address <network address> <network mask>

Use the **ip rip summary-address** command to manually summarize the routes Routing Information Protocol (RIP) will advertise and send out a specified interface. Use the **no** form of this command to disable this mode.

### Syntax Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network address&gt;</td>
<td>Specifies the IP address of the network to be summarized.</td>
</tr>
<tr>
<td>&lt;network mask&gt;</td>
<td>Specifies the network mask to be applied to the specific network to be summarized.</td>
</tr>
</tbody>
</table>

### Default Values

By default, no manual summarization is applied by RIP.

### Functional Notes

Unlike the automatic summarization on classful network boundaries, only specific network advertisements are made by RIP using the **ip rip summary-address** command. This command is only effective if RIP version 2 is configured.

### Usage Examples

The following example enables manual summarization on the specified IP address:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip rip summary-address 10.10.123.0 255.255.255.0
```
ip route-cache

Use the **ip route-cache** command to enable route caching on the interface. Use the **no** form of this command to disable route caching and return to process switching mode.

---

**Note**

Using Network Address Translation (NAT) or the SROS firewall capabilities on an interface requires process switching mode (using the **no ip route-cache** command).

---

**Syntax Description**

No subcommands.

**Default Values**

By default, route caching is enabled on all interfaces.

**Functional Notes**

Route caching allows an IP interface to provide optimum performance when processing IP traffic.

**Usage Examples**

The following example enables route caching on the tunnel interface:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#ip route-cache
```
keepalive <period> <retries>

Use the keepalive command to periodically send keepalive packets to verify the integrity of the tunnel from end to end. Use the no form of this command to disable keepalives.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;period&gt;</td>
<td>Defines the time interval (in seconds) between transmitted keepalive packets (valid range: 1 to 32,767 seconds).</td>
</tr>
<tr>
<td>&lt;retries&gt;</td>
<td>Defines the number of times to retry after failed keepalives before determining that the tunnel endpoint is down (valid range: 1 to 255 times).</td>
</tr>
</tbody>
</table>

Default Values

By default, keepalives are disabled. When enabled, the keepalive period defaults to 10 seconds and the retry count defaults to 3 times.

Functional Notes

Keepalives do not have to be configured on both ends of the tunnel in order to work. A tunnel is not aware of incoming keepalive packets.

Usage Examples

The following example enables keepalive with a period of 30 seconds and a retry count of 5 times:

ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#keepalive 30 5
lldp receive

Use the `lldp receive` command to allow LLDP packets to be received on this interface.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces are configured to send and receive LLDP packets.

**Usage Examples**

The following example configures the tunnel interface to receive LLDP packets:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#lldp receive
```
lldp send [management-address \ port-description \ system-capabilities \ system-description \ system-name]

Use the lldp send command to configure this interface to transmit LLDP packets or to control the types of information contained in the LLDP packets transmitted by this interface.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>management-address</td>
<td>Enables transmission of management address information on this interface.</td>
</tr>
<tr>
<td>port-description</td>
<td>Enables transmission of port description information on this interface.</td>
</tr>
<tr>
<td>system-capabilities</td>
<td>Enables transmission of this device’s system capabilities on this interface.</td>
</tr>
<tr>
<td>system-description</td>
<td>Enables transmission of this device’s system description on this interface.</td>
</tr>
<tr>
<td>system-name</td>
<td>Enables transmission of this device’s system name on this interface.</td>
</tr>
</tbody>
</table>

Default Values

Be default, all interfaces are configured to transmit and receive LLDP packets of all types.

Functional Notes

Individual LLDP information can be enabled or disabled using the various forms of the lldp send command. For example, use the lldp send-and-receive command to enable transmit and receive of all LLDP information. Then use the no lldp send port-description command to prevent LLDP from transmitting port description information.

Usage Examples

The following example configures the tunnel interface to transmit LLDP packets containing all enabled information types:

```plaintext
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#lldp send
```
lldp send-and-receive

Use the lldp send-and-receive command to configure this interface to transmit and receive LLDP packets.

Syntax Description

No subcommands.

Default Values

By default, all interfaces are configured to transmit and receive LLDP packets of all types.

Functional Notes

Individual LLDP information can be enabled or disabled using the various forms of the lldp send command. For example, use the lldp send-and-receive command to enable transmit and receive of all LLDP information. Then use the no lldp send port-description command to prevent LLDP from transmitting port description information.

Usage Examples

The following example configures the tunnel interface (tunnel 16) to transmit and receive LLDP packets containing all information types:

ProCurve(config)#interface tunnel 16
ProCurve(config-tunnel 16)#lldp send-and-receive
**mtu <size>**

Use the `mtu` command to configure the maximum transmit unit (MTU) size for the active interface. Use the `no` form of this command to return to the default value.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>64 to 256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>64 to 1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>64 to 18,190</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>64 to 1520</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>64 to 1500</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>The default values for the various interfaces are listed below:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Demand interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Ethernet interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>FDL interfaces</td>
<td>256</td>
</tr>
<tr>
<td>HDLC interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Loopback interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Tunnel interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual Frame Relay sub-interfaces</td>
<td>1500</td>
</tr>
<tr>
<td>Virtual PPP interfaces</td>
<td>1500</td>
</tr>
</tbody>
</table>

**Functional Notes**

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

**Usage Examples**

The following example specifies an MTU of 1200 on the tunnel interface:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#mtu 1200
```
spanning-tree path-cost <value>

Use the **spanning-tree path-cost** command to assign a cost to a bridge group that is used when computing the spanning-tree root path. To return to the default path-cost value, use the **no** form of this command.

**Syntax Description**

| <value> | Assigns a number to the bridge interface to be used as the path cost in spanning calculations (valid range: 0 to 65,535). |

**Default Values**

By default, the path-cost value is set to 19.

**Functional Notes**

The specified value is inversely proportional to the likelihood the bridge interface will be chosen as the root path. Set the path-cost value lower to increase the chance the interface will be the root. To obtain the most accurate spanning tree calculations, develop a system for determining path costs for links and apply it to all bridged interfaces.

**Usage Examples**

The following example assigns a path cost of 100 for bridge group 17 on a tunnel sub-interface:

```plaintext
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#spanning-tree path-cost 100
```

**Technology Review**

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.
tunnel checksum

Use the **tunnel checksum** command to verify the checksum of incoming Generic Routing Encapsulation (GRE) packets and to include a checksum on outgoing packets. Use the **no** form of this command to disable checksum.

### Syntax Description

No subcommands.

### Default Values

By default, **tunnel checksum** is disabled.

### Functional Notes

Both ends of the tunnel must have **tunnel checksum** enabled in order for the tunnel checksum feature to function. When both endpoints have **tunnel checksum** enabled, a packet with an incorrect checksum will be dropped. If the endpoints differ in their checksum configuration, all packets will still flow without any checksum verification.

### Usage Examples

The following example enables checksum on the tunnel 1 interface:

```procurve
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#tunnel checksum
```

### Technology Review

When enabled, the **tunnel checksum** will be calculated for each outgoing GRE packet, with the result stored in the GRE header. The checksum present bit will also be set in the header.
**tunnel destination** `<ip address>`

Use the `tunnel destination` command to specify the IP address to use as the destination address for all packets transmitted on this interface. Use the `no` form of this command to clear the `tunnel destination` address.

**Syntax Description**

`<ip address>` Specifies the IP address in dotted decimal notation to use as the destination address for all packets transmitted on this interface (for example: 192.22.73.101).

**Default Values**

By default, no tunnel destinations are defined.

**Functional Notes**

Until a tunnel interface has a destination IP address defined, it is not operational.

The tunnel destination IP address will be the value put into the destination field of the outer IP header after GRE encapsulation of the original packet. A route must be defined for the destination address. Make certain there are no recursive routes by ensuring that a tunnel's destination address will be routed out a physical interface. There is a possibility of creating a routing loop when tunnel interface traffic gets routed back to the same tunnel interface or to another tunnel interface, which in turn, does not have a route out a physical interface. In either case, the tunnel will go down for a period of one minute, after which it will come back up to determine whether the recursive routes have been resolved. This allows time for routing protocols to converge on a valid route. If a static route has caused the recursive routing loop, the tunnel status may oscillate until the route is changed.

**Usage Examples**

The following example sets the tunnel destination IP address to 192.168.73.101:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#tunnel destination 192.168.73.101
```
tunnel key <value>

Use the `tunnel key` command to specify a value shared by both endpoints of the tunnel that will provide minimal security and differentiate between tunnels with the same source and destination addresses. Use the `no` form of this command to disable the key.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;value&gt;</code></td>
<td>Defines the key value for this tunnel (valid range: 1 to 4,294,967,294).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, a key is not configured.

**Functional Notes**

When enabled, the key will be stored in the GRE header and the key present bit will be set. If tunnel keys are used, a matching key value must be defined on both endpoints of the tunnel or packets will be discarded.

**Usage Examples**

The following example sets the key on a tunnel interface to a value of 1234:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#tunnel key 1234
```
**tunnel mode gre**

Use the **tunnel mode gre** command to encapsulate traffic destined for the tunnel interface in a Generic Routing Encapsulation (GRE) header. Use the **no** form of this command to set the tunnel to its default mode.

**Syntax Description**

No subcommands.

**Default Values**

By default, the tunnel interface will be configured for GRE mode.

**Functional Notes**

GRE is currently the only allowed mode for tunnel interface operation.

**Usage Examples**

The following example configures the tunnel interface for GRE mode:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#tunnel mode gre
```
**tunnel sequence-datagrams**

Use the `tunnel sequence-datagrams` command to enable sequence number checking on incoming Generic Routing Encapsulation (GRE) packets, to drop packets arriving out of order, and to include a sequence number in outgoing packets. Use the `no` form of this command to disable sequence number checking.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Functional Notes**

Both ends of the tunnel must have sequence numbering enabled. When both endpoints have sequence numbering enabled, a packet arriving with a sequence number less than the current expected value will be dropped. If the endpoints differ in their sequence numbering configuration, all packets will still flow without any sequence number verification. Be careful enabling sequence number verification on a tunnel. The tunnel can easily become out of sequence due to network conditions outside of the tunnel endpoints. It may be difficult to establish a successful traffic flow after an out of sequence condition occurs.

**Usage Examples**

The following example enables sequence number processing on the tunnel interface:

```plaintext
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#tunnel sequence-datagrams
```

**Technology Review**

When enabled, the next valid sequence number will be placed in the GRE header of each outgoing packet, and the sequence number present bit will be set.
**tunnel source [<ip address> | <interface>]**

Use the `tunnel source` command to specify the IP address or name of a physical interface to use as the source address for all packets transmitted on this interface. Use the `no` form of this command to clear the tunnel source address.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Specifies the IP address in dotted decimal notation to use as the source address for all packets transmitted on this interface (for example: 192.168.73.101).</td>
</tr>
<tr>
<td><code>&lt;interface&gt;</code></td>
<td>Specifies the interface (in the format type <code>&lt;slot/port&gt;</code>) that contains the IP address to use as the source address for all packets transmitted on this interface.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, a tunnel source is not defined.

**Functional Notes**

Until a tunnel interface has a source IP address defined and the physical interface used as the source is operational, the tunnel is not operational.

The tunnel source IP address will be the value put into the source field of the outer IP header after GRE encapsulation of the original packet.

**Usage Examples**

The following example sets the tunnel source IP address to **192.168.73.101**:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#tunnel source 192.168.73.101
```

The following example sets the tunnel source IP address to the address of the Ethernet interface labeled 0/1:

```
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#tunnel source eth 0/1
```
ISDN GROUP CONFIG COMMAND SET

To activate the ISDN Group Configuration mode, enter the `isdn-group` command at the Global configuration command prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)# isdn-group 1
ProCurve(config-isdn-group 1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `call-type data` on page 1113
- `connect bri` on page 1114
- `incoming-accept-number <number>` on page 1115
- `max-channels <1-255>` on page 1117
- `min-channels <1-255>` on page 1118
- `resource pool-member <pool-name> [<cost>]` on page 1119
call-type data

Use the call-type data command to specify data operation for the ISDN group.

Syntax Description

data Specifies use as digital line.

Default Values

By default, the call type is set to data.

Usage Examples

The following example sets the call type for ISDN group 1 to data:

ProCurve(config)#isdn-group 1
ProCurve(config-isdn-group 1)#call-type data
connect bri

Use the **connect bri** command to associate a specific Basic Rate ISDN (BRI) interface with the ISDN group. Use the **no** form of this command to remove the specified interface from the ISDN group.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bri</td>
<td>Connects a BRI interface to the ISDN group. Use the <strong>show modules</strong> command for a list of valid BRI interfaces installed in the system.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example associates the **bri 1/1** interface with ISDN group **1**:

```
ProCurve(config)#isdn-group 1
ProCurve(config-isdn-group 1)#connect bri 1/1
```
incoming-accept-number <number>

Use the incoming-accept-number command to configure the incoming number to be accepted by this group from the PSTN. Use the no incoming-accept-number <number> command to remove a configured accept number.

Syntax Description

| <number> | Specify the phone number(s) accepted for this ISDN group. The accept number entered should match the digits that populate the Called Party Information Element received on the ISDN interface for the call. Refer to the Functional Notes for more information on entering the number. |

Default Values

By default, there are no configured incoming-accept-numbers. The ISDN group will not be able to accept calls without a configured incoming-accept-number.

Functional Notes

Special characters (parentheses, commas, and dashes) can be entered in the incoming-accept-number for readability, but they are ignored by the system. Incoming-accept-numbers are entered as a single number, or as a range of numbers using the available wildcard characters. The following wildcard inputs can be used to define numbers:

- X: Any single digit 0 through 9
- N: Any single digit 2 through 9
- [1,2,3]: Specifies single digit in this group
- $: Any number; effectively functions as a "don’t care"

The following list provides some examples for proper wildcard usage.

<table>
<thead>
<tr>
<th>Incoming Accept Number(s)</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>555-1111 and 555-1112</td>
<td>555-111[1,2]</td>
</tr>
<tr>
<td>All numbers from the 916 area code</td>
<td>916$</td>
</tr>
<tr>
<td>Numbers between 555-1000 and 555-2000</td>
<td>555-[1,2]XXX</td>
</tr>
</tbody>
</table>

Wildcard characters are especially useful in situations where ISDN hunt groups are deployed and the ISDN interfaces are all assigned to the same ISDN group in the router. ISDN hunt groups “bundle” multiple ISDN interfaces (with unique LDNs) together into a single group at the Central Office. When a call to any of the LDNs assigned to the ISDN interfaces in the hunt group is received at the Central Office, the switch sends the call to the first available ISDN interface. The ISDN group must be able to accept calls to multiple LDNs. Wildcard characters can simplify a configuration by allowing a single entry to match several numbers.
Usage Examples
The following example configures the group to accept calls for 916-555-1000 through 916-555-2000:

ProCurve(config)#isdn-group 1
ProCurve(config-isdn-group 1)#incoming-accept-number 916-555-[1,2]XXX
**max-channels <1-255>**

Use the `max-channels` command to specify the maximum number of channels allocated for the ISDN group. Use the `no` form of this command to return to the default value.

**Syntax Description**

| <1-255> | Specifies the maximum number of channels allocated for the ISDN group. Valid range is from 1 to 255. |

**Default Values**

By default, the maximum number of channels is set to 0. When `max-channels` is set to 0, the group does not limit the number of usable channels and can use all available channels. Use the `no max-channels` command to return to the default value.

**Usage Examples**

The following example sets the maximum number of channels for ISDN group 1 to 5:

```
ProCurve(config)#isdn-group 1
ProCurve(config-isdn-group 1)#max-channels 5
```
min-channels <1-255>

Use the **min-channels** command to specify the minimum number of channels allocated for the ISDN group. Use the **no** form of this command to return to the default value.

**Syntax Description**

- `<1-255>` Specifies the minimum number of channels allocated for the ISDN group. Valid range is from 1 to 255.

**Default Values**

By default, the minimum number of channels is set to 0. When **min-channels** is set to 0, no channels are reserved for this group. This group can use available channels, but does not have any channels specifically reserved. Use the **no min-channels** command to return to the default value.

**Usage Examples**

The following example sets the minimum number of channels for ISDN group 1 to 2:

```plaintext
ProCurve(config)#isdn-group 1
ProCurve(config-isdn-group 1)#min-channels 2
```
resource pool-member <pool-name> [<cost>]

Use the resource pool-member command to assign the group to a resource pool, making it a demand routing resource. Use the no form of this command to return to the default value.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool-name&gt;</td>
<td>Specifies the name of the resource pool to which this group is assigned.</td>
</tr>
<tr>
<td>&lt;cost&gt;</td>
<td>Optional. Specifies the cost of using this resource group within the specified pool.</td>
</tr>
</tbody>
</table>

In the event of a tie, a resource with a lower cost will be selected first. Interfaces with the same cost will be selected in alphabetical order by group name.

Default Values

By default, the group is not assigned to any resource pool.

Usage Examples

The following example configures the ISDN group 1 as a member of resource pool MyPool:

ProCurve(config)#isdn-group 1
ProCurve(config-isdn-group 1)#resource pool-member MyPool
CA PROFILE CONFIGURATION COMMAND SET

To activate the Certificate Authority (CA) Profile Configuration mode, enter the `crypto ca profile` command at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#crypto ca profile MyProfile
Configuring New CA Profile MyProfile
ProCurve(ca-profile)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `crl optional` on page 1121
- `email address <email address>` on page 1122
- `enrollment retry [count | period]` on page 1123
- `enrollment terminal` on page 1124
- `enrollment url <url>` on page 1125
- `fqdn <fqdn>` on page 1126
- `ip-address <address>` on page 1127
- `password <password>` on page 1128
- `serial-number` on page 1129
- `subject-name <name>` on page 1130
crl optional

Use the `crl optional` command to make CRL verification optional.

**Syntax Description**

No subcommands.

**Default Values**

By default, CRL optional is enabled.

**Functional Notes**

If enabled, the SROS is able to accept certificates even if no CRL is loaded into the configuration. Currently, this is the only mode supported by the SROS for CRL negotiations.

**Usage Examples**

The following example sets CRL verification as optional:

```
ProCurve(ca-profile)#crl optional
```
**email address** `<email address>`

Use the `email address` command to specify that an email address should be included in the certificate request.

**Syntax Description**

| `<email address>` | Specifies the complete email address to use when sending certificate requests. This field allows up to 51 characters. |

**Default Values**

No defaults necessary for this command.

**Functional Notes**

Configuring this setting simplifies the `crypto ca enroll` dialog, allowing you to enter the email address only once rather than every time you go through the enrollment process. See `crypto ca enroll <name>` on page 315.

**Usage Examples**

The following example specifies `jdoe@company.com` as the email address to be sent in certificate requests:

```
ProCurve(ca-profile)#email address jdoe@company.com
```
enrollment retry [count | period]

Use the enrollment retry command to determine how the SROS handles certificate requests.

**Syntax Description**

- **count <count>**
  Specifies the number of times the SROS re-sends a certificate request when it does not receive a response from the previous request. Range: 1 to 100.

- **period <minutes>**
  Specifies the time period between certificate request retries. The default is 1 minute between retries. Range: 1 to 60 minutes.

**Default Values**

By default, period is set to 5 minutes, and count is set to 12 retries.

**Usage Examples**

The following example configures the SROS to send certificate requests every two minutes, stopping after 50 retries (if no response is received):

```
ProCurve(ca-profile)#enrollment retry count 50
ProCurve(ca-profile)#enrollment retry period 2
```
enrollment terminal

Use the enrollment terminal command to specify manual (i.e., cut-and-paste) certificate enrollment.

Syntax Description

No subcommands.

Default Values

By default, this command is enabled.

Functional Notes

This mode is overridden if the enrollment url command specifies the CA to which automatic certificate requests are to be sent via simple certificate exchange protocol (SCEP). Issuing an enrollment terminal command after using the enrollment url command deletes the URL and forces the unit to use manual enrollment. See enrollment url <url> on page 1125 for more information.

Usage Examples

The following example configures the Secure Router OS to accept manual certificate enrollment input:

ProCurve(ca-profile)#enrollment terminal
enrollment url <url>

Use the enrollment url command to specify the URL of the CA to which the Secure Router OS should send certificate requests.

Syntax Description

| <url> | Enter the certificate authority’s URL (e.g., http://10.10.10.1:400/abcdefg/pkiclient.exe). |

Default Values

No defaults necessary for this command.

Functional Notes

When entering the URL http:// is required, followed by the IP address or DNS name of the CA. If the port number is something other than 80, include it after the IP address or DNS name separated with a colon (:).

The CA may have other necessary information to include in the CGI path before ending with the actual CGI program. An example template to follow is http://hostname:port/path/to/program.exe.

NOTE: To use the default program pkiclient.exe without specifying it, end the URL with a slash (/). Otherwise, you must enter the program name to use. For example, http://10.10.10.1:400/abcdefg/ will assume pkiclient.exe as the program (but not including the terminating slash is a configuration error).

Specifying this command will override the enrollment terminal setting as described previously (see enrollment terminal on page 1124).

Usage Examples

The following example specifies http://CAserver/certsrv/mscep/mscep.dll as the URL to which the Secure Router OS will send certificate requests:

ProCurve(ca-profile)# enrollment url http://CAserver/certsrv/mscep/mscep.dll
**fqdn <fqdn>**

Use the **fqdn** command to specify a fully-qualified domain name (FQDN) to be included in the certificate requests.

**Syntax Description**

| <fqdn> | Specifies the FQDN (e.g., company.com) to be included in requests. |

**Default Values**

No defaults necessary for this command.

**Functional Notes**

Configuring this setting simplifies the **crypto ca enroll** dialog, allowing you to enter the FQDN only once rather than every time you go through the enrollment process. See **crypto ca enroll <name>** on page 315.

**Usage Examples**

The following example specifies **company.com** as the FQDN to be sent in certificate requests:

ProCurve(ca-profile)#fqdn company.com
ip-address <address>

Use the ip-address command to specify an IP address to be included in the certificate requests.

**Syntax Description**

<address> Defines the IP address in dotted decimal notation (e.g., 192.22.73.101).

**Default Values**

No defaults necessary for this command.

**Functional Notes**

Configuring this setting simplifies the crypto ca enroll dialog, allowing you to enter the IP address only once rather than every time you go through the enrollment process. Refer to crypto ca enroll <name> on page 315.

**Usage Examples**

The following example specifies 66.203.52.193 as the IP address to be sent in certificate requests:

ProCurve(ca-profile)#ip-address 66.203.52.193
password <password>

Use the `password` command to specify the challenge password for simple certificate exchange protocol (SCEP). Use the `no` form of this command to allow CA requests to be sent automatically (using SCEP) without requiring a password.

**Syntax Description**

<password> Specifies the SCEP password (up to 80 characters).

**Default Values**

By default, no password is required.

**Functional Notes**

There are two places for configuring a SCEP password:

- At the `(ca-profile)#` prompt.
- If it is not configured at the `(ca-profile)#` prompt, you are prompted to enter one when going through the certificate enrollment process.

The password is sent to the CA from which you are requesting a certificate. The CA may then ask for the password later before a certificate can be revoked. See `crypto ca enroll <name>` on page 315.

**Usage Examples**

The following example sets the SCEP challenge password to `procurve`:

ProCurve(ca-profile)# password procurve
serial-number

Use the `serial-number` command to specify that a serial number will be included in the certificate request.

**Syntax Description**
No subcommands.

**Default Values**
By default, this command is disabled.

**Functional Notes**
By default, this command is set to `no serial-number`, which means that the serial number is not included in the certificate requests.

**Usage Examples**
The following example configures Secure Router OS to include a serial number in the certificate request:

ProCurve(ca-profile)#`serial-number`
subject-name <name>

Use the subject-name command to specify the subject name used in the certificate request.

**Syntax Description**

| <name> | Specifies a subject name string with up to 256 characters entered in X.500 LDAP format. |

**Default Values**

By default, there is no subject name configured.

**Functional Notes**

Configuring this setting simplifies the crypto ca enroll dialog, allowing you to enter the subject name only once rather than every time you go through the enrollment process. Refer to crypto ca enroll <name> on page 315.

**Usage Examples**

The following example assigns a subject name of cert to certificate requests:

ProCurve(ca-profile)#subject-name cert
CERTIFICATE CONFIGURATION COMMAND SET

To activate the Certificate Configuration mode, enter the `crypto ca certificate chain` command at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# crypto ca certificate chain MyProfile
ProCurve(config-cert-chain)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `certificate <serial-number>` on page 1132
- `certificate ca <serial-number>` on page 1133
- `crl` on page 1134
**certificate <serial-number>**

Use the `certificate` command to restore a certificate. Use the `no` form of this command to remove a specific certificate from the certificate chain.

**Syntax Description**

<table>
<thead>
<tr>
<th><code>&lt;serial-number&gt;</code></th>
<th>Specifies the certificate's serial number (up to 51 characters). This value can be found for existing certificates by using the <code>show run</code> command.</th>
</tr>
</thead>
</table>

**Default Values**

No defaults necessary for this command.

**Functional Notes**

The user typically does not enter this command. It is primarily used to restore certificates from the startup configuration when the product is powered up.

**Usage Examples**

The following example removes the certificate with the serial number 73f0bfe5ed8391a54d1214390a36cee7:

```plaintext
ProCurve(config)#crypto ca certificate chain MyProfile
ProCurve(config-cert-chain)#no certificate 73f0bfe5ed8391a54d1214390a36cee7
```
certificate ca <serial-number>

Use the `certificate ca` command to restore a CA certificate. Use the `no` form of this command to remove a specific certificate from the certificate chain for a CA.

**Syntax Description**

- `<serial-number>` Specifies the certificate's serial number (up to 51 characters). This value can be found for existing certificates by using the `show run` command.

**Default Values**

No defaults necessary for this command.

**Functional Notes**

The user typically does not enter this command. It is primarily used to restore certificates from the startup configuration when the product is powered up.

**Usage Examples**

The following example removes the CA certificate with the serial number 0712:

```
ProCurve(config)#crypto ca certificate chain MyProfile
ProCurve(config-cert-chain)#no certificate ca 0712
```
crl

Use the crl command to restore a CRL. Use the no form of this command to remove the CRL for the specific CA.

Syntax Description
No subcommands.

Default Values
No defaults necessary for this command.

Functional Notes
The user typically does not enter this command. It is primarily used to restore CRLs from the startup configuration when the product is powered up.

Usage Examples
The following example removes the CRL for the current CA:

ProCurve(config)#crypto ca certificate chain MyProfile
ProCurve(config-cert-chain)#no crl
CRYPTO MAP IKE COMMAND SET

To activate the Crypto Map IKE mode, enter a valid version of the `crypto map ipsec-ike` command at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#crypto map Map-Name 10 ipsec-ike
ProCurve(config-crypto-map)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `antireplay` on page 1136
- `ike-policy <policy number>` on page 1139
- `match address <listname>` on page 1140
- `set peer <address>` on page 1142
- `set pfs [group1 | group2]` on page 1143
- `set security-association lifetime [kilobytes | seconds] <value>` on page 1144
- `set transform-set <setname1 - setname6>` on page 1145
antireplay

Use the antireplay command to enable antireplay sequence number checking for all security associations created on this crypto map. Use the no form of this command to disable.

Syntax Description
No subcommands.

Default Values
By default, this command is enabled.

Usage Examples
The following example enables antireplay sequence checking on crypto map VPN 100:

ProCurve(config)#crypto map VPN 100 ipsec-ike
ProCurve(config-crypto-map)#antireplay
commit-bit

Use the commit-bit command to set the commit-bit in the Internet Security Association and Key Management Protocol (ISAKMP) header when sending the second message of quick mode on an IPSec tunnel negotiation. Use the no form of this command to disable this feature.

Syntax Description

No subcommands.

Default Values

By default, the commit-bit will be used.

Functional Notes

As an extra security measure, the commit-bit can be set by the responder of a quick mode negotiation to force the initiator to wait for the fourth message of quick mode before bringing up its IPSec security associations (SA's). By default, this feature is enabled on all SROS routers. Some vendors, however, may have incorrect implementations of the commit-bit that do not interoperate well with SROS routers. In that case, the commit-bit should be disabled on all crypto maps that have a peer which does not support the commit-bit.

Usage Example

The following example disables the use of commit-bit:

ProCurve(config-crypto-map)#no commit-bit

The following example displays a configuration with the commit-bit disabled:

```
ip crypto
!  
crypto ike policy 100
    initiate main
    respond main
    local-id address 10.10.10.1
    peer 192.168.1.1
    attribute 2
    encryption aes-256-cbc
    authentication pre-share
    lifetime 3600
!
crypto ike remote-id address 10.10.10.1 preshared-key procurve ike-policy 100 crypto map VPN 10 no-mode-config no-xauth
```
crypto ipsec transform-set esp-aes-256-cbc-esp-sha-hmac esp-aes-256-cbc esp-sha-hmac
  mode tunnel

crypto map VPN 10 ipsec-ike
  description VPN to Main Site
  match address VPN-10-vpn-selectors
  set peer 192.168.1.1
  set transform-set esp-aes-256-cbc-esp-sha-hmac
  set security-association lifetime seconds 3600
  no commit-bit

ike-policy 100
ike-policy <policy number>

Use the ike-policy command to ensure that only a specified IKE policy is used to establish the IPSec tunnel. This prevents any mobile VPN policies from using IPSec policies that are configured for static VPN peer policies.

**Syntax Description**

| **<policy number>** | Specifies the policy number of the policy to assign to this crypto map. |

**Default Values**

No defaults necessary for this command.

**Usage Examples**

The following example shows a typical crypto map configuration:

ProCurve(config)#crypto ike policy 100
ProCurve(config)#crypto map VPN 10 ipsec-ike
ProCurve(config-crypto-map)#description "Remote Office"
ProCurve(config-crypto-map)#match address VPN-10-vpn-selectors
ProCurve(config-crypto-map)#set peer 10.22.17.13
ProCurve(config-crypto-map)#set transform-set esp-3des-esp-md5-hmac
ProCurve(config-crypto-map)#ike-policy 100
match address <listname>

Use the match address command to assign an IP access list to a crypto map definition. The access list designates the IP packets to be encrypted by this crypto map. See ip access-list extended <listname> on page 350 for more information on creating access lists.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listname&gt;</td>
<td>Enter the name of the access list you wish to assign to this crypto map.</td>
</tr>
</tbody>
</table>

Default Values

By default, no IP access lists are defined.

Functional Notes

Crypto map entries do not directly contain the selectors used to determine which data to secure. Instead, the crypto map entry refers to an access control list. An access control list (ACL) is assigned to the crypto map using the match address command. If no ACL is configured for a crypto map, then the entry is incomplete and will have no effect on the system.

The entries of the ACL used in a crypto map should be created with respect to traffic sent by the product. The source information must be the local product and the destination must be the peer.

Only extended access lists can be used in crypto maps.

Usage Examples

The following example shows setting up an ACL (called NewList) and then assigning the new list to a crypto map (called NewMap):

ProCurve(config)#ip access-list extended NewList

Configuring New Extended ACL "NewList"

ProCurve(config-ext-nacl)#exit
ProCurve(config)#crypto map NewMap 10 ipsec-ike
ProCurve(config-crypto-map)#match address NewList
A crypto map entry is a single policy that describes how certain traffic is to be secured. There are two types of crypto map entries: **ipsec-manual** and **ipsec-ike**. Each entry is given an index, which is used to sort the ordered list.

When an unsecure packet arrives on an interface, the crypto map set associated with that interface is processed in order. If a crypto map entry matches the unsecure traffic, the traffic is discarded.

When a packet is to be transmitted on an interface, the crypto map set associated with that interface is processed in order. The first crypto map entry that matches the packet will be used to secure the packet. If a suitable SA exists, that is used for transmission. Otherwise, IKE is used to establish an SA with the peer. If no SA exists, and the crypto map entry is “respond only,” the packet is discarded.

When a secured packet arrives on an interface, its SPI is used to look up an SA. If an SA does not exist, or if the packet fails any of the security checks (bad authentication, traffic does not match SA selectors, etc.), it is discarded. If all checks pass, the packet is forwarded normally.
set peer <address>

Use the set peer command to set the IP address of the peer device.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Specifies the IP address of the peer device. If this is not configured, it implies responder only to any peer.</td>
</tr>
</tbody>
</table>

Default Values

There are no default settings for this command.

Functional Notes

If no peer IP addresses are configured, the entry will only be used to respond to IPSec requests; it cannot initiate the requests (since it doesn't know which IP address to send the packet to). If a single peer IP address is configured, the crypto map entry can be used to both initiate and respond to SAs.

The peer IP address is the public IP address of the device which will terminate the IPSec tunnel. If the peer IP address is not static, the product cannot initiate the VPN tunnel. By setting no peer IP address, the product can respond to an IPSec tunnel request.

Usage Examples

The following example sets the peer IP address of 10.100.23.64:

ProCurve(config-crypto-map)#set peer 10.100.23.64
**set pfs [group1 | group2]**

Use the `set pfs` command to choose the type of perfect forward secrecy (if any) that will be required during IPSec negotiation of security associations for this crypto map. Use the `no` form of this command to require no PFS.

### Syntax Description

<table>
<thead>
<tr>
<th>group1</th>
<th>Requires IPSec to use Diffie-Hellman Group 1 (768-bit modulus) exchange during IPSec SA key generation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>group2</td>
<td>Requires IPSec to use Diffie-Hellman Group 2 (1024-bit modulus) exchange during IPSec SA key generation.</td>
</tr>
</tbody>
</table>

### Default Values

By default, no PFS will be used during IPSec SA key generation.

### Functional Notes

If left at the default setting, no perfect forward secrecy (PFS) will be used during IPSec SA key generation. If PFS is specified, then the specified Diffie-Hellman Group exchange will be used for the initial and all subsequent key generation, thus providing no data linkage between prior keys and future keys.

### Usage Examples

The following example specifies use of the Diffie-Hellman Group 1 exchange during IPSec SA key generation:

```
ProCurve(config-crypto-map)#set pfs group 1
```
set security-association lifetime [kilobytes | seconds] <value>

Use the `set security-association lifetime` command to define the lifetime (in kilobytes and/or seconds) of the IPSec SAs created by this crypto map.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>kilobytes &lt;value&gt;</td>
<td>Specifies the SA lifetime limit in kilobytes.</td>
</tr>
<tr>
<td>seconds &lt;value&gt;</td>
<td>Specifies the SA lifetime limit in seconds.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the security-association lifetime is set to 28,800 seconds and there is no default for the kilobytes lifetime.

**Functional Notes**

Values can be entered for this command in both kilobytes and seconds. Whichever limit is reached first will end the security association.

**Usage Examples**

The following example sets the SA lifetime to 300 kilobytes and 2 hours:

ProCurve(config-crypto-map)# set security-association lifetime kilobytes 300
ProCurve(config-crypto-map)# set security-association lifetime seconds 7200
**set transform-set <setname1 - setname6>**

Use the `set transform-set` command to assign up to six transform sets to a crypto map. Use the `no` form of this command to remove an assigned transform set. Refer to `crypto ipsec transform-set <setname> <parameters>` on page 325 for information on defining transform sets.

**Syntax Description**

| <setname> | Assign up to six transform-sets to this crypto map by listing the set names, separated by a space. |

**Default Values**

By default, there is no transform set assigned to the crypto map.

**Functional Notes**

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms.

If no transform set is configured for a crypto map, then the entry is incomplete and will have no effect on the system.

**Usage Examples**

The following example first creates a transform set (`Set1`) consisting of two security algorithms (up to three may be defined), and then assigns the transform set to a crypto map (`Map1`):

```
ProCurve(config)#crypto ipsec transform-set Set1 esp-3des esp-sha-hmac
ProCurve(cfg-crypto-trans)#exit
ProCurve(config)#crypto map Map1 1 ipsec-ike
ProCurve(config-crypto-map)#set transform-set Set1
```
To activate the Crypto Map Manual mode, enter a valid version of the `crypto map ipsec-manual` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#crypto map Map-Name 10 ipsec-manual
ProCurve(config-crypto-map)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `antireplay` on page 1147
- `ike-policy <policy number>` on page 1148
- `match address <listname>` on page 1149
- `set peer <address>` on page 1151
- `set session-key [inbound | outbound]` on page 1152
- `set transform-set <setname>` on page 1156
antireplay

Use the `antireplay` command to enable antireplay sequence number checking for all security associations created on this crypto map. Use the `no` form of this command to disable.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is enabled.

**Usage Examples**

The following example enables antireplay sequence checking on crypto map VPN 100:

```
ProCurve(config)#crypto map VPN 100 ipsec-manual
ProCurve(config-crypto-map)#antireplay
```
ike-policy <policy number>

Use the **ike-policy** command to ensure that only a specified IKE policy is used to establish the IPSec tunnel. This prevents any mobile VPN policies from using IPSec policies that are configured for static VPN peer policies.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy number&gt;</td>
<td>Specifies the policy number of the policy to assign to this crypto map.</td>
</tr>
</tbody>
</table>

**Default Values**

No defaults necessary for this command.

**Usage Examples**

The following example shows a typical crypto map configuration:

ProCurve(config)#crypto ike policy 100
ProCurve(config)#crypto map VPN 10 ipsec-manual
ProCurve(config-crypto-map)#description "Remote Office"
ProCurve(config-crypto-map)#match address VPN-10-vpn-selectors
ProCurve(config-crypto-map)#set peer 10.22.17.13
ProCurve(config-crypto-map)#set transform-set esp-3des-esp-md5-hmac
ProCurve(config-crypto-map)#ike-policy 100
match address <listname>

Use the match address command to assign an IP access list to a crypto map definition. The access list designates the IP packets to be encrypted by this crypto map. Use the no form of this command to remove a defined IP access list. See ip access-list extended <listname> on page 350 for more information on creating access lists.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listname&gt;</td>
<td>Specifies the name of the access list you wish to assign to this crypto map.</td>
</tr>
</tbody>
</table>

Default Values

By default, no IP access lists are defined.

Functional Notes

Crypto map entries do not directly contain the selectors used to determine which data to secure. Instead, the crypto map entry refers to an access control list. An access control list (ACL) is assigned to the crypto map using the match address command (see crypto map on page 326). If no ACL is configured for a crypto map, then the entry is incomplete and will have no effect on the system.

The entries of the ACL used in a crypto map should be created with respect to traffic sent by the product. The source information must be the local product, and the destination must be the peer.

Only extended access lists can be used in crypto maps.

Usage Examples

The following example shows setting up an access list (called NewList) and then assigning the new list to a crypto map (called NewMap):

ProCurve(config)#ip access-list extended NewList

Configuring New Extended ACL "NewList"

ProCurve(config-ext-nacl)#exit
ProCurve(config)#crypto map NewMap 10 ipsec-manual
ProCurve(config-crypto-map)#match address NewList
A crypto map entry is a single policy that describes how certain traffic is to be secured. There are two types of crypto map entries: **ipsec-manual** and **ipsec-ike**. Each entry is given an index, which is used to sort the ordered list.

When an unsecured packet arrives on an interface, the crypto map set associated with that interface is processed in order. If a crypto map entry matches the unsecured traffic, the traffic is discarded.

When a packet is to be transmitted on an interface, the crypto map set associated with that interface is processed in order. The first crypto map entry that matches the packet will be used to secure the packet. If a suitable SA exists, that is used for transmission. Otherwise, IKE is used to establish an SA with the peer. If no SA exists, and the crypto map entry is “respond only,” the packet is discarded.

When a secured packet arrives on an interface, its SPI is used to look up an SA. If an SA does not exist, or if the packet fails any of the security checks (bad authentication, traffic does not match SA selectors, etc.), it is discarded. If all checks pass, the packet is forwarded normally.
set peer <address>

Use the set peer command to set the IP address of the peer device.

**Syntax Description**

- `<address>` Specifies the IP address of the peer device.

**Default Values**

There are no default settings for this command.

**Functional Notes**

If no peer IP address is configured, the manual crypto map is not valid and not complete. A peer IP address is required for manual crypto maps. To change the peer IP address, the `no set peer` command must be issued first; then the new peer IP address can be configured.

**Usage Examples**

The following example sets the peer IP address of 10.100.23.64:

ProCurve(config-crypto-map)#set peer 10.100.23.64
set session-key [inbound | outbound]

Use the set session-key command to define the encryption and authentication keys for this crypto map. Variations of this command include the following:

- set session-key inbound ah <SPI> <keyvalue>
- set session-key inbound esp <SPI> authenticator <keyvalue>
- set session-key inbound esp <SPI> cipher <keyvalue>
- set session-key inbound esp <SPI> cipher <keyvalue> authenticator <keyvalue>
- set session-key outbound ah <SPI> <keyvalue>
- set session-key outbound esp <SPI> authenticator <keyvalue>
- set session-key outbound esp <SPI> cipher <keyvalue>
- set session-key outbound esp <SPI> cipher <keyvalue> authenticator <keyvalue>

Syntax Description

- inbound: Defines encryption keys for inbound traffic.
- outbound: Defines encryption keys for outbound traffic.
- ah <SPI>: Specifies authentication header protocol.
- esp <SPI>: Specifies encapsulating security payload protocol.
- cipher <keyvalue>: Specifies encryption/decryption key.
- authenticator <keyvalue>: Specifies authentication key.

Default Values

There are no default settings for this command.

Functional Notes

The inbound local security parameter index (SPI) must equal the outbound remote SPI. The outbound local SPI must equal the inbound remote SPI. The key values are the hexadecimal representations of the keys. They are not true ASCII strings. Therefore, a key of 3031323334353637 represents “01234567”.

See the following table for key length requirements.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Minimum key length required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>64-bits in length; 8 hexadecimal bytes</td>
</tr>
<tr>
<td>3DES</td>
<td>192-bits in length; 24 hexadecimal bytes</td>
</tr>
<tr>
<td>AES-128-CBC</td>
<td>128-bits in length; 16 hexadecimal bytes</td>
</tr>
<tr>
<td>AES-192-CBC</td>
<td>192-bits in length; 24 hexadecimal bytes</td>
</tr>
<tr>
<td>AES-256-CBC</td>
<td>256-bits in length; 32 hexadecimal bytes</td>
</tr>
<tr>
<td>MD5</td>
<td>128-bits in length; 16 hexadecimal bytes</td>
</tr>
<tr>
<td>SHA1</td>
<td>160-bits in length; 20 hexadecimal bytes</td>
</tr>
</tbody>
</table>
Technology Review

The following example configures an Secure Router OS product for VPN using IPSec manual keys. This example assumes that the Secure Router OS product has been configured with a WAN IP Address of 172.27.45.57 on interface ppp 1 and a LAN IP Address of 10.10.10.254 on interface ethernet 0/1. The Peer Private IP Subnet is 10.10.20.0.

For more detailed information on VPN configuration, refer to the VPN Configuration Guide located on the ProCurve SROS Documentation CD provided with your unit.

Step 1:
Enter the Global Configuration mode (i.e., config terminal mode).

ProCurve>enable
ProCurve#configure terminal

Step 2:
Enable VPN support using the ip crypto command. This command allows crypto maps to be applied to interfaces, and enables the IKE server to listen for IKE negotiation sessions on UDP port 500.

ProCurve(config)#ip crypto

Step 3:
Define the transform set. A transform set defines the encryption and/or authentication algorithms to be used to secure the data transmitted over the VPN tunnel. Multiple transform sets may be defined in a system. Once a transform set is defined, many different crypto maps within the system can reference it. In this example, a transform set named highly_secure has been created. This transform set defines ESP with authentication implemented using 3DES encryption and SHA1 authentication.

ProCurve(config)#crypto ipsec transform-set highly_secure esp-3des esp-sha-hmac
ProCurve(cfg-crypto-trans)#mode tunnel

Step 4:
Define an IP access list. An Extended Access Control List is used to specify which traffic needs to be sent securely over the VPN tunnel. The entries in the list are defined with respect to the local system. The source IP address will be the source of the traffic to be encrypted. The destination IP address will be the receiver of the data on the other side of the VPN tunnel.

ProCurve(config)#ip access-list extended corporate_traffic
ProCurve(config-ext-nacl)#permit ip 10.10.10.0 0.0.0.255 10.10.20.0 0.0.0.255 log
deny ip any any
Step 5:
Create crypto map and define manual keys. A Crypto Map is used to define a set of encryption schemes to be used for a given interface. A crypto map entry has a unique index within the crypto map set. The crypto map entry will specify whether IKE is used to generate encryption keys or if manually specified keys will be used. The crypto map entry will also specify who will be terminating the VPN tunnel, as well as which transform set or sets will be used to encrypt and/or authenticate the traffic on that VPN tunnel. It also specifies the lifetime of all created IPSec security associations.

The keys for the algorithms defined in the transform set associated with the crypto map will be defined by using the `set session-key` command. A separate key is needed for both inbound and outbound traffic. The key format consists of a string of hexadecimal values without the leading `0x` for each character. For example, a cipher key of `this is my cipher key` would be entered as:

```
74686973206973206D7920636970686572206B6579.
```

A unique Security Parameter Index (SPI) is needed for both inbound and outbound traffic. The local system's inbound SPI and keys will be the peer's outbound SPI and keys. The local system's outbound SPI and keys will be the peer's inbound SPI and keys. In this example the following keys and SPIs are used:

- Inbound cipher SPI: 300
  Inbound cipher key: "2te$#g89jnr(j@4rvnfhg5e"
- Outbound cipher SPI: 400
  Outbound cipher key: "8564hgjelrign*(gnb#1$d3"
- Inbound authenticator key: "r5%^ughembkdjh34$x.<"
- Outbound authenticator key: "io78*7gner#4(mgnsd!3"

```
ProCurve(config)#crypto map corporate_vpn 1 ipsec-ike
ProCurve(config-crypto-map)#match address corporate_traffic
ProCurve(config-crypto-map)#set peer 172.27.15.129
ProCurve(config-crypto-map)#set transform-set highly_secure
ProCurve(config-crypto-map)#set session-key inbound esp 300 cipher 7235255E756768656E62666466686572206B6579.
ProCurve(config-crypto-map)#set session-key outbound esp 400 cipher 696F37382A37676E65722334286D676E673642133
```

Step 6:
Configure public interface. This process includes configuring the IP address for the interface and applying the appropriate crypto map to the interface. Crypto maps are applied to the interface on which encrypted traffic will be transmitted.

```
ProCurve(config)#interface ppp 1
ProCurve(config-ppp 1)#ip address 172.27.45.57 255.255.255.248
ProCurve(config-ppp 1)#crypto map corporate_vpn
ProCurve(config-ppp 1)#no shutdown
```
Step 7:
Configure private interface to allow all traffic destined for the VPN tunnel to be routed to the appropriate gateway.

ProCurve(config)#interface ethernet 0/1
ProCurve(config-eth 0/1)#ip address 10.10.10.254 255.255.255.0
ProCurve(config-eth 0/1)#no shutdown
ProCurve(config-eth 0/1)#exit
**set transform-set** `<setname>`

Use the `set transform-set` command to assign a transform set to a crypto map. Use the `no` form of this command to remove assigned transform sets. Refer to `crypto ipsec transform-set <setname> <parameters>` on page 325 for information on defining transform sets.

**Syntax Description**

| `<setname>` | Assigns a transform set to this crypto map by entering the set name. |

**Default Values**

By default, no transform set is assigned to the crypto map.

**Functional Notes**

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms.

If no transform set is configured for a crypto map, then the entry is incomplete and will have no effect on the system. For manual key crypto maps, only one transform set can be specified.

**Usage Examples**

The following example first creates a transform set (**Set1**) consisting of two security algorithms (up to three may be defined), and then assigns the transform set to a crypto map (**Map1**):

```
ProCurve(config)#crypto ipsec transform-set Set1 esp-3des esp-sha-hmac
ProCurve(config)#exit
```

```
ProCurve(config)#crypto map Map1 1 ipsec-manual
ProCurve(config)#set transform-set Set1
```
IKE Client Command Set

To activate the IKE Client mode, enter the `crypto ike client` command at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# crypto ike client configuration pool
ProCurve(config-ike-client-pool)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `dns-server <address1> <address2>` on page 1158
- `ip-range <start ip> <end ip>` on page 1159
- `netbios-name-server <address1> <address2>` on page 1160

**Note** For VPN configuration example scripts, refer to the *VPN Configuration Guide* located on the *ProCurve SROS Documentation* CD provided with your unit.
**dns-server** `<address1> <address2>`

Use the `dns-server` command to specify the DNS server address(es) to assign to a client. Use the `no` form of this command to remove defined server address(es).

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address1&gt;</code></td>
<td>Assigns the first DNS server address.</td>
</tr>
<tr>
<td><code>&lt;address2&gt;</code></td>
<td>Optional. Assigns the second DNS server address.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no DNS server address is defined.

**Usage Examples**

The following example defines two DNS server addresses for this configuration pool:

ProCurve(config-ike-client-pool)#`dns-server 172.16.17.1 172.16.17.3`
Use the `ip-range` command to specify the range of addresses from which the router draws when assigning an IP address to a client. Use the `no` form of this command to remove defined IP ranges.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;start ip&gt;</code></td>
<td>Specifies the first IP address in the range for this pool.</td>
</tr>
<tr>
<td><code>&lt;end ip&gt;</code></td>
<td>Specifies the last IP address in the range for this pool.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no IP address range is defined.

**Usage Examples**

The following example defines an IP address range for this configuration pool:

```
ProCurve(config-ike-client-pool)#ip-range 172.16.1.1 172.16.1.25
```
netbios-name-server <address1> <address2>

Use the netbios-name-server command to specify the NetBIOS Windows Internet Naming Service (WINS) name servers to assign to a client. Use the no form of this command to remove assigned name servers.

Syntax Description

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address1&gt;</td>
</tr>
<tr>
<td>&lt;address2&gt;</td>
</tr>
</tbody>
</table>

Specifies the first WINS server address to assign.
Specifies the second WINS server address to assign.

Default Values

By default, no WINS server address is defined.

Usage Examples

The following example defines two WINS server addresses for this configuration pool:

ProCurve(config-ike-client-pool)#netbios-name-server 172.16.17.1 172.16.17.25
IKE POLICY ATTRIBUTES COMMAND SET

To activate the IKE Policy Attributes mode, enter the `attribute` command at the IKE Policy prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#crypto ike policy 1
ProCurve(config-ike)#attribute 10
ProCurve(config-ike-attribute)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `authentication [dss-sig | pre-share | rsa-sig]` on page 1162
- `encryption [aes-xxx-cbc | des | 3des]` on page 1163
- `group [1 | 2]` on page 1164
- `hash [md5 | sha]` on page 1165
- `lifetime <seconds>` on page 1166

Note: For VPN configuration example scripts, refer to the VPN Configuration Guide located on the ProCurve SROS Documentation CD provided with your unit.
authentication [dss-sig | pre-share | rsa-sig]

Use the authentication command to configure this IKE policy’s use of pre-shared secrets and signed certificates during IKE negotiation.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dss-sig</td>
<td>Specifies to use DSS-signed certificates during IKE negotiation to validate the peer.</td>
</tr>
<tr>
<td>pre-share</td>
<td>Specifies the use of pre-shared secrets during IKE negotiation to validate the peer.</td>
</tr>
<tr>
<td>rsa-sig</td>
<td>Specifies to use RSA-signed certificates during IKE negotiation to validate the peer.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is enabled.

**Functional Notes**

Both sides must share the same pre-shared secret in order for the negotiation to be successful.

**Usage Example**

The following example enables pre-shared secrets for this IKE policy:

```
ProCurve(config-ike-attribute)#authentication pre-share
```
encryption [aes-xxx-cbc | des | 3des]

Use the `encryption` command to specify which encryption algorithm this IKE policy will use to transmit data over the IKE-generated SA.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aes-128-cbc</td>
<td>Specifies the AES 128 CBC encryption algorithm.</td>
</tr>
<tr>
<td>aes-192-cbc</td>
<td>Specifies the AES 192 CBC encryption algorithm.</td>
</tr>
<tr>
<td>aes-256-cbc</td>
<td>Specifies the AES 256 CBC encryption algorithm.</td>
</tr>
<tr>
<td>des</td>
<td>Specifies the DES encryption algorithm.</td>
</tr>
<tr>
<td>3des</td>
<td>Specifies the 3DES encryption algorithm.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, encryption is set to des.

**Usage Examples**

The following example selects 3DES as the encryption algorithm for this IKE policy:

```
ProCurve(config-ike-attribute)#encryption 3des
```
Use the `group` command to specify the Diffie-Hellman Group (1 or 2) to be used by this IKE policy to generate the keys (which are then used to create the IPSec SA).

**Syntax Description**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specifies 768-bit mod P.</td>
</tr>
<tr>
<td>2</td>
<td>Specifies 1024-bit mod P.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, group is set to 1.

**Functional Notes**

The local IKE policy and the peer IKE policy must have matching group settings in order for negotiation to be successful.

**Usage Examples**

The following example sets this IKE policy to use Diffie-Hellman Group 2:

```
ProCurve(config-ike-attribute)#group 2
```
hash [md5| sha]

Use the `hash` command to specify the hash algorithm to be used to authenticate the data transmitted over the IKE SA.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>md5</td>
<td>Choose the MD5 hash algorithm.</td>
</tr>
<tr>
<td>sha</td>
<td>Choose the SHA hash algorithm.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, hash is set to `sha`.

**Usage Examples**

The following example specifies `md5` as the hash algorithm:

```
ProCurve(config-ike-attribute)#hash md5
```
**lifetime** `<seconds>`

Use the `lifetime` command to specify how long an IKE SA is valid before expiring.

**Syntax Description**

```
<seconds>  Specifies how many seconds an IKE SA will last before expiring.
```

**Default Values**

By default, `lifetime` is set to 28,800 seconds.

**Usage Examples**

The following example sets a lifetime of two hours:

```
ProCurve(config-ike-attribute)#lifetime 7200
```
IKE POLICY COMMAND SET

To activate the IKE Policy mode, enter the `crypto ike policy` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#crypto ike policy 1
ProCurve(config-ike)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `attribute <policynumber>` on page 1168
- `client authentication host` on page 1169
- `client authentication host xauth-type [generic | otp | radius]` on page 1170
- `client authentication server list <listname>` on page 1171
- `client configuration pool <poolname>` on page 1172
- `initiate [aggressive | main]` on page 1173
- `local-id [address | asn1-dn | fqdn | user-fqdn] <ipaddress or name>` on page 1174
- `nat-traversal <version> [allow | disable | force]` on page 1175
- `peer [<ip address> | any]` on page 1176
- `respond [aggressive | anymode | main]` on page 1177

**Note** For VPN configuration example scripts, refer to the *VPN Configuration Guide* located on the ProCurve SROS Documentation CD provided with your unit.
attribute <policynumber>

Use the **attribute** command to define attributes for the associated IKE policy. Multiple attributes can be created for a single IKE policy. Once you enter this command, you are in the IKE Policy Attribute. Refer to [*IKE Policy Attributes Command Set*](#) on page 1161 for more information.

**Syntax Description**

| <policynumber> | Assigns a number (range: 1 to 65,535) to the attribute policy. The number is the attribute's priority number and specifies the order in which the resulting VPN proposals get sent to the far end.

This command takes you to the *(config-ike-attribute)#* prompt. From here, you can configure the settings for the attribute as outlined in the section [*IKE Policy Attributes Command Set*](#) on page 1161.

**Default Values**

By default, no attribute is defined.

**Functional Notes**

Multiple attributes on an IKE policy are ordered by number (with the lowest number representing the highest priority).

**Usage Examples**

The following example defines a policy attribute (10) and takes you into the IKE Policy Attributes:

```
ProCurve(config-ike)#attribute 10
ProCurve(config-ike)#
```
client authentication host

Use the `client authentication host` command to enable the unit to act as an Xauth host when this IKE policy is negotiated with a peer.

Variations of this command include the following:

- `client authentication host username <username>`
- `client authentication host username <username> password <word>`
- `client authentication host username <username> password <word> passphrase <phrase>`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>username &lt;username&gt;</code></td>
<td>Enter the value sent via Xauth as the username.</td>
</tr>
<tr>
<td><code>password &lt;word&gt;</code></td>
<td>Enter the value sent via Xauth as the password.</td>
</tr>
<tr>
<td><code>passphrase &lt;phrase&gt;</code></td>
<td>Optional. Enter the value sent via Xauth as the passphrase. This is only used with authentication type OTP (one time password).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, if this command is not present in the IKE policy the unit does not act as an Xauth host.

**Functional Notes**

The specified credentials are programmed into the unit and there is no prompt for entering values real-time. Therefore, schemes requiring real-time input or additional responses (e.g., SecureID) are not supported. The `client authentication host` command and the `client authentication server` commands are mutually exclusive. See `client authentication server list <listname>` on page 1171 for more information.

**Usage Examples**

The following example specifies the login credentials to be sent:

```
ProCurve(config-ike)#client authentication host username jsmith password password1 passphrase phrase
```
client authentication host xauth-type [generic | otp | radius]

Use the `client authentication host xauth-type` command to allow the user to specify the Xauth authentication type if a type other than `generic` is desired.

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>generic</td>
<td>Specifies generic authentication type.</td>
</tr>
<tr>
<td>otp</td>
<td>Specifies OTP authentication type.</td>
</tr>
<tr>
<td>radius</td>
<td>Specifies RADIUS authentication type.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this is set to generic.

**Functional Notes**

This command is used along with the `client authentication host username`. See `client configuration pool <poolname>` on page 1172 for more information. When acting as an Xauth host, this command allows the user to specify the Xauth authentication type if a type other than generic is desired.

**Usage Examples**

The following example sets the Xauth type to `radius`:

```
ProCurve(config-ike)#client authentication host xauth-type radius
```
client authentication server list `<listname>`

Use the `client authentication server list` command to enable the unit to act as an Xauth server (edge device). Use the `no` form of this command to disable this feature.

**Syntax Description**

`<listname>` Specifies the named list created with the `aaa authentication login` command.

**Default Values**

By default, the router does not act as an Xauth server and extended authentication is not performed.

**Functional Notes**

When this IKE policy is negotiated and the peer has indicated Xauth via the IKE authentication method and/or the Xauth vendor ID, this command allows the unit to perform as an Xauth server (edge device). The specified AAA login method is used to identify the location of the user authentication database. The `client authentication host` and the `client authentication server` commands are mutually exclusive. See `client configuration pool `<poolname>` on page 1172 for more information.

**Usage Examples**

The following example enables Xauth as an Xauth server and specifies which AAA method list to use in locating the user database:

ProCurve(config-ike)#client authentication server list clientusers
client configuration pool <poolname>

Use the client configuration pool command to configure the SROS to perform as mode-config server (edge device) when an IKE policy is negotiated.

Variations of this command include the following:

- client configuration pool <poolname>
- client configuration pool <poolname> initiate
- client configuration pool <poolname> initiate respond
- client configuration pool <poolname> respond
- client configuration pool <poolname> respond initiate

Syntax Description

<poolname> The pool from which to obtain parameters to assign to the client.

Default Values

By default, if this command is not present in the IKE policy, the device allocates mode-config IP addresses, DNS server addresses, and NetBIOS name server addresses, and mode-config is not performed.

Functional Notes

This command ties an existing client configuration pool to an IKE policy.

Usage Examples

The following example ties the ConfigPool1 configuration pool to this IKE policy:

ProCurve(config-ike)#client configuration pool ConfigPool
**initiate [aggressive | main]**

Use the `initiate` command to allow the IKE policy to initiate negotiation (in main mode or aggressive mode) with peers. Use the `no` form of this command to allow the policy to respond only.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggressive</td>
<td>Specifies to initiate using aggressive mode. Aggressive mode can be used when one end of the VPN tunnel has a dynamically assigned address. The side with the dynamic address must be the initiator of the traffic and tunnel. The side with the static address must be the responder.</td>
</tr>
<tr>
<td>main</td>
<td>Specifies to initiate using main mode. Main mode requires that each end of the VPN tunnel has a static WAN IP address. Main mode is more secure than aggressive mode because more of the main mode negotiations are encrypted.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the `main` initiation is enabled.

**Functional Notes**

By using the `initiate` and `respond` commands, you can configure the IKE policy to initiate and respond, initiate only, or respond only. It is an error if you have both `initiate` and `respond` disabled.

**Usage Examples**

The following example enables the SROS device to initiate IKE negotiation in main mode:

```
ProCurve(config-ike)#initiate main
```
local-id [address | asn1-dn | fqdn | user-fqdn] <ipaddress or name>

Use the local-id command to set the local ID for the IKE policy. This setting overrides the system local ID setting (set in the Global configuration mode using the crypto ike local-id address command).

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>Specifies a remote ID of IPv4 type.</td>
</tr>
<tr>
<td>asn1-dn</td>
<td>Specifies an Abstract Syntax Notation Distinguished Name as the remote ID</td>
</tr>
<tr>
<td></td>
<td>(enter this value in LDAP format).</td>
</tr>
<tr>
<td>fqdn</td>
<td>Specifies a fully qualified domain name as the remote ID.</td>
</tr>
<tr>
<td>user-fqdn</td>
<td>Specifies a user fully qualified domain name or email address (e.g.,</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:user1@hp.com">user1@hp.com</a>) as the remote ID.</td>
</tr>
</tbody>
</table>

Default Values

By default, the local ID is not defined.

Functional Notes

The local ID for a particular IKE policy can be set in two ways. The first (default) method is to use the global system command:

ProCurve(config)#crypto ike local-id address

This command, which by default is executed on start-up, makes the local ID of an IKE policy equal to the IPv4 address of the interface on which an IKE negotiation is occurring. This is particularly useful for products that could have multiple public interfaces.

The second method is to use the IKE policy command:

ProCurve(config-ike)#local-id [address | fqdn | user-fqdn] <ipaddress or fqdn>

This policy-specific command allows you to manually set the local ID for an IKE policy on a per-policy basis. You can use both methods simultaneously in the product. Several IKE policies can be created, some of which use the default system setting of the IPv4 address of the public interface. Others can be set to override this system setting and manually configure a local ID specific to those policies. When a new IKE policy is created, they default to no local-id. This allows the system local ID setting to be applied to the policy.

Usage Examples

The following example sets the local ID of this IKE policy to the IPv4 address 172.17.45.57:

ProCurve(config-ike)#local-id address 172.17.45.57
nat-traversal <version> [allow | disable | force]

Use the nat-traversal command to allow, force, or disable NAT traversal version 1 and 2 on a specific Ike policy.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;version&gt;</th>
<th>Enter v1 or v2 to select the NAT traversal version.</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow</td>
<td>Sets the IKE policy to allow the specified NAT traversal version.</td>
</tr>
<tr>
<td>disable</td>
<td>Sets the IKE policy to disable the specified NAT traversal version.</td>
</tr>
<tr>
<td>force</td>
<td>Sets the IKE policy to force the specified NAT traversal version.</td>
</tr>
</tbody>
</table>

Default Values

The defaults for this command are nat-traversal v1 allow and nat-traversal v2 allow.

Usage Examples

The following example disables version 2 on Ike policy 1:

ProCurve(config)#crypto ike policy 1
ProCurve(config-ike)#nat-traversal v2 disable
peer [<ip address> | any]

Use the **peer** command to enter the IP address of the peer device. Repeat this command for multiple peers. Use the **any** keyword if you want to set up a policy that will initiate or respond to any peer.

**Syntax Description**

| <ip address> | Specifies a peer IP address. |
| any | Allows any peer to connect to this IKE policy. |

**Default Values**

There are no default settings for this command.

**Functional Notes**

An IKE policy is incomplete unless one of the peer commands is specified. Only one IKE policy can be configured with **peer any**.

**Usage Examples**

The following example sets multiple peers on an IKE policy for an initiate-and-respond policy using pre-shared secret, DES, MD5, and Diffie-Hellman Group 1:

```
ProCurve(config)#crypto ike policy 100
ProCurve(config-ike)#peer 172.17.45.57
ProCurve(config-ike)#peer 172.31.15.129
ProCurve(config-ike)#peer 192.168.1.3
ProCurve(config-ike)#respond anymode
ProCurve(config-ike)#initiate main
```

The following example sets up a policy allowing any peer to initiate using pre-shared secret, DES, MD5, and Diffie-Hellman Group 1.

```
ProCurve(config)#crypto ike policy 100
ProCurve(config-ike)#peer any
ProCurve(config-ike)#respond anymode
ProCurve(config-ike)#initiate main
```

**Technology Review**

IKE policies must have a peer address associated with them to allow certain peers to negotiate with the product. This is a problem when you have “roaming” users (those who obtain their IP address using DHCP or some other dynamic means). To allow for “roaming” users, the IKE policy can be set up with **peer any** to allow any peer to negotiate with the product. There can only be one **peer any** policy in the running configuration.
respond [aggressive | anymode | main]

Use the respond command to allow the IKE policy to respond to negotiations by a peer. Use the no form of this command to allow the policy to only initiate negotiations.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggressive</td>
<td>Specifies to respond only to aggressive mode.</td>
</tr>
<tr>
<td>anymode</td>
<td>Specifies to respond to any mode.</td>
</tr>
<tr>
<td>main</td>
<td>Specifies to respond only to main mode.</td>
</tr>
</tbody>
</table>

Default Values

By default, respond to any mode is enabled.

Functional Notes

By using the initiate and respond commands, you can configure the IKE policy to initiate and respond, initiate only, or respond only. It is an error if you have both initiate and respond disabled.

Usage Examples

The following example configures the router to initiate and respond to IKE negotiations:

ProCurve(config-ike)#respond anymode
ProCurve(config-ike)#initiate main
AS PATH LIST COMMAND SET

To activate the Autonomous System (AS) Path List Configuration mode, enter the `ip as-path-list` command at the Global Configuration mode prompt. For example:

ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#ip as-path-list listname
ProCurve(config-as-path-list)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310

All other commands for this command set are described in this section in alphabetical order.

- `deny <community-number> | internet | local-as | no-export | no-advertise` on page 1179
- `permit [ <community-number> | internet | local-as | no-export | no-advertise ]` on page 1180
deny<community-number> | internet | local-as | no-export | no-advertise]

Use the **deny** command to add an entry to the community list that denies BGP routes containing the specified community number in the community attribute. Use the **no** form of this command to remove the statement from the community list.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;community-number&gt;</td>
<td>Denies routes that contain this value in their community attribute. This is a numeric value that can be an integer from 1 to 4,294,967,295 or string in the form &quot;aa:nn&quot;, where the value of “aa” is the AS number and the value of “nn” is the community number. Multiple community number parameters can be present in the command.</td>
</tr>
<tr>
<td>internet</td>
<td>Denies routes that contain the reserved community number for the INTERNET community.</td>
</tr>
<tr>
<td>local-as</td>
<td>Denies routes that contain the reserved community number for NO_EXPORT_SUBCONFED. Routes containing this attribute should not be advertised to external BGP peers.</td>
</tr>
<tr>
<td>no-export</td>
<td>Denies routes that contain the reserved community number for NO_EXPORT. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.</td>
</tr>
<tr>
<td>no-advertise</td>
<td>Denies routes that contain the reserved community number for NO_ADVERTISE. Routes containing this attribute should not be advertised to any BGP peer.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example creates a community list named **MyList** to deny BGP routes that have the Internet community number in their community attribute:

ProCurve(config)#ip as-path-list MyList
ProCurve(config-comm-list)#deny internet
permit [<community-number> | internet | local-as | no-export | no-advertise]

Use the `permit` command to add an entry to the community list that allows only BGP routes containing the specified community number in the community attribute. Use the `no` form of this command to remove the statement from the community list.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;community-number&gt;</code></td>
<td>Permits routes that contain this value in their community attribute. This is a numeric value that can be an integer from 1 to 4,294,967,295 or string in the form &quot;aa:nn&quot;, where the value of &quot;aa&quot; is the AS number and the value of &quot;nn&quot; is the community number. Multiple community number parameters can be present in the command.</td>
</tr>
<tr>
<td>internet</td>
<td>Permits routes that contain the reserved community number for the INTERNET community.</td>
</tr>
<tr>
<td>local-as</td>
<td>Permits routes that contain the reserved community number for NO_EXPORT_SUBCONFED. Routes containing this attribute should not be advertised to external BGP peers.</td>
</tr>
<tr>
<td>no-export</td>
<td>Permits routes that contain the reserved community number for NO_EXPORT. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.</td>
</tr>
<tr>
<td>no-advertise</td>
<td>Permits routes that contain the reserved community number for NO_ADVERTISE. Routes containing this attribute should not be advertised to any BGP peer.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example permits BGP routes that match the as-path attributes:

```
ProCurve(config)#ip as-path-list listname
ProCurve(config-as-path-list)#permit 30:22
```
ROUTE MAP COMMAND SET

To activate the Route Map Interface Configuration mode, enter the `route-map` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#route-map MyMap permit 100
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `match as-path <name>` on page 1182
- `match community <name> [exact-match]` on page 1183
- `match ip address <access list name>` on page 1184
- `match ip address prefix-list <prefix-list name>` on page 1185
- `match ip dscp [<value> | afxx | csxx | default | ef]` on page 1186
- `match ip precedence [<value> | critical | flash | flash-override | immediate | internet | network | priority | routine]` on page 1189
- `match metric <value>` on page 1192
- `set as-path prepend [<number> | last-as <number>]` on page 1193
- `set comm-list <name> delete` on page 1194
- `set community [<community-number> | internet | local-as | no-export | no-advertise | none] [add]` on page 1195
- `set default interface [<interface> | null 0]` on page 1196
- `set interface <interface type> <interface id>` on page 1197
- `set ip default next-hop [<interface> | null 0]` on page 1198
- `set ip df` on page 1199
- `set ip dscp [<value> | afxx | csxx | default | ef]` on page 1200
- `set ip next-hop <address>` on page 1201
- `set ip precedence [<value> | critical | flash | flash-override | immediate | internet | network | priority | routine]` on page 1202
- `set local-preference <value>` on page 1203
- `set metric <value>` on page 1204
**match as-path** `<name>`

Use the `match as-path` command to configure the route map to route traffic based on the AS path list name. Use the `no` form of this command to discontinue matching.

**Syntax Description**

| `<name>` | Specifies the name of the AS path list to match. AS path lists are created using the `ip as-path-list` command (in the Global configuration mode). |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named `MyMap` to match the AS path list named `TestPath`:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#match as-path TestPath
```
**match community <name> [exact-match]**

Use the `match community` command to configure the route map to route traffic based on a specified community. Use the `no` form of this command to discontinue matching.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;name&gt;</th>
<th>Specifies the name of the community you want to match.</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact-match</td>
<td>Optional. Specifies that the route map must match the community name exactly.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named `MyMap` to match the community named `MyCommunity`:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#match community MyCommunity
```
match ip address <access list name>

Use the **match ip address** command to configure the route map to route traffic based on the access list name defined with the **ip access-list** command. Refer to **ip access-list extended <listname>** on page 350 for more information. Use the **no** form of this command to discontinue matching.

**Syntax Description**

| <access list name> | Specifies the name of the access list to match. |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named **MyMap** to match the IP address access list named **MyList**:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#match ip address MyList
```
**match ip address prefix-list** `<prefix-list name>`

Use the **match ip address prefix-list** command to configure the route map to route traffic based on a prefix list route filter. The name of the prefix list is defined with the **ip prefix-list** command. Refer to **ip prefix-list** `<listname>` *description* `<"text">` on page 409 for more information. Use the **no** form of this command to discontinue matching.

**Syntax Description**

| `<prefix-list name>` | Specifies matching the IP address based on the prefix list name. |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named **MyMap** to match the IP address prefix list named **MyList**:

ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#match ip address prefix-list MyList
match ip dscp [<value> | afxx | csxx | default | ef]

Use the `match ip dscp` command to configure the route map to route traffic based on the Differentiated Services Code Point (DSCP) value in the IP header of the packet. Use the `no` form of this command to discontinue matching.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;value&gt;</code></td>
<td>Specifies the DSCP numeric value to match. (Valid range: 0 to 63.)</td>
</tr>
<tr>
<td>afxx</td>
<td>Specifies the assured forwarding (AF) value to match. (Select from: 11, 12, 13, 21, 22, 23, 31, 32, 33, 41, 42, or 43.)</td>
</tr>
<tr>
<td>csxx</td>
<td>Specifies the class selector (CS) value to match. (Valid range: 1 to 7.)</td>
</tr>
<tr>
<td>default</td>
<td>Specifies matching the default IP DSCP value.</td>
</tr>
<tr>
<td>ef</td>
<td>Specifies matching those packets marked for expedited forwarding (EF).</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Functional Notes**

The Differentiated Services (DiffServ or DS) model was created in RFC2474 and 2475 to build on the original Type of Service field by creating a six-bit sequence (combining the precedence value with the delay, throughput, and reliability bits). This six-bit sequence increased the number of available values from 8 to 64. The DiffServ model introduced a new concept to QoS in the IP network environment: per-hop behaviors (PHBs). The PHB premise is that pieces equipment using the DiffServ model have an agreed upon set of rules (PHB types) for handling certain network traffic. Though the RFC explicitly defines what each PHB should be capable of, it does not restrict vendor-specific implementation of the PHBs. Each vendor is free to decide how their network product implements the various defined PHBs.

According to RFC2474, the DS field contains the following bits:

```
   0   1   2   3   4   5   6   7
    |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |
Differentiated Service Code Point Unused*
```

* The previously unused bits in the DS field are now used for congestion control and are not discussed in this document.

Equipment following the DiffServ model (DS-compliant nodes) must use the entire six-bit DSCP value to determine the appropriate PHB. The PHBs are defined as the following:

- Default PHB
- Class Selector PHB
- Assured Forwarding PHB (RFC2597)
- Expedited Forwarding PHB (RFC2598)
**Default PHB**

All DS-compliant nodes must provide a Default PHB to offer best-effort forwarding service. For Default PHBs, the DSCP value is 0. Any packet that does not contain a standardized DSCP should be mapped to the Default PHB and handled accordingly.

**Class Selector PHB**

In the Class Selector PHB, the first three bits in the DSCP value are used for backwards compatibility to systems implementing IP precedence. In this scenario, all but the first three bits of the DS field are set to 0. This compatibility requires DS-compliant nodes to provide the same data services as are provided by nodes implementing IP precedence. The following table is a comparison of IP precedence values to their corresponding DSCP values.

<table>
<thead>
<tr>
<th>IP Precedence Value (bits)</th>
<th>DSCP Value (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (000)</td>
<td>0 (000000)</td>
</tr>
<tr>
<td>1 (001)</td>
<td>8 (001000)</td>
</tr>
<tr>
<td>2 (010)</td>
<td>16 (010000)</td>
</tr>
<tr>
<td>3 (011)</td>
<td>24 (011000)</td>
</tr>
<tr>
<td>4 (100)</td>
<td>32 (100000)</td>
</tr>
<tr>
<td>5 (101)</td>
<td>40 (101000)</td>
</tr>
<tr>
<td>6 (110)</td>
<td>48 (110000)</td>
</tr>
<tr>
<td>7 (111)</td>
<td>56 (111000)</td>
</tr>
</tbody>
</table>

**Assured Forwarding PHB**

The flexibility of DiffServ allows for more developed subclasses of service within each main class using the last three bits of the DSCP. As defined in RFC2597, the Assured Forwarding PHB creates four main classes of service:

<table>
<thead>
<tr>
<th>Class</th>
<th>DSCP Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF1</td>
<td>001XX0</td>
</tr>
<tr>
<td>AF2</td>
<td>010XX0</td>
</tr>
<tr>
<td>AF3</td>
<td>011XX0</td>
</tr>
<tr>
<td>AF4</td>
<td>100XX0</td>
</tr>
</tbody>
</table>

X indicates a "do not care" value

The first three bits of the DSCP specify the class, and the last bit is always zero. Each class is separated into subclasses using the two remaining bits in the DSCP (bits 3 and 4). The subclasses are divided based on the likelihood that packets in the class are dropped in the event of network congestion. The higher the
value for bits 3 and 4, the greater the likelihood that the packets will be dropped.

<table>
<thead>
<tr>
<th>Bit 3</th>
<th>Bit 4</th>
<th>Drop Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Low</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Medium</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>High</td>
</tr>
</tbody>
</table>

The following table lists the Assured Forwarding PHB subclasses and their corresponding DSCP bits and values.

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
<th>DSCP Bits</th>
<th>DSCP Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF1</td>
<td>1</td>
<td>001010</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>001100</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>001110</td>
<td>14</td>
</tr>
<tr>
<td>AF2</td>
<td>1</td>
<td>010010</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>010100</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>010110</td>
<td>22</td>
</tr>
<tr>
<td>AF3</td>
<td>1</td>
<td>011010</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>011100</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>011110</td>
<td>30</td>
</tr>
<tr>
<td>AF4</td>
<td>1</td>
<td>100010</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>100100</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>100110</td>
<td>38</td>
</tr>
</tbody>
</table>

**Expedited Forwarding PHB**
RFC2598 created a new DiffServ PHB intended to provide the best service possible on an IP network. Packets using the Expedited Forwarding PHB markings should be provided service to reduce latency, jitter, and dropped packets, and should be guaranteed bandwidth during the entire end-to-end transmission journey through the network. The DSCP value for the Expedited Forwarding PHB is 46 (DSCP bits are 101110).

**Usage Examples**
The following example instructs the route map named **MyMap** to match the IP header with a DSCP Assured Forwarding Class 1, Subclass 2 (**af12**):

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#match ip dscp af12
```
**match ip precedence [<value> | critical | flash | flash-override | immediate | internet | network | priority | routine]**

Use the `match ip precedence` command to configure the route map to route traffic based on the precedence value in the IP header of the packet. Use the `no` form of this command to discontinue matching.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;value&gt;</td>
<td>Specifies matching the IP precedence (in numeric value). (Valid range: 0 to 7 in ascending order of importance.)</td>
</tr>
<tr>
<td>routine</td>
<td>Specifies matching the IP precedence <code>routine</code>. (Numeric value of 0.)</td>
</tr>
<tr>
<td>priority</td>
<td>Specifies matching the IP precedence <code>priority</code>. (Numeric value of 1.)</td>
</tr>
<tr>
<td>immediate</td>
<td>Specifies matching the IP precedence <code>immediate</code>. (Numeric value of 2.)</td>
</tr>
<tr>
<td>flash</td>
<td>Specifies matching the IP precedence <code>flash</code>. (Numeric value of 3.)</td>
</tr>
<tr>
<td>flash-override</td>
<td>Specifies matching the IP precedence <code>flash-override</code>. (Numeric value of 4.)</td>
</tr>
<tr>
<td>critical</td>
<td>Specifies matching the IP precedence <code>critical</code>. (Numeric value of 5.)</td>
</tr>
<tr>
<td>internet</td>
<td>Specifies matching the IP precedence <code>internet</code>. (Numeric value of 6.)</td>
</tr>
<tr>
<td>network</td>
<td>Specifies matching the IP precedence <code>network</code>. (Numeric value of 7.)</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Functional Notes**

RFC791 created a single octet (labeled Type of Service) to help with the difficulty of trying to provide QoS handling in IP networks.

According to RFC791, the Type of Service field contains the following bits:

```
  0  1  2  3  4  5  6  7
   |   |   |   |   |   |   |
   Precedence  Delay  Reliability  Unused
```
The three-bit IP Precedence values (0 through 7) are specified as:

111  Network Control Packets
110  Internetwork Control Packets
101  Critical Traffic
100  Flash Override
011  Flash
010  Immediate Servicing
001  Priority Traffic
000  Routine Data

The IP Precedence values provide network routers with information about the kind of traffic contained in the IP packet. Based on the IP Precedence values, some networks (when supported) can offer special handling to certain packets. In addition, providing IP Precedence values to critical traffic (such as route information) ensures that critical packets will always be delivered regardless of network congestion. This traffic is often critical to network and internetwork operation. In general, the higher the IP Precedence value, the more important the traffic and the better handling it should receive in the network. It is important to remember that not all equipment in the public IP network will be configured to recognize and handle IP Precedence values. While it is a good idea to set the values for critical traffic, it does not guarantee special handling.

In addition to the IP Precedence values, RFC791 specifies bits for delay, throughput, and reliability to help balance the needs of particular traffic types when traveling on the IP network infrastructure. When these bits are set to 0, they are handled with normal operation. When set to 1, each bit specifies premium handling for that parameter. For example, a 1 in the delay position indicates that the traffic is delay-sensitive and care should be taken to minimize delay. A 1 in the throughput position indicates that the traffic has higher bandwidth requirements that should be met. A 1 in the reliability position indicates that the traffic is sensitive to delivery issues and care should be taken to ensure proper delivery with all packets of this type. These extra bits are rarely used because it is quite difficult to balance the cost and benefits of each parameter (especially when more than one bit is set to 1).

Usage Examples

The following example instructs the route map named MyMap to match the IP Precedence value of critical:

ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#match ip precedence critical
**match length** `<minimum> <maximum>`

Use the `match length` command to configure the route map to route traffic based on the packet length. Use the `no` form of this command to discontinue matching.

**Syntax Description**

|   | Specifies the minimum packet length you want to match. (Valid range: 1 to 4,294,967,295.) |
|   | Specifies the maximum packet length you want to match. (Valid range: 1 to 4,294,967,295.) |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named **MyMap** to match packets with a minimum length of 1 and a maximum length of 200:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#match length 1 200
```
match metric <value>

Use the **match metric** command to configure the route map to route traffic based on a specified Multi-Exit Discriminators (MED) value. Use the **no** form of this command to discontinue matching.

**Syntax Description**

| <value>  | Specifies the MED value you want to match. (Valid range: 1 to 4,294,967,295.) |

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named **MyMap** to match the MED value of **100**:  

ProCurve(config)#route-map MyMap permit 100  
ProCurve(config-route-map)#match metric 100
set as-path prepend [<number> | last-as <number>]

Use the set as-path prepend command to prepend a number to the AS path to influence the best-path selection process by making the AS path appear further away. Use the no form of this command to disable this feature.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>as-path prepend &lt;number&gt;</td>
<td>Specifies a number to be prepended to the AS path value as an autonomous number. (Valid range: 1 to 65,535.)</td>
</tr>
<tr>
<td>as-path prepend last-as &lt;number&gt;</td>
<td>Specifies a number to be prepended to the last AS path number. (Valid range: 1 to 10.)</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example prepends the number 2 to the last AS path number:

ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set as-path prepend last-as 2
set comm-list <name> delete

Use the set comm-list delete command to specify a list of communities to delete. Use the no form of this command to disable this feature.

**Syntax Description**

- `<name>` Specifies the name of the list of communities to delete.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example deletes the community list named `listname`:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set comm-list listname delete
```
**set community** [<community-number>] | internet | local-as | no-export | no-advertise | none] [add]

Use the `set community` command to modify the community attribute for all paths serviced by the route map. Use the `no` form of this command to disable this feature.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;community-number&gt;</code></td>
<td>Sets the community attribute to the specified community number for routes serviced by this route map. This is a numeric value that can be an integer from 1 to 4,294,967,295 or string in the form &quot;aa:nn&quot;, where the value of &quot;aa&quot; is the AS number and the value of &quot;nn&quot; is the community number. Multiple community-number parameters can be present in the command.</td>
</tr>
<tr>
<td>internet</td>
<td>Sets the community attribute to the INTERNET community number for routes serviced by this route map.</td>
</tr>
<tr>
<td>local-as</td>
<td>Sets the community attribute to the NO_EXPORT_SUBCONFED community number for routes serviced by this route map. Routes containing this attribute should not be advertised to external BGP peers.</td>
</tr>
<tr>
<td>no-export</td>
<td>Sets the community attribute to the NO_EXPORT community number for routes serviced by this route map. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.</td>
</tr>
<tr>
<td>no-advertise</td>
<td>Sets the community attribute to the NO_ADVERTISE community number for routes serviced by this route map. Routes containing this attribute should not be advertised to any BGP peer.</td>
</tr>
<tr>
<td>none</td>
<td>Removes all communities from BGP routes serviced by this route map.</td>
</tr>
<tr>
<td>add</td>
<td>Appends the listed community number to the end of the community attribute for routes serviced by this route map.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example sets the community number for BGP routes to the well-known INTERNET community:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set community internet
```
**set default interface [\(<interface>\) | null 0]**

Use the `set default interface` command to specify a default interface to redirect traffic to the specified interface if there is no specific routing information for the traffic. If more than one interface is specified, the router uses the first available interface from the list. Use the `no` form of this command to remove the default interface.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;interface&gt;</code></td>
<td>Specifies the default interface. Type <code>set default interface ?</code> for a list of valid interface types.</td>
</tr>
<tr>
<td>null 0</td>
<td>Redirects traffic to the specified interface regardless of available routing information.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example sets the default interface to the ppp 1 interface:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set default interface ppp 1
```
set interface <interface type> <interface id>

Use the set interface command to specify an output interface for the packet. Multiple interfaces can be specified. The router forwards the packet along the first usable interface. Use the no form of this command to cancel output from the specified interface.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt;</td>
<td>Sets output interface type for the packet. Type set interface ? for a list of valid interfaces.</td>
</tr>
<tr>
<td>&lt;interface id&gt;</td>
<td>Specifies the ID of the specified interface type.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example sets the output interface as ppp 1:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set interface ppp 1
```
set ip default next-hop [\textless interface\textgreater | null 0]

Use the \texttt{set ip default next-hop} command to set the next-hop IP address to the specified interface’s address for all routes serviced by the route map that do no have explicit routing information available. Use the \texttt{no} form of this command to remove the configured default next-hop.

\textbf{Syntax Description}

\begin{itemize}
  \item \texttt{<interface>} Specifies the default interface. Type \texttt{set default next-hop ?} for a list of valid interface types.
  \item \texttt{null 0} Redirects traffic to the specified interface regardless of available routing information.
\end{itemize}

\textbf{Default Values}

No default value necessary for this command.

\textbf{Usage Examples}

The following example sets the default next-hop interface to the ppp 1 interface:

\begin{verbatim}
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set default next-hop ppp 1
\end{verbatim}
**set ip df**

Use the `set ip df` command to identify the packet as “don’t fragment” (DF). Use the `no` form of this command to remove this designation.

**Syntax Description**
No subcommands.

**Default Values**
No default value necessary for this command.

**Usage Examples**
The following example designates the packet as “don’t fragment”:

```plaintext
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set ip df
```
**set ip dscp [<value> | afxx | csxx | default | ef]**

Use the `set ip dscp` command to configure the route map to set the Differentiated Services Code Point (DSCP) value in the IP header of the packet for traffic serviced by this route map. For more details on DSCP values, see the command `match ip dscp [<value> | afxx | csxx | default | ef]` on page 1186. Use the `no` form of this command to remove the specified DSCP value.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;value&gt;</td>
<td>Specifies the DSCP numeric value. (Valid range: 0 to 63.)</td>
</tr>
<tr>
<td>afxx</td>
<td>Specifies the assured forwarding (AF) class and subclass. Select from: 11 (001010), 12 (001100), 13 (001110), 21 (010010), 22 (010100), 23 (010110), 31 (011010), 32 (011100), 33 (011110), 41 (100010), 42 (100100), or 43 (100110).</td>
</tr>
<tr>
<td>csxx</td>
<td>Specifies the class selector (CS) value. (Valid range: 1 to 7.)</td>
</tr>
<tr>
<td>default</td>
<td>Specifies the default IP DSCP value (000000).</td>
</tr>
<tr>
<td>ef</td>
<td>Specifies marking for expedited forwarding (EF).</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named `MyMap` to set the IP header with a DSCP Assured Forwarding Class 1, Subclass 2 (`af12`):

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set ip dscp af12
```
**set ip next-hop <address>**

Use the `set ip next-hop` command to set the next-hop IP address to the specified address for all routes serviced by the route map. Use the `no` form of this command to remove the configured next-hop address.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Specifies the IP address in dotted decimal notation (a.b.c.d). More than one address can be entered, and the router uses the first available route from the list.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example sets the ip next-hop interface to 192.168.5.61:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set ip next-hop 192.168.5.61
```
set ip precedence [<value> | critical | flash | flash-override | immediate | internet | network | priority | routine]

Use the `set ip precedence` command to configure the route map to set the precedence value in the IP header of the packet for traffic serviced by the route map. For more details on IP Precedence values, see the command `match ip precedence [<value> | critical | flash | flash-override | immediate | internet | network | priority | routine]` on page 1189. Use the `no` form of this command to removed the specified IP Precedence value.

**Syntax Description**

- `<value>`: Specifies the IP precedence (in numeric value). (Valid range: 0 to 7 in ascending order of importance.)
- `routine`: Specifies the IP precedence routine. (Numeric value of 0.)
- `priority`: Specifies the IP precedence priority. (Numeric value of 1.)
- `immediate`: Specifies the IP precedence immediate. (Numeric value of 2.)
- `flash`: Specifies the IP precedence flash. (Numeric value of 3.)
- `flash-override`: Specifies the IP precedence flash-override. (Numeric value of 4.)
- `critical`: Specifies the IP precedence critical. (Numeric value of 5.)
- `internet`: Specifies the IP precedence internet. (Numeric value of 6.) This level is reserved for internal network use.
- `network`: Specifies the IP precedence network. (Numeric value of 7.) This level is reserved for internal network use.

**Default Values**

No default value necessary for this command.

**Usage Examples**

The following example instructs the route map named MyMap to set the IP Precedence value to critical:

```
ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set ip precedence critical
```
set local-preference <value>

Use the set local-preference command to restrict traffic to a local autonomous system. Use the no form of this command to cancel the local preference.

Syntax Description

| <value> | Sets the local preference value. (Valid range: 0 to 4,294,967,295.) |

Default Values

No default value necessary for this command.

Usage Examples

The following example sets the local preference from MyMap to a value of 100:

ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set local-preference 100
set metric <value>

Use the set metric command to specify a metric value for the route map. Use the no form of this command to cancel the metric value.

Syntax Description

<value>  Sets the metric value. (Valid range: 0 to 4,294,967,295.)

Default Values

No default value necessary for this command.

Usage Examples

The following example sets the metric value for MyMap to 100:

ProCurve(config)#route-map MyMap permit 100
ProCurve(config-route-map)#set metric 100
BGP CONFIGURATION COMMAND SET

To activate the BGP Configuration mode, enter the `bgp` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

```
bgp always-compare-med on page 1206
bgp [compare-med | ignore-med] on page 1207
bgp default local-preference <metric> on page 1208
bgp deterministic-med on page 1209
bgp fast-external-fallover on page 1210
bgp log-neighbor-changes on page 1211
bgp router-id <ip address> on page 1212
distance bgp <external> <internal> <local> on page 1213
hold-timer <hold time> on page 1214
maximum-paths <value> on page 1215
neighbor <address> on page 1216
network <address> mask <mask> on page 1217
```
**bgp always-compare-med**

Use the `bgp always-compare-med` command to configure the SROS to always compare the Multi-Exit Discriminators (MEDs) for all paths for a route, regardless of the autonomous system (AS) through which the paths pass. Use the `no` form of this command to disable this feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is disabled.

**Usage Examples**

The following example enables this option:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#bgp always-compare-med
```
bgp [compare-med | ignore-med]

Use the **bgp compare-med** command to configure the SROS to compare the Multi-Exit Discriminators (MEDs) for all routes from the same autonomous system (AS). Use the **bgp ignore-med** to configure the SROS to disregard all MED attributes for all received routes.

**Syntax Description**

No subcommands.

**Default Values**

By default, the SROS compares the MED attributes for routes from the same AS.

**Usage Examples**

The following example enables this option:

```
ProCurve(config)#router bgp 65000
ProCurve(config)#bgp compare-med
```
**bgp default local-preference** `<metric>`

Use the `bgp default local-preference` command to change the local preference for all BGP routes. The local preference is an attribute (LOCAL_PREF) that indicates a degree of preference for a route relative to other routes in the local autonomous system (AS). BGP4 neighbors can send the local preference value as an attribute of a route in an UPDATE message. Local preference only applies to routes within the local AS. Use the `no` form of this command to return to the default setting.

**Syntax Description**

`<metric>` Specifies the new local preference. Valid range is 0 to 4,294,967,295.

**Default Values**

The default local preference is 100.

**Usage Examples**

The following example changes the default local preference to 200:

```
ProCurve(config)#router bgp 65000
ProCurve(config)#bgp local-preference 200
```
bgp deterministic-med

Use the `bgp deterministic-med` command to configure the SROS to compare the Multi-Exit Discriminators (MEDs) for all BGP routes received from different neighbors within the same AS. Use the `no` form of this command to disable this option.

**Syntax Description**

No subcommands.

**Default Values**

By default, this option is disabled.

**Usage Examples**

The following example enables the SROS to use the deterministic MED option:

```
ProCurve(config)#router bgp 65000
ProCurve(config)#bgp deterministic-med
```
**bgp fast-external-fallover**

Use the `bgp fast-external-fallover` command to enable the fast-external-fallover feature. Use the `no` form of this command to disable this feature.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is enabled.

**Functional Notes**

When enabled, if the link interface over which the router is communicating with a BGP peer goes down, the BGP session with that peer is immediately cleared. When fallover is disabled and the link goes down, the session is maintained until the BGP hold timer expires.

**Usage Examples**

The following example enables this option:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#bgp fast-external-fallover
```
**bgp log-neighbor-changes**

Use the `bgp log-neighbor-changes` command to control the logging of neighbor state changes. Use the `no` form of this command to return to the default setting.

**Syntax Description**

No subcommands.

**Default Values**

By default, neighbor changes are not logged.

**Functional Notes**

This command controls logging of BGP neighbor state changes (up/down) and resets. This information is useful for troubleshooting and determining network stability.

**Usage Examples**

The following example enables logging of BGP neighbor state changes:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#bgp log-neighbor-changes
```
**bgp router-id <ip address>**

Use the **bgp router-id** command to specify the IP address that the router should use as its BGP router ID. Use the **no** form of this command to return to the default setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Designates the IP address this router should use as its BGP router ID.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no router ID is configured. The default action is detailed in *Functional Notes*, below.

**Functional Notes**

This command allows an IP address to be specified for use as the BGP router ID. If no IP address is configured at BGP startup, it uses the highest IP address configured on a loopback interface. If no loopback interfaces are configured, it uses the highest IP address configured on any interface that is active. If the specified router ID is changed, existing sessions with BGP neighbors are reset.

**Usage Examples**

The following example configures IP address 10.0.0.1 as the BGP router ID:

```plaintext
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#bgp router-id 10.0.0.1
```
**distance bgp** `<external> <internal> <local>`

Use the `distance bgp` command to set the administrative distance for BGP routes. Use the `no` form of this command to return to the default setting.

**Syntax Description**

- `<external>`
  Sets the administrative distance for BGP routes learned via eBGP sessions.
  A value of 255 means the route is not installed in the route table. Range: 1 to 254.

- `<internal>`
  Sets the administrative distance for BGP routes learned via iBGP sessions.
  A value of 255 means the route is not installed in the route table. Range: 1 to 254.

- `<local>`
  Sets the administrative distance for BGP routes learned via the network command and redistribution. A value of 255 means the route is not installed in the route table. Range: 1 to 254.

**Default Values**

By default external is set to 20, internal to 200, and local to 200. Normally, these default settings should not be changed.

**Functional Notes**

This command sets the administrative distance for BGP routes. The administrative distance is a local variable that allows a router to choose the best route when there are multiple paths to the same network. Routes with smaller administrative distances are favored.

**Usage Examples**

The following example gives external BGP routes an administrative distance of 30, internal BGP routes an administrative distance of 200, and local routes an administrative distance of 240:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#distance bgp 30 200 240
```
**hold-timer** `<hold time>`

Use the `hold-timer` command to set the default hold time for all neighbors in the BGP process.

**Syntax Description**

<`hold time>` Specifies a time interval (in seconds) within which a keepalive must be received from a peer before it is declared dead peer. Range: 0 to 65535

**Default Values**

*By default, the hold time is 90 seconds.*

**Functional Notes**

Using the `hold-timer` command in BGP configuration mode sets the default hold time for all neighbors in that BGP process. Using the `hold-timer` command in BGP neighbor configuration mode sets the hold time for only that neighbor. The peers will negotiate and use the lowest configured setting. The keepalive interval will be set to one third of the negotiated hold time.

**Usage Examples**

The following example sets a hold time of 120 seconds for a specific neighbor, with an understood keepalive interval of 40 seconds:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#hold-timer 120
```
maximum-paths <value>

Use the maximum-paths command to specify the number of parallel routes (shared paths) eBGP neighbors can inject into the route table. When IP load sharing is enabled, BGP4 can balance traffic to a specific destination across up to six equal paths. Use the no form of this command to return to the default value.

Syntax Description

| <value> | Specifies the number of parallel routes eBGP neighbors can inject into the route table. Valid range is 1 to 6. |

Default Values

By default, a single path can exist in the route table.

Usage Examples

The following example configures the SROS to allow four parallel paths in the route table:

ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#maximum-paths 4
neighbor <address>

Use the `neighbor` command to create a BGP neighbor, specify an IP address, and activate the BGP neighbor configuration commands. Refer to *BGP Neighbor Configuration Command Set* on page 1218 for more information on neighbor-specific configuration parameters. Use the `no` form of this command to remove the configured neighbor.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Specifies the IP address for the neighbor in dotted decimal notation.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no configured BGP neighbors.

**Usage Examples**

The following example configures a BGP neighbor with an IP address of 172.20.2.50:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.20.2.50
ProCurve(config-bgp-neighbor)#
```
**network** `<address>` **mask** `<mask>`

Use the **network** command to add a network to the BGP network table. Use the **no** form of this command to remove the configured network.

**Syntax Description**

| `<address>` | Specifies the network address for the neighbor the SROS will advertise over BGP. Enter the network in dotted decimal notation `<A.B.C.D>`.
| **mask `<mask>`** | Specifies the subnet mask for the specified neighbor in dotted decimal notation `<A.B.C.D>`.

**Default Values**

By default, there are no configured BGP networks.

**Usage Examples**

The following example adds the 172.20.2.0 network with a subnet mask of 255.255.255.0:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#network 172.20.2.0 mask 255.255.255.0
```
BGP NEIGHBOR CONFIGURATION COMMAND SET

To activate the BGP Neighbor Configuration mode, enter the `bgp-neighbor` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 65.4.3.192
ProCurve(config-bgp-neighbor)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `exit` on page 1310
- `ping <address>` on page 1311
- `shutdown` on page 1315

All other commands for this command set are described in this section in alphabetical order.

- `advertisement-interval <seconds>` on page 1219
- `as-path-list <listname> [in | out]` on page 1220
- `ebgp-multihop <hop count>` on page 1221
- `hold-timer <hold time>` on page 1222
- `local-as <as-number>` on page 1223
- `next-hop-self` on page 1224
- `password <password>` on page 1225
- `prefix-list <listname> [in | out]` on page 1226
- `remote-as <as-number>` on page 1227
- `route-map <map-name> [in | out]` on page 1228
- `send-community standard` on page 1229
- `soft-reconfiguration inbound` on page 1230
- `update-source <interface>` on page 1231
advertisement-interval <seconds>

Use the `advertisement-interval` command to configure the SROS to specify how long the BGP process waits before sending updates to the neighbor.

**Syntax Description**

<table>
<thead>
<tr>
<th>&lt;seconds&gt;</th>
<th>Specifies the advertisement interval in seconds. Range: 0 to 600.</th>
</tr>
</thead>
</table>

**Default Values**

By default, the advertisement interval is 30 seconds for external neighbors and 5 seconds for internal neighbors.

**Functional Notes**

This command sets the minimum interval between sending updates to the specified neighbor.

**Usage Examples**

The following example configures the BGP process to wait at least 100 seconds before sending updates to the neighbor:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#advertisement-interval 100
```
as-path-list <listname> [in | out]

Use the `as-path-list` command to assign a predefined autonomous system (AS) path list to a BGP neighbor. This list is then used to filter inbound and/or outbound BGP route updates. Use the `no` form of this command to discontinue use of the list.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listname&gt;</td>
<td>Assigns an AS path list to this BGP neighbor.</td>
</tr>
<tr>
<td>in</td>
<td>Specifies the filtering of all inbound BGP route updates.</td>
</tr>
<tr>
<td>out</td>
<td>Specifies the filtering of all outbound BGP route updates.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no AS path lists are specified for filtering.

**Functional Notes**

Before they can be assigned to a neighbor, AS path lists must first be defined using the `ip as-path-list` command. See `ip as-path-list <listname>` on page 356 for more information.

**Usage Examples**

The following example uses the `no15` AS path list to filter all inbound BGP route updates:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 10.4.3.192
ProCurve(config-bgp-neighbor)#as-path-list no15 in
```
ebgp-multihop <hop count>

Use the **ebgp-multihop** command to configure the maximum hop count of BGP messages to a neighbor. Use the **no** form of this command to return to the default setting.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;hop count&gt;</td>
<td>Specifies the maximum hop count of BGP messages to a neighbor. Range: 1 to 254.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, eBGP multihop is set to 1.

**Functional Notes**

This command allows an eBGP neighbor to be on a network that is not directly connected. Normally, eBGP peers are directly connected. In certain applications, a non-BGP device such as a firewall or router may reside between eBGP peers. In this case, the eBGP multihop command is required to allow updates to have a TTL greater than 1 and to allow received BGP updates to be added to the BGP table when the next hop address is not directly connected.

**Usage Examples**

The following example allows a BGP message to travel 10 hops to a neighbor:

ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 65.4.3.192
ProCurve(config-bgp-neighbor)#ebgp-multihop 10
**hold-timer** `<hold time>`

Use the `hold-timer` command to set the default hold time for BGP neighbors.

**Syntax Description**

```
<hold time> Specifies a time interval (in seconds) within which a keepalive must be received from a peer before it is declared dead peer. Range: 0 to 65,535.
```

**Default Values**

By default, the hold time is 90 seconds.

**Functional Notes**

Using the `hold-timer` command in BGP configuration mode sets the default hold time for all BGP neighbors. Using the `hold-timer` command in BGP neighbor configuration mode sets the hold time for the specific neighbor. The peers will negotiate and use the lowest configured setting. The keepalive interval will be set to one-third of the negotiated hold time.

**Usage Examples**

The following example sets a hold time of 120 seconds for a specific neighbor, with an understood keepalive interval of 40 seconds:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#hold-timer 120
```
**local-as <as-number>**

Use the *local-as* command to specify an autonomous system (AS) number for the unit to use when communicating with this BGP neighbor. Use the *no* form of this command to return to default settings.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;as-number&gt;</code></td>
<td>Specifies the AS number to use when communicating with this neighbor. Must be different than the AS number for this router and the peer router. Only valid for eBGP connections. Range is 0 to 65,535.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, no local AS number is defined. The router’s BGP AS number is used.

**Functional Notes**

This command substitutes a different AS number to be used for communicating with this BGP neighbor. (other than the one the router is actually a member of). This can be used to satisfy network designs requiring a customer to appear as one AS number when communicating with one Internet service provider (ISP) and another when communicating with another ISP.

**Usage Examples**

The following example configures this BGP neighbor’s AS number to be 300:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#local-as 300
```

**Technology Review**

This router appears (to the peer router) to be in the AS specified with the *local-as* command. In network advertisements from routers using the *local-as* command, the first router’s true AS number (the number specified using the *router bgp as-number* command) is prepended to the AS path attribute, then the local AS (the number specified using the *local-as* command) is prepended to the AS path attribute. This makes it appear that the path from this router to the network is first through the local AS, and then through the true AS. To further illustrate, consider the following example network.

In this network:
- Router A is in AS 100.
- Router B is in AS 300.
- Router A is an eBGP peer with Router B.
- Router A’s connection to Router B specifies a *local-as* of 200.
- Router B is configured to connect to Router A in AS 200.

Therefore:
- To Router B, all aspects of Router A appear as AS 200.
- Networks advertised from Router A to Router B will have the AS path **200 100** prepended to the AS path attribute.
- Router A will add AS 200 to the AS path of networks learned from Router B.
next-hop-self

Use the next-hop-self command to force the next hop attribute to be changed to this unit’s address when advertising networks that would not have the next hop changed under normal rules. Normal next hop rules are described in the Functional Notes section below. Use the no form of this command to cause normal next hop rules to apply.

Syntax Description

No subcommands.

Default Values

By default, this command is disabled and normal next hop rules apply.

Functional Notes

In eBGP, routes are normally advertised with a next hop set to the IP address that the receiving peer has configured in its neighbor statement for this router. In the eBGP case where the receiving router is in the same subnet as the current next hop, the current next hop is not changed.

For broadcast multiaccess networks (Ethernet), this provides more efficient routing. For non-broadcast multiaccess networks (NBMA) such as Frame Relay with a partial mesh using point-to-multipoint circuits, this rule can cause significant problems. Since the partial mesh is on the same subnet, BGP applies the rule of not changing the next hop address, rendering invalid routes in certain topologies. This is one case where this command is necessary to solve a problem.

Usage Examples

The following example enables next-hop-self:

ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#next-hop-self
password <password>

Use the password command to enable MD5 password authentication on TCP. Use the no form of this command to disable authentication.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;password&gt;</th>
<th>Specifies the password string to be used for authentication. The password is case-sensitive and must not exceed 80 characters.</th>
</tr>
</thead>
</table>

Default Values

By default, authentication is disabled.

Functional Notes

Authentication must be configured on both peers using the same password. Every BGP TCP segment sent is authenticated. Configuring authentication causes an existing session to be torn down and re-established using the currently specified authentication.

Usage Examples

The following example enables authentication for this BGP neighbor and sets a password of user1:

ProCurve(config)#router bgp 65000
ProCurve(config-router)#neighbor 172.24.3.192
ProCurve(config-router)#password user1
prefix-list <listname> [in | out]

Use the prefix-list command to assign a predefined prefix list to a BGP neighbor. The list is then used to filter BGP route updates received and/or sent from/by the specified peer. Use the no form of this command to discontinue use of the prefix list.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listname&gt;</td>
<td>Assigns a prefix list to this BGP neighbor.</td>
</tr>
<tr>
<td>in</td>
<td>Specifies the filtering of all inbound BGP route updates received from the specified peer.</td>
</tr>
<tr>
<td>out</td>
<td>Specifies the filtering of all outbound BGP route updates being sent to the specified peer.</td>
</tr>
</tbody>
</table>

Default Values

By default, no prefix lists are specified for filtering.

Functional Notes

Before they can be assigned to a BGP neighbor, prefix lists must first be defined using the ip prefix-list command. See ip prefix-list <listname> description <"text"> on page 409 for more information.

Usage Examples

The following example uses the MyList prefix list to filter all BGP updates received from the specified peer:

```
ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#prefix-list MyList in
```
remote-as <as-number>

Use the remote-as command to specify the BGP autonomous system (AS) to which the neighbor belongs, adding an entry to the BGP neighbor table. Use the no form of this command to return to default settings.

Syntax Description

<as-number> Specifies the AS number. This number must be different from the AS number of the local router (which is defined using the router bgp command). Range: 1 to 65,535. See router bgp <AS number> on page 456 for more information.

Default Values

By default, no BGP neighbors are defined.

Usage Examples

The following example configures a remote AS number of 200 for this neighbor:

ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#remote-as 200
route-map <map-name> [in | out]

Use the `route-map` command to assign a route map to this BGP neighbor. The route map is then used to filter or modify inbound and/or outbound BGP route updates. Use the `no` form of this command to return to default settings.

**Syntax Description**

- `<map-name>` Assigns a route map to this BGP neighbor.
- `in` Specifies the filtering/modification of all inbound BGP route updates.
- `out` Specifies the filtering/modification of all outbound BGP route updates.

**Default Values**

By default, no route map is assigned.

**Functional Notes**

Before a route map can be assigned to a BGP neighbor, it must first be defined using the `route-map` command. See `route-map <map-name> [permit | deny] <sequence number>` on page 455 for more information.

**Usage Examples**

The following example assigns a route map to this neighbor for outbound filtering:

ProCurve(config)#router bgp 65000  
ProCurve(config-bgp)#neighbor 172.24.3.192  
ProCurve(config-bgp-neighbor)#route-map MapName out
send-community standard

Use the send-community standard command to insert a standard BGP community attribute into all outgoing route updates for this neighbor. Use the no form of this command to return to default settings.

Syntax Description

No subcommands.

Default Values

By default, this command is disabled.

Usage Examples

The following example inserts a standard BGP community attribute into all outgoing route updates for the specified neighbor:

ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#send-community standard
soft-reconfiguration inbound

Use the **soft-reconfiguration inbound** command to enable this unit to store BGP updates for the specified neighbor. Use the **no** form of this command to return to default settings.

**Syntax Description**

No subcommands.

**Default Values**

By default, this command is enabled.

**Functional Notes**

BGP updates are stored prior to filtering, thus allowing the **clear ip bgp soft** command to be used in the absence of route refresh (RFC2918) capability. This command affects all neighbors. See **clear ip bgp [ * | <as-number> | <ip address>] [in | out | soft]** on page 35 for more information.

**Usage Examples**

The following example enables the unit to store BGP updates for the specified neighbor:

ProCurve(config)#**router bgp 65000**
ProCurve(config-bgp)#**neighbor 172.24.3.192**
ProCurve(config-bgp-neighbor)#**soft-reconfiguration inbound**
update-source <interface>

Use the update-source command to specify which virtual interface’s IP address will be used as the source IP address for the BGP TCP connection (when connecting to this peer). Use the no form of this command to return to default settings.

Syntax Description

<interface> Specifies the interface ID (e.g., loopback 1) of the virtual interface to be used as the source IP address.

Default Values

By default, the outbound interface's IP address is used for BGP updates.

Functional Notes

This is most often configured as a loopback interface that is reachable by the peer router. The peer will specify this address in its neighbor commands for this router.

Usage Examples

The following example configures the loopback 1 interface as the source IP:

ProCurve(config)#router bgp 65000
ProCurve(config-bgp)#neighbor 172.24.3.192
ProCurve(config-bgp-neighbor)#update-source loopback 1
COMMUNITY LIST COMMAND SET

To activate the Community List Configuration mode, enter the `ip community-list` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#ip community-list listname
ProCurve(config-comm-list)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `deny [<community-number> | internet | local-as | no-export | no-advertise]` on page 1233
- `permit [<community-number> | internet | local-as | no-export | no-advertise]` on page 1234
deny [\textit{community-number}] | internet | local-as | no-export | no-advertise

Use the \texttt{deny} command to add an entry to the community list that denies BGP routes containing the specified community number in the community attribute. Use the \texttt{no} form of this command to remove the statement from the community list.

\begin{table}[h]
\centering
\begin{tabular}{|l|p{0.7\textwidth}|}
\hline
\textit{community-number} & Denies routes that contain this value in their community attribute. This is a numeric value that can be an integer from 1 to 4,294,967,295 or string in the form "aa:nn", where the value of "aa" is the AS number and the value of "nn" is the community number. Multiple community-number parameters can be present in the command. \\
 internet & Denies routes that contain this value in their community attribute. This represents the well-known reserved community number for the INTERNET community. \\
 local-as & Denies routes that contain this value in their community attribute. This represents the well-known reserved community number for NO_EXPORT_SUBCONFED. Routes containing this attribute should not be advertised to external BGP peers. \\
 no-export & Denies routes containing this value in their community attribute. This represents the well-known reserved community number for NO_EXPORT. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary. \\
 no-advertise & Denies routes containing this value in their community attribute. This represents the well-known reserved community number for NO_ADVERTISE. Routes containing this attribute should not be advertised to any BGP peer. \\
\hline
\end{tabular}
\end{table}

\textbf{Default Values}

No default value necessary for this command.

\textbf{Usage Examples}

The following example creates a community list (\texttt{MyList}) to deny BGP routes that have the INTERNET community number in their community attribute:

```
ProCurve(config)#ip community-list MyList
ProCurve(config-comm-list)#deny internet
```
**permit [<community-number> | internet | local-as | no-export | no-advertise]**

Use the *permit* command to add an entry to the community list that allows only BGP routes containing the specified community number in the community attribute. Use the *no* form of this command to remove the statement from the community list.

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;community-number&gt;</code></td>
<td>Permits routes that contain this value in their community attribute. This is a numeric value that can be an integer from 1 to 4,294,967,295 or string in the form “aa:nn”, where the value of “aa” is the AS number and the value of “nn” is the community number. Multiple community-number parameters can be present in the command.</td>
</tr>
<tr>
<td>internet</td>
<td>Permits routes that contain this value in their community attribute. This represents the well-known reserved community number for the INTERNET community.</td>
</tr>
<tr>
<td>local-as</td>
<td>Permits routes that contain this value in their community attribute. This represents the well-known reserved community number for NO_EXPORT_SUBCONFED. Routes containing this attribute should not be advertised to external BGP peers.</td>
</tr>
<tr>
<td>no-export</td>
<td>Permits routes containing this value in their community attribute. This represents the well-known reserved community number for NO_EXPORT. Routes containing this attribute should not be advertised to BGP peers outside a confederation boundary.</td>
</tr>
<tr>
<td>no-advertise</td>
<td>Permits routes containing this value in their community attribute. This represents the well-known reserved community number for NO_ADVERTISE. Routes containing this attribute should not be advertised to any BGP peer.</td>
</tr>
</tbody>
</table>

### Default Values

No default value necessary for this command.

### Usage Examples

The following example creates a community list (*MyList*) to permit BGP routes that have the INTERNET community number in their community attribute:

```plaintext
ProCurve(config)#ip community-list MyList
ProCurve(config-comm-list)#permit internet
```
ROUTER (OSPF) CONFIGURATION COMMAND SET

To activate the Router (OSPF) Configuration mode, enter the `router ospf` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#router ospf
ProCurve(config-ospf)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

```
bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port> on page 1304
do on page 1308
end on page 1309
exit on page 1310
ping <address> on page 1311
```

All other commands for this command set are described in this section in alphabetical order.

```
area <area id> default-cost <value> on page 1236
area <area id> range <ip address> <network mask> [advertise | not-advertise] on page 1237
area <area id> stub [no-summary] on page 1238
auto-cost reference-bandwidth <rate> on page 1239
default-information-originate [always] [metric <value>] [metric-type type] on page 1240
default-metric <value> on page 1241
maximum paths <number> on page 1242
network <ip address> <wildcard> area <area id> on page 1243
redistribute connected [metric <value>] [metric-type <type>] [subnets] on page 1244
redistribute [rip] [static] [metric <value>] [metric-type <type>] [subnets] on page 1245
summary-address <address> <mask> [prefix mask] not-advertise on page 1246
timers lsa-group-pacing <seconds> on page 1247
timers spf <delay> <hold> on page 1248
```
area <area id> default-cost <value>

Use the area default-cost command to assign a cost of the default summary route sent into a stub area or not-so-stubby-area (NSSA). Use the no form of this command to delete the assigned cost.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;area id&gt;</td>
<td>Specifies identifier for this area. Enter as an integer (range: 0 to 4,294,967,295) or an IP address &lt;A.B.C.D&gt;.</td>
</tr>
<tr>
<td>&lt;value&gt;</td>
<td>Specifies default summary route cost. Range: 0 to 166,777,214.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, the summary route cost is set to 0. There is no default for the area ID.

**Usage Examples**

The following example defines a default cost of 85 to a specific area:

```
ProCurve(config)#router ospf
ProCurve(config-ospf)#area 192.168.72.0 default-cost 85
```
area <area id> range <ip address> <network mask> [advertise | not-advertise]

Use the area range command to configure area route summarizations and to determine whether an address range is advertised to the networks.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;area id&gt;</td>
<td>Specifies identifier for this area. Enter as an integer (range: 0 to 4,294,967,295) or an IP address &lt;A.B.C.D&gt;.</td>
</tr>
<tr>
<td>&lt;ip address&gt;</td>
<td>The IP address of the advertised summary route.</td>
</tr>
<tr>
<td>&lt;network mask&gt;</td>
<td>The mask of the advertised summary route.</td>
</tr>
<tr>
<td>advertise</td>
<td>The specified address range will be advertised to other networks.</td>
</tr>
<tr>
<td>not-advertise</td>
<td>The specified address range will not be advertised to other networks.</td>
</tr>
</tbody>
</table>

Default Values

By default, OSPF is not enabled.

Usage Examples

The following example defines an address range for a specific area and allows the unit to advertise this range to other networks.

ProCurve(config)#router ospf
ProCurve(config-ospf)#area 10.0.0.0 range 10.0.0.0 255.0.0.0 advertise
area <area id> stub [no-summary]

Use the area stub command to configure an area as a stub area. Use the no form of this command to disable stub-designation for areas defined as stubs using this command.

Syntax Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;area id&gt;</td>
<td>Specifies identifier for this stub area. Enter as an integer (range: 0 to 4,294,967,295) or an IP address &lt;A.B.C.D&gt;.</td>
</tr>
<tr>
<td>no-summary</td>
<td>Optional. Use this optional keyword to designate the area as a total stub area. No summary link advertisements will be sent by the ABR into the stub area.</td>
</tr>
</tbody>
</table>

Default Values

By default, OSPF is not enabled.

Technology Review

It is important to coordinate configuration of all routers and access servers in the stub area. The area stub command must be configured for each of those pieces of equipment. Use the area router configuration command with the area default-cost command to specify the cost of a default internal router sent into a stub area by an ABR. See area <area id> default-cost <value> on page 1236 for related information.

Usage Examples

The following example configures area 2 as a stub area.

ProCurve(config)#router ospf
ProCurve(config-ospf)#area 2 stub
auto-cost reference-bandwidth <rate>

Use the auto-cost reference-bandwidth command to assign a different interface cost to an interface. It may be necessary to assign a higher number to high-bandwidth links. This value is used in OSPF metric calculations.

**Syntax Description**

| <rate> | Sets the default reference-bandwidth rate (range: 1 to 4,294,967 Mbps). |

**Default Values**

By default, the rate is set to 100.

**Usage Examples**

The following example sets the auto cost reference-bandwidth to 1000 Mbps:

ProCurve(config)#router ospf
ProCurve(config-ospf)#auto-cost reference-bandwidth 1000
Use the `default-information-originate` command to cause an ASBR to generate a default route. It must have its own default route before it generates one unless the `always` keyword is used.

**Syntax Description**

<table>
<thead>
<tr>
<th>Clause</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>always</code></td>
<td>Specifies to always advertise default route.</td>
</tr>
<tr>
<td><code>metric &lt;value&gt;</code></td>
<td>Configures metric value (range is 0 to 16,777,214).</td>
</tr>
<tr>
<td><code>metric type &lt;type&gt;</code></td>
<td>Configures metric type (1 or 2).</td>
</tr>
</tbody>
</table>

**Default Values**

<table>
<thead>
<tr>
<th>Clause</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>metric &lt;value&gt;</code></td>
<td>10</td>
</tr>
<tr>
<td><code>metric type &lt;type&gt;</code></td>
<td>2</td>
</tr>
</tbody>
</table>

**Usage Examples**

The following example configures a router to always advertise default routes, and assigns the default routes a metric value of 10000 and a metric type of 2.

```
ProCurve(config)#router ospf
ProCurve(config-ospf)#default-information-originate always metric 10000 metric-type 2
```
**default-metric <value>**

Use the `default-metric` command to set a metric value for redistributed routes.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;value&gt;</code></td>
<td>Sets the default metric value (range: 0 to 4,294,967,295).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this value is set at 20.

**Functional Notes**

The metric value defined using the `redistribute` command overrides the `default-metric` command’s metric setting. See `redistribute ospf [metric <value>]` on page 1260 and `redistribute static [metric <value>]` on page 1261 for related information.

**Usage Examples**

The following example shows a router using both RIP and OSPF routing protocols. The example advertises RIP-derived routes using the OSPF protocol and assigns the RIP-derived routes an OSPF metric of 10.

```
ProCurve(config)#router ospf
ProCurve(config-ospf)#default-metric 10
ProCurve(config-ospf)#redistribute rip
```
maximum paths <number>

Use the maximum paths command to set the maximum number of multipath routes to advertise to the route table via OSPF.

Syntax Description

| <number> | Specifies the number of routes OSPF can insert into the route table. Valid range: 1 to 6. |

Default Values

By default, maximum paths is set to 4.

Usage Examples

The following example sets the maximum number of multipath routes OSPF can insert in the route table to 5.

ProCurve(config)#router ospf
ProCurve(config-ospf)#maximum paths 5
network <ip address> <wildcard> area <area id>

Use the network area command to enable routing on an IP stack and to define area IDs for the interfaces on which OSPF will run. Use the no form of this command to disable OSPF routing for interfaces defined using this command.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ip address&gt;</td>
<td>Network address &lt;A.B.C.D&gt;.</td>
</tr>
<tr>
<td>&lt;wildcard&gt;</td>
<td>The wildcard mask is in an IP-address-type format and includes &quot;don’t care&quot; bits.</td>
</tr>
<tr>
<td>&lt;area id&gt;</td>
<td>Specifies identifier for this area. Enter as an integer (range: 0 to 4,294,967,295) or an IP address &lt;A.B.C.D&gt;.</td>
</tr>
</tbody>
</table>

Default Values

No default values required for this command.

Technology Review

In order for OSPF to operate on an interface, the primary address for the interface must be included in the network area command. Assigning an interface to an OSPF area is done using the network area command. There is no limit to the number of network area commands used on a router. If the address ranges defined for different areas overlap, the first area in the network area command list is used and all other overlapping portions are disregarded. Try to avoid overlapping to avoid complications.

Usage Examples

In the following example, the OSPF routing process is enabled and two OSPF areas are defined:

ProCurve(config)#router ospf
ProCurve(config-ospf)#network 192.168.10.101 0.0.0.255 area 0
ProCurve(config-ospf)#network 10.0.0.0 0.255.255.255 area 10.0.0.0
**redistribute connected [metric <value> | metric-type <type> | subnets]**

Use the `redistribute connected` command to advertise routes from one protocol to another. Using the `connected` keyword allows the advertisement of connected routes into the OSPF routing protocol. This will advertise all connected routes on OSPF-enabled interfaces. It does not enable OSPF on all interfaces. Use the `no` form of this command to disable the propagation of the specified route type.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric &lt;value&gt;</td>
<td>Optional. Specifies a metric value to be carried from one OSPF process to the next (if no other value is specified).</td>
</tr>
<tr>
<td>metric-type &lt;type&gt;</td>
<td>Optional. Specifies a type 1 or type 2 external route as the external link type. If not specified, the default is 2.</td>
</tr>
<tr>
<td>subnets</td>
<td>Optional. Specifies subnet redistribution when redistributing routes into OSPF.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is disabled.

**Functional Notes**

Redistributing connected routes imports those routes into OSPF without the interfaces in question actually participating in OSPF. The connected routes imported this way are not covered by a network command and therefore do not send/receive OSPF traffic.

**Usage Examples**

The following example imports connected routes into OSPF:

```
ProCurve(config)#router ospf
ProCurve(config-ospf)#redistribute connected
```
redistribute [rip | static] [metric <value> | metric-type <type> | subnets]

Use the `redistribute` command to advertise routes from one protocol to another, regardless of the routing protocol implemented on the routing domain. Using the `rip` keyword allows the propagation of RIP routes into OSPF. Using the `static` keyword allows the advertisement of static routes into the OSPF routing protocol. This will advertise all static routes on OSPF-enabled interfaces. It does not enable OSPF on all interfaces. Use the `no` form of this command to disable the propagation of the specified route type.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rip</td>
<td>Specifies advertising RIP routes using OSPF.</td>
</tr>
<tr>
<td>static</td>
<td>Specifies advertising static routes using OSPF.</td>
</tr>
<tr>
<td>metric &lt;value&gt;</td>
<td>Optional. Specifies a metric value to be carried from one OSPF process to the next (if no other value is specified).</td>
</tr>
<tr>
<td>metric-type &lt;type&gt;</td>
<td>Optional. Specifies a type 1 or type 2 external route as the external link type. If not specified, the default is 2.</td>
</tr>
<tr>
<td>subnets</td>
<td>Optional. Specifies subnet redistribution when redistributing routes into OSPF.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is disabled.

**Functional Notes**

Redistributing routes imports routes (routes learned through RIP or static routes) into OSPF without the interfaces in question actually participating in OSPF. RIP and static routes imported this way are not covered by a network command and therefore do not send/receive OSPF traffic.

**Usage Examples**

The following example imports RIP routes into OSPF:

```
ProCurve(config)#router ospf
ProCurve(config-ospf)#redistribute rip
```

The following example imports static routes into OSPF:

```
ProCurve(config)#router ospf
ProCurve(config-ospf)#redistribute static
```
**summary-address** `<address> <mask | prefix mask> not-advertise`

Use this command to control address summarization of routes that are redistributed into OSPF from other sources (e.g., RIP-to-OSPF, static-to-OSPF, etc.). The **not-advertise** option causes suppression of routes that match the specified mask/prefix mask pair.

**Syntax Description**

- `<address>`: Specifies IP address or Prefix A.B.C.D.
- `<mask | prefix mask>`: Routes matching this mask/prefix mask pair will be suppressed if the **not-advertise** command is enabled.
- **not advertise**: Optional. Causes suppression of routes that match the specified mask/prefix mask pair.

**Default Values**

By default, this command is disabled.

**Usage Examples**

The following example suppresses advertisement of the routes which match the specified address/mask:

ProCurve(config)#**router ospf**  
ProCurve(config-ospf)#**summary-address 10.0.0.0 255.0.0.0 not-advertise**
timers lsa-group-pacing <seconds>

Use the timers lsa-group-pacing command to change the link state advertisement (LSA) refresh interval.

Syntax Description

- `<seconds>` Sets the LSA refresh interval in seconds (range: 10 to 1,800).

Default Values

- By default, this value is set at 240 seconds.

Usage Examples

- The following example sets the refresh interval for six minutes:

  ProCurve(config)#router ospf
  ProCurve(config-ospf)#timers lsa-group-pacing 360
timers spf <delay> <hold>

Use the timers spf command to configure the shortest path first (SPF) calculation and hold intervals.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;delay&gt;</th>
<th>Specifies time in seconds between OSPF’s receipt of topology changes and the beginning of SPF calculations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;hold&gt;</td>
<td>Specifies time in seconds between consecutive SPF calculations. Range: 10 to 1800 seconds.</td>
</tr>
</tbody>
</table>

Default Values

<table>
<thead>
<tr>
<th>&lt;delay&gt;</th>
<th>5 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;hold&gt;</td>
<td>10 seconds</td>
</tr>
</tbody>
</table>

Usage Examples

The following example defines a delay of 10 seconds and a hold-time of 30 seconds:

ProCurve(config)#router ospf
ProCurve(config-ospf)#timers spf 10 30
ROUTER (PIM SPARSE) CONFIGURATION COMMAND SET

To activate the Router (PIM Sparse) Configuration mode, enter the `router pim-sparse` command at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# router pim-sparse
ProCurve(config-pim-sparse)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `join-prune-msg-interval <seconds>` on page 1250
- `rp-address <ip address> access-group <access-list-name>` on page 1251
- `spt-threshold <packets> infinity` on page 1252
join-prune-msg-interval <seconds>

Use the join-prune-msg-interval command to set a timing rate for PIM sparse join/prune messages. Use the no form of this command to return to the default setting.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;seconds&gt;</th>
<th>Specifies the PIM sparse join/prune message interval.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Valid range: 10 to 65,534 seconds.</td>
</tr>
</tbody>
</table>

Default Values

By default, the message interval is set to 60 seconds.

Usage Examples

The following example sets the interval for 50 seconds:

ProCurve(config)#router pim-sparse
ProCurve(config-pim-sparse)#join-prune-msg-interval 50
rp-address <ip address> access-group <access-list-name>

Use the `rp-address` command to specify a static IP address for the Rendezvous Point (RP) router. The `access-group` keyword is used to limit the multicast group addresses to which the RP applies. Use the `no` form of this command to remove a static IP address for the RIP router.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ip address&gt;</code></td>
<td>Specifies the IP address for the RP.</td>
</tr>
<tr>
<td><code>access-group &lt;access-list-name&gt;</code></td>
<td>Optional. Specifies the particular access group to which the RP applies.</td>
</tr>
</tbody>
</table>

**Default Values**

No default necessary for this command.

**Functional Notes**

The `access-group` keyword is used to limit the multicast group addresses to which the RP applies. If more than one RP is configured for a given multicast group address, then a hash algorithm determines the appropriate hierarchy (see below). The results of the hash algorithm can be seen with the `show ip pim-sparse rp-map` command.

The hash algorithm is defined in RFC 2117 section 3.7 as follows:

For each RP address C(i) in the RP-Set, whose Group-prefix covers G, compute a value:

\[
\text{Value}(G,M,C(i)) = (1103515245 \times ((1103515245 \times (G \& M) + 12345) \text{ XOR } C(i)) + 12345) \mod 2^{31}
\]

where M is a hash-mask included in Bootstrap messages. This hash-mask allows a small number of consecutive groups (e.g., 4) to always hash to the same RP. For instance, hierarchically-encoded data can be sent on consecutive group addresses to get the same delay and fate-sharing characteristics.

The candidate with the highest resulting value is then chosen as the RP for that group, and its identity and hash value are stored with the entry created.

Ties between C-RPs having the same hash value, are broken in advantage of the highest address.

**Usage Examples**

The following example specifies an IP address of 172.22.5.100 for the RP:

ProCurve(config)#router pim-sparse
ProCurve(config-pim-sparse)#rp-address 172.22.5.100
spt-threshold <packets> infinity

Use the spt-threshold command to change the PIM Sparse Shortest Path Tree (SPT) threshold, which specifies the number of packets the router sends using the rendezvous point (RP) before switching to the SPT.

Syntax Description

| <packets>   | Specifies the number of packets the router sends using the RP before switching to the SPT. Valid range: 1 to 4,294,967,295. |
| infinity    | Causes all sources to use the shared RP tree. |

Default Values

By default, the SPT threshold is set to 1 packet.

Usage Examples

The following example sets the SPT threshold at five packets:

ProCurve(config)#router pim-sparse
ProCurve(config-pim-sparse)#spt-threshold 5
ROUTER (RIP) CONFIGURATION COMMAND SET

To activate the Router (RIP) Configuration mode, enter the `router rip` command at the Global Configuration mode prompt. For example:

ProCurve>enable  
ProCurve#configure terminal  
ProCurve(config)#router rip  
ProCurve(config-rip)#

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `auto-summary` on page 1254
- `default-metric <value>` on page 1255
- `distribute-list <access list> [in | out] <interface type> <slot/port>` on page 1256
- `network <address> <subnet mask>` on page 1257
- `passive-interface <interface>` on page 1258
- `redistribute connected [metric <value>]` on page 1259
- `redistribute ospf [metric <value>]` on page 1260
- `redistribute static [metric <value>]` on page 1261
- `timeout-timer <seconds>` on page 1262
- `update-timer <seconds>` on page 1263
- `version [1 | 2]` on page 1264
auto-summary

Use the auto-summary command to have RIP version 2 summarize subnets to the classful boundaries. Use the no form of this command to disable this summarization.

Syntax Description
No subcommands.

Default Values
By default, auto-summary is disabled.

Functional Notes
Use this command if you are subdividing a classful network into many subnets and these subnets are to be advertised over a slow link (64K or less) to a router that can only reach the classful network via the router you are configuring.

Usage Examples
The following example configures the router to not automatically summarize network numbers:

ProCurve(config)#router rip
ProCurve(config-rip)#no auto-summary
**default-metric** <value>

Use the `default-metric` command to set the default metric value for the RIP routing protocol. Use the `no` form of this command to return to the default settings.

**Syntax Description**

| <value> | Sets the default metric value (range: 1 to 4,294,967,295 Mbps). |

**Default Values**

By default, this value is set at 0.

**Functional Notes**

The metric value defined using the `redistribute` command overrides the `default-metric` command's metric setting. See `redistribute ospf [metric <value>]` on page 1260 and `redistribute static [metric <value>]` on page 1261 for related information.

**Usage Examples**

The following example shows a router using both RIP and OSPF routing protocols. The example advertises OSPF-derived routes using the RIP protocol and assigns the OSPF-derived routes a RIP metric of 10.

ProCurve(config)#router rip
ProCurve(config)#default-metric 10
ProCurve(config-rip)#redistribute ospf
**distribute-list** `<access list> [in | out] <interface type> <slot/port>`

Use the `distribute-list` command to add RIP filtering functionality by assigning inbound and outbound access lists on either a per-interface or global basis. Only one inbound/outbound pair of access lists can be configured for a particular interface. Use the `no` form of this command to disable the filtering.

**Syntax Description**

- `<access list>`: Specifies an access list name. This is a standard IP access control list (ACL) against which the contents of the incoming/outgoing routing updates are matched.
- `in`: Applies RIP filtering to inbound data.
- `out`: Applies RIP filtering to outbound data.
- `<interface type> <slot/port>`: Optional. Specifies the interface in which to apply the ACL. Type `distribute-list xxxx in ?` for a complete list of applicable interfaces.

**Default Values**

By default, distribute-list filtering is disabled.

**Usage Examples**

The following example will filter out all RIP network advertisements received via Ethernet interface 0/1 with the exception of the 10.10.10.0 network:

```
ProCurve>(config)#router rip
ProCurve>(config-rip)#version 2
ProCurve>(config-rip)#network 192.168.1.0 255.255.255.0
ProCurve>(config-rip)#distribute-list list_1 in eth 0/1
ProCurve>(config-rip)#exit
ProCurve>(config)#ip access-list standard list_1
ProCurve>(config-std-nacl)#permit 10.10.10.0 0.0.0.255
```
network <address> <subnet mask>

Use the network command to enable RIP on the specified network. Use the no form of this command to remove a network from the list.

Syntax Description

<address> Specifies IP address of the network on which RIP will be enabled.
<subnet mask> Specifies subnet mask that corresponds to the entered IP address.

Default Values

By default, RIP is not enabled.

Functional Notes

The SROS will only allow processing (sending and receiving) RIP messages on interfaces with IP addresses that are contained in the networks listed using this command. All RIP messages received on interfaces not listed using this command will be discarded. To allow for receiving and participating in RIP but not for transmitting, use the passive-interface command (refer to passive-interface <interface> on page 1258).

Usage Examples

The following example enables RIP on the 192.168.72.252/30, 192.168.2.0/24, and 10.200.0.0/16 networks:

ProCurve(config)#router rip
ProCurve(config-rip)#network 192.168.72.252 255.255.255.252
ProCurve(config-rip)#network 192.168.2.0 255.255.255.0
ProCurve(config-rip)#network 10.200.0.0 255.255.0.0
passive-interface <interface>

Use the passive-interface command to disable the transmission of routing updates on the specified interface. Use the no form of this command to enable the transmission of routing updates on an interface.

**Syntax Description**

<interface> Specifies the interface that will not transmit routing updates.

**Default Values**

By default, RIP is not enabled.

**Functional Notes**

All routing updates received on that interface will still be processed (and advertised to other interfaces), but no updates will be transmitted to the network connected to the specified interface. Multiple passive-interface commands may be used to create a customized list of interfaces.

**Usage Examples**

The following example disables routing updates on the Frame Relay link (labeled 1.17) and the PPP link (labeled 1):

ProCurve(config)#router rip
ProCurve(config-rip)#passive-interface frame-relay 1.17
ProCurve(config-rip)#passive-interface ppp 1
redistribute connected [metric <value>]

Use the redistribute connected command to pass routes from one network to another, regardless of the routing protocol implemented on the routing domain. Using the connected keyword allows the propagation of routes for connected devices to the network by transmitting RIP packets. Use the no form of this command to disable the propagation of the specified route type.

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric &lt;value&gt;</td>
<td>Optional. Specifies the hop count to use when advertising redistributed OSPF routes using the RIP protocol.</td>
</tr>
</tbody>
</table>

Default Values

By default, RIP is not enabled.

Functional Notes

Redistributing connected routes imports those routes into RIP without the interfaces in question actually participating in RIP. The connected routes imported this way are not covered by a network command and therefore do not send/receive RIP traffic.

Usage Examples

The following example passes the connected routes found in the route table to other networks running the RIP routing protocol:

```
ProCurve(config)#router rip
ProCurve(config-rip)#redistribute connected
```
**redistribute ospf [metric <value>]**

Use the `redistribute ospf` command to advertise routes from one protocol to another, regardless of the routing protocol implemented on the routing domain. Using the `ospf` keyword allows the propagation of known OSPF routes to the network by transmitting RIP packets. Use the `no` form of this command to disable the propagation of the specified route type.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>metric &lt;value&gt;</code></td>
<td>Optional. Specifies the hop count to use when advertising redistributed OSPF routes using the RIP protocol.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is disabled.

**Functional Notes**

Redistributing OSPF routes imports those routes into RIP without the interfaces in question actually participating in RIP. The OSPF routes imported this way are not covered by a network command and therefore do not send/receive RIP traffic.

If `redistribute ospf` is enabled and no metric value is specified, the value defaults to 0. The metric value defined using the `redistribute ospf metric` command overrides the `default-metric` command’s metric setting. Refer to the section `default-metric <value>` on page 1255 for more information.

**Usage Examples**

The following example imports OSPF routes into RIP:

```
ProCurve>(config)#router rip
ProCurve>(config-rip)#redistribute ospf
```
redistribute static [metric <value>]

Use the redistribute static command to pass routes from one network to another, regardless of the routing protocol implemented on the routing domain. Using the static keyword allows the propagation of static routes to the network by transmitting RIP packets. Use the no form of this command to disable the propagation of the specified route type.

**Note**

The gateway network for the static route must participate in RIP by using the network command for the gateway network.

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric &lt;value&gt;</td>
<td>Optional. Specifies the hop count to use for advertising redistributed OSPF routes in RIP.</td>
</tr>
</tbody>
</table>

**Default Values**

By default, this command is disabled.

**Functional Notes**

Redistributing static routes allows other network devices to learn about paths (not compatible with their system) without requiring manual input to each device on the network.

**Usage Examples**

The following example passes the static routes found in the route table to other networks running the RIP routing protocol:

```
ProCurve>(config)#router rip
ProCurve>(config-rip)#redistribute static
```
**timeout-timer <seconds>**

Use the `timeout-timer` command to set the timeout timer value for a route when it is learned via RIP. Each time a RIP update for that route is received, the timeout timer is reset to this value. If no updates for that route are received in the specified number of seconds and the timeout timer expires, the route is considered invalid, and it will be removed from the route table. Use the `no` form of this command to return to the default settings.

**Syntax Description**

```
<seconds>          Sets the timeout-timer value. Valid range: 5 to 4294967295 seconds.
```

**Default Values**

By default, this value is set at 180 seconds.

**Functional Notes**

Note that the `timeout-timer` value cannot be set to a value less than the `update-timer` value. It is recommended that this timer be set to a value that is three times the value of the `update-timer` (see `update-timer <seconds>` on page 1263).

**Usage Examples**

The following example configures the router to mark routes invalid if no RIP updates for those routes are received within 120 seconds.

```
ProCurve(config)#router rip
ProCurve(config-rip)#timeout-timer 120
```
**update-timer <seconds>**

Use the `update-timer` command to set the value of the RIP update interval timer. The RIP update interval is the number of seconds which must elapse between RIP update packet transmissions. Use the `no` form of this command to return to the default settings.

**Syntax Description**

| `<seconds>` | Specifies the number of seconds allowed to elapse between RIP update packet transmissions. Valid range: 5 to 4,294,967,295 seconds. |

**Default Values**

By default, this value is set at 30 seconds.

**Functional Notes**

Note that the `timeout-timer` value cannot be set to a value less than the `update-timer` value. It is recommended that the `timeout-timer` be set to a value that is three times the value of the `update-timer`. (See `default-metric <value>` on page 1255 for more information.)

**Usage Examples**

The following example sets the rate at which RIP update messages are transmitted from the router to 20 seconds.

```
ProCurve(config)#router rip
ProCurve(config-rip)#update-timer 20
```
version [1 | 2]

Use the `version` command to specify (globally) the Routing Information Protocol (RIP) version used on all IP interfaces. This global configuration is overridden using the configuration commands `ip rip send version` and `ip rip receive version`. Use the `no` form of this command to return to the default value.

**Syntax Description**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RIP version 1</td>
</tr>
<tr>
<td>2</td>
<td>RIP version 2</td>
</tr>
</tbody>
</table>

**Default Values**

By default, RIP is not enabled.

**Usage Examples**

The following example specifies RIP version 2 as the global RIP version:

```
ProCurve(config)#router rip
ProCurve(config-rip)#version 2
```
QUALITY OF SERVICE (QoS) MAP COMMANDS

A QoS policy is defined using a QoS map in the CLI. The QoS map is a named list with sequenced entries. An entry contains a single match reference and one or more actions (priority, set, or both). To activate the QoS Command Set (which allows you to create and/or edit a map), enter a valid version of the QoS command at the Global Configuration mode prompt. Multiple map entries for the same QoS map are differentiated by a sequence number. The sequence number is used to assign match order.

Once created, a QoS map must be applied to an interface (using the `qos-policy out <map-name>` command) in order to actively process traffic. Any traffic for the interface that is not sent to the priority queue is sent using the default queuing method for the interface (such as weighted fair queuing).

For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#qos map VOICEMAP 10
ProCurve(config-qos-map)#match precedence 5
ProCurve(config-qos-map)#priority 512
ProCurve(config-qos-map)#exit
ProCurve(config)#interface fr 1
ProCurve(config-fr 1)#qos-policy out VOICEMAP
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `bandwidth [percent <value> | remaining percent <value> | <value>]` on page 1266
- `match` on page 1269
- `priority` on page 1275
- `set dscp <0-63>` on page 1277
- `set precedence <0-7>` on page 1278
bandwidth [percent <value> | remaining percent <value> | <value>]

Use the bandwidth command to specify bandwidth allocation for individual traffic classes (for CBWFQ configurations). Use the no form of this command to remove a configured bandwidth allocation.

Syntax Description

- **percent <value>**: Allocates a minimum bandwidth for a traffic class, specifying the minimum as a percentage of the total interface bandwidth. See Functional Notes for more details.

- **remaining percent <value>**: Allocates a minimum bandwidth for a traffic class, specifying the minimum as a percentage of the total interface bandwidth not allocated to priority classes in the QoS map. See Functional Notes for more details.

- **<value>**: Allocates the minimum bandwidth for a traffic class, specifying the minimum as an absolute bandwidth in kilobits per second (Range: 8 to 2,000,000 kbps).

Default Values

By default, there is no bandwidth allocation configured for a QoS map entry.

Functional Notes

When configuring bandwidth allocations for CBWFQ, there are a few rules that must be obeyed.

1. The units of the bandwidth (kbps, percent, or remaining percent) must be consistent for all class-based entries (using the bandwidth command) in a QoS map set.

2. The total bandwidth between all priority entries and class-based entries (bandwidth) in a QoS map set should not be configured beyond the specified max-reserved-bandwidth (default 75 percent) on the interface that the QoS policy is applied to (using the qos-policy command), or the map will be disabled. Even with the configuration limit, class-based queues can still use more than the max-reserved-bandwidth limitation, up to 100% of the bandwidth, if enough traffic is present.

When the configured QoS map is applied to a physical interface, the SROS displays bandwidth information for the map and the physical interface. For example, the Frame Relay interface (fr 1) has been connected to the E1 interface (e1 1/1) using the bind command. Applying the QoS map (MyMapA) to the Frame Relay interface (fr 1) produces the following status message:

```
2005.08.09 07:28:22 QOS.INTERFACE QOS policy "MyMapA" requires 1288 kbps of bandwidth and 1488 kbps is now available for interface fr 1 -> the QOS policy for this port has been forced ACTIVE.
```

This status message displays the sum total of the bandwidths specified in the QoS map (1288 kbps) and the available interface bandwidth using the total line rate configured on the interface (1488 kbps).

3. Up to 4 class-based entries (bandwidth commands) can be configured in a particular QoS map set. Up to 16 class-based entries can be configured (four entries on four QoS maps).

4. Within a QoS map entry, CBWFQ bandwidth and low latency priority actions are mutually exclusive. However, bandwidth and priority actions may be applied to different entries in the same QoS map.
Determining Bandwidth Entries

When determining the **percent** <value> entry, use the following formula:

\[
\text{Bandwidth} \quad \frac{\text{Line Rate}}{\text{percent value}}
\]

where

- **Bandwidth** minimum amount of bandwidth needed for the traffic (in kbps)
- **Line Rate** total data rate configured on the interface (for example, 8 DS0s (64 kbps per DS0) on a T1 equals a line rate of 512 kbps)

For example, to specify 80 kbps of data on an interface with a total of 512 kbps of available bandwidth, enter the following command:

```
ProCurve(config-qos-map 1)#bandwidth percent 16
```

When determine the **remaining percent** <value> entry, use the following formula:

\[
\text{Bandwidth} \quad \frac{\left[\left(\text{max-reserved-bandwidth}\right)\left(\text{Line Rate}\right)\right] - \text{Priority Traffic}}{\left(\text{max-reserved-bandwidth}\right)\left(\text{Line Rate}\right)}
\]

where

- **Bandwidth** minimum amount of bandwidth needed for the traffic (in kbps)
- **max-reserved-bandwidth** specifies the percentage of the total line rate available for use by QoS
- **Line Rate** total data rate configured on the interface (for example, 8 DS0s (64 kbps per DS0) on a T1 equals a line rate of 512 kbps)
- **Priority Traffic** amount of bandwidth reserved using the **priority** command

For example, to specify 80 kbps of data on an interface with a total of 512 kbps of available bandwidth, 256 kbps reserved (using the **priority** command), and reserving 15% of the bandwidth for routing and L2 protocol traffic (**max-reserved-bandwidth** = 85) enter the following command:

```
ProCurve(config-qos-map 1)#bandwidth remaining percent 45
```
Usage Examples

The following example creates a QoS map with 4 traffic classes (based on IP packet precedence values) and allocates bandwidth to each class:

ProCurve(config)#qos map MyMap 1
ProCurve(config-qos-map)#match precedence 5
ProCurve(config-qos-map)#bandwidth percent 25

ProCurve(config)#qos map MyMap 2
ProCurve(config-qos-map)#match precedence 3
ProCurve(config-qos-map)#bandwidth percent 10

ProCurve(config)#qos map MyMap 3
ProCurve(config-qos-map)#match precedence 2
ProCurve(config-qos-map)#bandwidth percent 10

ProCurve(config)#qos map MyMap 4
ProCurve(config-qos-map)#match precedence 1
ProCurve(config-qos-map)#bandwidth percent 15
**match**

Use the **match** command to specify which traffic should be processed by this QoS map. Possible variations of this command include:

- `match dscp <0-63>`
- `match ip rtp <start><end>`
- `match ip rtp <first port # in range> <last port # in range>`
- `match ip rtp <first port # in range> <last port # in range> all`
- `match list <listname>`
- `match precedence <0-7>`
- `match protocol bridge`
- `match protocol bridge netbeui`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Optional. Specifies matching all UDP port numbers in the specified range (even and odd). Valid only for <strong>ip rtp</strong> matches.</td>
</tr>
<tr>
<td>dscp &lt;0-63&gt;</td>
<td>Matches IP packets with the specified Differentiated Service Code Point (DSCP) value.</td>
</tr>
<tr>
<td>ip rtp &lt;start&gt;&lt;end&gt;</td>
<td>Matches RTP packets with even UDP destination port numbers in the specified range (between start and end).</td>
</tr>
<tr>
<td>list &lt;listname&gt;</td>
<td>Specifies the name of the access list (ACL) you wish to use to match packets for this QoS map. See <strong>ip access-list extended &lt;listname&gt;</strong> on page 288 for more information on creating access lists.</td>
</tr>
<tr>
<td>precedence &lt;0-7&gt;</td>
<td>Matches IP packets with the specified IP precedence value.</td>
</tr>
<tr>
<td>protocol bridge</td>
<td>Matches frames being bridged by the router.</td>
</tr>
<tr>
<td>protocol bridge netbeui</td>
<td>Matches only NetBEUI frames being bridged by the router.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value is necessary for this command.

**Functional Notes**

QoS policies are configured in the SROS CLI to dictate the priority for servicing specified traffic types on a particular interface. QoS policies contain at least one match reference (using the **match** command) and one or more action items (using the **priority**, **bandwidth**, or **set** commands).

The **match** command specifies the criteria used when determining whether incoming traffic is a candidate for the QoS policy action items. Multiple **match** statements can exist within the same QoS policy, allowing a single QoS policy to service various types of traffic.

**Note**

Each listed **match** statement is handled independently by the processor. Entering too many **match** statements in a QoS policy can burden the processor.
For example, consider a network that contains Class A and Class B traffic that each require 25% of the total allocated interface bandwidth.

ProCurve(config)#qos map MyMap 1
ProCurve(config-qos-map)#match list Class_A
ProCurve(config-qos-map)#match list Class_B
ProCurve(config-qos-map)#bandwidth percent 25

Alternately, the following configuration is also valid:

ProCurve(config)#qos map MyMap 1
ProCurve(config-qos-map)#match list Class_A
ProCurve(config-qos-map)#match list Class_B
ProCurve(config-qos-map)#bandwidth percent 25

ProCurve(config)#qos map MyMap 2
ProCurve(config-qos-map)#match list Class_B
ProCurve(config-qos-map)#bandwidth percent 25

To remove a configured match statement, enter the entire match statement with a preceding no. For example, to remove the match statements from the above configured QoS map:

ProCurve(config)#qos map MyMap 1
ProCurve(config-qos-map)#no match list Class_A
and
ProCurve(config)#qos map MyMap 2
ProCurve(config-qos-map)#no match list Class_B

Usage Examples

The following example configures QoS for a network with the following needs:

Reserve 15% of the line rate for routing traffic and L2 protocol traffic (**max-reserved-bandwidth** = 85)
Line Rate = 512 kbps
Guaranteed 256 kbps for Voice
Guaranteed 96 kbps for Class 1
Guaranteed 52 kbps for Class 2

To configure this QoS policy, enter the following QoS map and interface commands:

1. Allocate LLQ Priority voice traffic
ProCurve(config)#qos map MyMap 1
ProCurve(config-qos-map)#match list VOICE
ProCurve(config-qos-map)#priority 256
2. Allocate the CBWFQ data traffic bandwidth for classes 1 and 2

```bash
ProCurve(config)# qos map MyMap 2
ProCurve(config-qos-map)# match list CLASS_1
ProCurve(config-qos-map)# bandwidth 96
```

```bash
ProCurve(config)# qos map MyMap 3
ProCurve(config-qos-map)# match list CLASS_2
ProCurve(config-qos-map)# bandwidth 52
```

3. Specify the reserved bandwidth and apply the map

```bash
ProCurve(config-fr 1)# max-reserved-bandwidth 85
ProCurve(config-fr 1)# qos-policy out MyMap
```

**Technology Review**

RFC791 created a single octet (labeled Type of Service) to help with the difficulty of trying to provide QoS handling in IP networks.

According to RFC791, the Type of Service field contains the following bits:

- **0 - 3**: 3-bit IP Precedence values (0 through 7) are specified as:
  - 111 Network Control Packets
  - 110 Internetwork Control Packets
  - 101 Critical Traffic
  - 100 Flash Override
  - 011 Flash
  - 010 Immediate Servicing
  - 001 Priority Traffic
  - 000 Routine Data

The IP Precedence values provide network routers with information about what kind of traffic is contained in the IP packet. Based on the IP Precedence values, some networks (when supported) can offer special handling to certain packets. In addition, providing IP Precedence values to critical traffic (such as route information) ensures that critical packets will always be delivered regardless of network congestion. This traffic is often critical to network and internetwork operation. In general, the higher the IP Precedence value, the more important the traffic and the better handling it should receive in the network. It is important to remember that not all equipment in the public IP network will be configured to recognize and handle IP Precedence values. While it is a good idea to set the values for critical traffic, it does not guarantee special handling.
In addition to the precedence values, RFC791 specifies bits for delay, throughput, and reliability to help balance the needs of particular traffic types when traveling on the IP network infrastructure. When these bits are set to 0, they are handled with normal operation. When set to 1, each bit specifies premium handling for that parameter. For example, a 1 in the delay position indicates that the traffic is delay sensitive and care should be taken to minimize delay. A 1 in the throughput position indicates that the traffic has higher bandwidth requirements that should be met. A 1 in the reliability position indicates that the traffic is sensitive to delivery issues and care should be taken to ensure proper delivery with all packets of this type. These extra bits are rarely used because they are quite difficult to balance the cost and benefits of each parameter (especially when more than one bit is set to 1).

The Differentiated Services (DiffServ or DS) model was created in RFC2474 and 2475 to build on the original Type of Service field by creating a six-bit sequence (combining the precedence value with the delay, throughput, and reliability bits). This six-bit sequence increased the number of available values from 8 to 64. The DiffServ model introduced a new concept to QoS in the IP network environment: per-hop behaviors (PHBs). The PHB premise is that equipment using the DiffServ model have an agreed upon set of rules (PHB types) for handling certain network traffic. Though the RFC explicitly defines what each PHB should be capable of, it does not restrict vendor-specific implementation of the PHBs. Each vendor is free to decide how their network product implements the various defined PHBs.

According to RFC2474, the DS field contains the following bits:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differentiated Service Code Point</td>
<td>Unused*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* The previously unused bits in the DS field are now used for congestion control and are not discussed in this document.

Equipment following the DiffServ model (DS-compliant nodes) must use the entire six-bit DSCP value to determine the appropriate PHB. The PHBs are defined as the following:

- Default PHB
- Class Selector PHB
- Assured Forwarding PHB (RFC2597)
- Expedited Forwarding PHB (RFC2598)

**Default PHB**

All DS-compliant nodes must provide a Default PHB to offer best-effort forwarding service. For Default PHBs, the DSCP value is 0. Any packet that does not contain a standardized DSCP should be mapped to the Default PHB and handled accordingly.

**Class Selector PHB**

In the Class Selector PHB, the first three bits in the DSCP value are used for backwards compatibility to systems implementing IP precedence. In this scenario, all but the first three bits of the DS field are set to 0. This compatibility requires DS-compliant nodes to provide the same data services as are provided by nodes implementing IP precedence. The following table is a comparison of IP precedence values to their
corresponding DSCP values.

<table>
<thead>
<tr>
<th>IP Precedence Value (bits)</th>
<th>DSCP Value (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (000)</td>
<td>0 (000000)</td>
</tr>
<tr>
<td>1 (001)</td>
<td>8 (001000)</td>
</tr>
<tr>
<td>2 (010)</td>
<td>16 (010000)</td>
</tr>
<tr>
<td>3 (011)</td>
<td>24 (011000)</td>
</tr>
<tr>
<td>4 (100)</td>
<td>32 (100000)</td>
</tr>
<tr>
<td>5 (101)</td>
<td>40 (101000)</td>
</tr>
<tr>
<td>6 (110)</td>
<td>48 (110000)</td>
</tr>
<tr>
<td>7 (111)</td>
<td>56 (111000)</td>
</tr>
</tbody>
</table>

**Assured Forwarding PHB**

The flexibility of DiffServ allows for more developed sub-classes of service within each main class using the last three bits of the DSCP. As defined in RFC2597, the Assured Forwarding PHB creates four main classes of service:

<table>
<thead>
<tr>
<th>Class</th>
<th>DSCP Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF1</td>
<td>001XX0</td>
</tr>
<tr>
<td>AF2</td>
<td>010XX0</td>
</tr>
<tr>
<td>AF3</td>
<td>011XX0</td>
</tr>
<tr>
<td>AF4</td>
<td>100XX0</td>
</tr>
</tbody>
</table>

X indicates a do not care value

The first three bits of the DSCP specify the class and the last bit is always zero. Each class is separated into subclasses using the two remaining bits in the DSCP (bits 3 and 4). The subclasses are divided based on the likelihood that packets in the class are dropped in the event of network congestion. The higher the value for bits 3 and 4, the greater the likelihood that the packets will be dropped.

<table>
<thead>
<tr>
<th>Bit 3</th>
<th>Bit 4</th>
<th>Drop Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Low</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Medium</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>High</td>
</tr>
</tbody>
</table>
The following table lists the Assured Forwarding PHB subclasses and their corresponding DSCP bits and values.

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
<th>DSCP Bits</th>
<th>DSCP Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF1</td>
<td>1</td>
<td>001010</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>001100</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>001110</td>
<td>14</td>
</tr>
<tr>
<td>AF2</td>
<td>1</td>
<td>010010</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>010100</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>010110</td>
<td>22</td>
</tr>
<tr>
<td>AF3</td>
<td>1</td>
<td>011010</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>011100</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>011110</td>
<td>30</td>
</tr>
<tr>
<td>AF4</td>
<td>1</td>
<td>100010</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>100100</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>100110</td>
<td>38</td>
</tr>
</tbody>
</table>

**Expedited Forwarding PHB**

RFC2598 created a new DiffServ PHB intended to provide the best service possible on an IP network. Packets using the Expedited Forwarding PHB markings should be provided service to reduce latency, jitter, dropped packets, and be guaranteed bandwidth during the entire end-to-end transmission journey through the network. The DSCP value for the Expedited Forwarding PHB is 46 (DSCP bits are 101110).
priority

The `priority` command provides a high-priority, low-latency queue, prioritizing this traffic above all others. If no traffic is present in any other queue, priority traffic is allowed to burst up to the interface rate; otherwise, priority traffic above the specified bandwidth is dropped. Use the `no` form of this command to disable this feature. Variations of this command include:

```
priority <bandwidth>
priority <bandwidth> <burst>
priority percent <bandwidth>
priority unlimited
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;bandwidth&gt;</code></td>
<td>Specifies a low latency queue, prioritizing this traffic above all other user traffic in kilobits per second (Range: 8 to 1,000,000). If no traffic is present in any other queue, priority traffic is allowed to burst up to the interface rate. Otherwise, priority traffic above the specified bandwidth will be dropped.</td>
</tr>
<tr>
<td><code>&lt;burst&gt;</code></td>
<td>Optional. Specifies the maximum burst size (in bytes) for traffic in this priority queue. This parameter should be left unconfigured for optimal performance (Range: 3 to 1,000,000).</td>
</tr>
<tr>
<td><code>percent &lt;value&gt;</code></td>
<td>Allocates a minimum bandwidth for a traffic class, specifying the minimum as a percentage of the total interface bandwidth. This command is especially useful for protecting bandwidth allocation in multilink applications. See <em>Functional Notes</em> for more details.</td>
</tr>
<tr>
<td><code>unlimited</code></td>
<td>Optional. Specifies no limits on the priority queue bandwidth. Excessive traffic matching the QoS map can potentially use all of the available bandwidth on the interface, so use this feature with caution.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value is necessary for this command.

**Functional Notes**

The priority queue is intended for constant bit rate traffic such as voice due to the rate limiting. The sum of the bandwidths reserved by priority commands for all entries of a QoS map cannot exceed the `max-reserved-bandwidth` rate specified for the interfaces that the map is applied to. Priority bandwidth is guaranteed bandwidth (in kbps).

**Determining Bandwidth Entries**

*Note* When possible, use the `priority <bandwidth>` command to specify an absolute amount of bandwidth (in kbps) for the priority queue.
When determining the priority percent \(<\text{bandwidth}>\) entry, use the following formula:

\[
\frac{\text{Bandwidth}}{\text{Line Rate}}
\]

where

- **Bandwidth** minimum amount of bandwidth needed for the traffic (in kbps)
- **Line Rate** total data rate configured on the interface (for example, 8 DS0s (64 kbps per DS0) on a T1 equals a line rate of 512 kbps)

For example, to specify 80 kbps of data on an interface with a total of 512 kbps of available bandwidth, enter the following command:

```
ProCurve(config-qos-map 1)#priority percent 16
```

**Usage Examples**

The following example configures QoS for a network with the following needs:

- Reserve 15% of the line rate for routing traffic and L2 protocol traffic (\(\text{max-reserved-bandwidth} = 85\)) Line Rate = 512 kbps
- Guaranteed 256 kbps for Voice
- Guaranteed 96 kbps for Class 1
- Guaranteed 52 kbps for Class 2

To configure this QoS policy, enter the following QoS map and interface commands:

1. Allocate LLQ Priority voice traffic
   
   ```
   ProCurve(config)#qos map MyMap 1
   ProCurve(config-qos-map)#match list VOICE
   ProCurve(config-qos-map)#priority 256
   
   ProCurve(config)#qos map MyMap 2
   ProCurve(config-qos-map)#match list CLASS_1
   ProCurve(config-qos-map)#bandwidth 96
   
   ProCurve(config)#qos map MyMap 3
   ProCurve(config-qos-map)#match list CLASS_2
   ProCurve(config-qos-map)#bandwidth 52
   
   3. Specify the reserved bandwidth and apply the map
   ProCurve(config-fr 1)#max-reserved-bandwidth 85
   ProCurve(config-fr 1)#qos-policy out MyMap
   ```
The `set dscp <0-63>` command is an optional command for a QoS map that can be used to modify the Differentiated Service Code Point (DSCP) field (on matching packets) to the specified value. For more details on determining the DSCP field, refer to the *Technology Review* section of the `match` command on page 1269. Use the `no` form of this command to remove a specified DSCP value.

### Syntax Description

```plaintext
<0-63>   Specifies the decimal DSCP value.
```

### Default Values

No default value is necessary for this command.

### Functional Notes

QoS policies are configured in the SROS CLI to dictate the priority for servicing specified traffic types on a particular interface. QoS policies contain at least one match reference (using the `match` command) and one or more action items (using the `priority`, `bandwidth`, or `set` commands).

The `set dscp` command can be used to change the Differentiated Services (DS) Field for incoming traffic serviced by the QoS policy. Every IPv4 header contains an 8-bit Type of Service (ToS) field used for marking data types requiring special handling when traveling through the network. Originally this ToS field was used for IP precedence markings (using only the first 3 bits of the 8-bit field), and was later revised in RFC 2474 to create the 6-bit DS field (reserving the last two bits of the field for future use). The DS field can be manipulated to indicate higher or lower traffic priority using decimal values between 0 and 63.

### Usage Examples

This command sets the DSCP value (for all matching traffic) to 46:

```
ProCurve(config)#qos map VOICEMAP 10
ProCurve(config-qos-map)#set dscp 46
```
set precedence <0-7>

The set precedence command is an optional command for a QoS map that can be used to modify the IP Precedence value (on matching packets) to the specified value. For more details on determining the IP Precedence value, refer to the Technology Review section of the match command on page 1269. Use the no form of this command to remove a specified precedence.

Syntax Description

<0-7> Specifies the decimal IP precedence value.

Default Values

No default value is necessary for this command.

Functional Notes

QoS policies are configured in the SROS CLI to dictate the priority for servicing specified traffic types on a particular interface. QoS policies contain at least one match reference (using the match command) and one or more action items (using the priority, bandwidth, or set commands).

The set precedence command can be used to change the Differentiated Services (DS) Field for incoming traffic serviced by the QoS policy. Every IPv4 header contains an 8-bit Type of Service (ToS) field used for marking data types requiring special handling when traveling through the network. Originally this ToS field was used for IP Precedence markings (using only the first 3 bits of the 8-bit field), and was later revised in RFC 2474 to create the 6-bit DS field (reserving the last two bits of the field for future use). The IP Precedence value can be manipulated to indicate higher or lower traffic priority using decimal values between 0 and 7.

Usage Examples

This command sets the IP Precedence value (for all matching traffic) to 5:

ProCurve(config)#qos map VOICEMAP 10
ProCurve(config-qos-map)#set precedence 5
DHCP POOL COMMAND SET

To activate the DHCP Pool mode, enter the **ip dhcp-server pool** command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `bootfile <filename>` on page 1280
- `client-identifier <identifier>` on page 1281
- `client-name <name>` on page 1283
- `default-router <address> <secondary>` on page 1284
- `dns-server <address> <secondary>` on page 1285
- `domain-name <domain>` on page 1286
- `hardware-address <hardware-address> <type>` on page 1287
- `host <address> [subnet mask] or [prefix length]` on page 1289
- `lease <days> <hours> <minutes>` on page 1290
- `netbios-name-server <address> <secondary>` on page 1291
- `netbios-node-type <type>` on page 1292
- `network <address> [subnet mask] or [prefix length]` on page 1293
- `ntp-server <ip address>` on page 1294
- `option <option value> [ascii | hex | ip] <value>` on page 1295
- `tftp-server <server>` on page 1296
- `timezone-offset <offset>` on page 1297
bootfile <filename>

Use the bootfile command to specify a fully qualified directory-path name to a file located on a TFTP server on the network. Some network devices use the file (the path sent to the DHCP client in the DHCP-OFFER message) for initial configuration. Use the no form of this command to remove a configured boot file.

**Syntax Description**

| <filename> | Specifies a fully qualified directory-path name to the file located on the network. If the file is located in the root directory of the TFTP server, enter the filename only. |

**Default Values**

By default there is no specified boot file.

**Functional Notes**

RFC2131 provides specifications for DHCP servers to supply clients with information that allows the clients to exchange packets with other hosts on the network. DHCP clients that do not store the correct boot software on an internal flash drive can receive a bootfile from a TFTP server. The ProCurve Secure Router DHCP server can provide these devices with the address of the network TFTP server and the configuration filename. For example, some IP phones use this functionality to download the feature and key activation file. Use the tftp-server command (in the DHCP Pool command set) to specify the IP address of the network TFTP server.

RFC2131 includes provisions to allow DHCP servers to utilize the 128 octets designated for the bootfile directory-path for expanding the DHCP options field. RFC1533 outlines the available DHCP variables for the options field. This process must be negotiated between client and server during the DHCPDISCOVER process and should only take place if the client specifies a small maxDHCPmessage size in the DHCPDISCOVER message.

**Usage Examples**

The following example specifies the location of a TFTP server on the LAN at 10.10.0.4 and a bootfile of myconfig.cfg (located in the TFTP server root directory) for the DHCP pool IP_Phones:

```
ProCurve(config)#ip dhcp-server pool IP_Phones
ProCurve(config-dhcp)#tftp sever 10.10.0.4
ProCurve(config-dhcp)#bootfile myconfig.cfg
```
client-identifier <identifier>

Use the client-identifier command to specify a unique identifier (in dotted hexadecimal notation) for a Dynamic Host Configuration Protocol (DHCP) client. Use the no form of this command to remove a configured client identifier.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;identifier&gt;</td>
<td>Specify a custom client identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters).</td>
</tr>
<tr>
<td>OR</td>
<td>Specify the hexadecimal Media Access Control (MAC) address including a hexadecimal number added to the front of the MAC address to identify the media type.</td>
</tr>
<tr>
<td>For example, specifying the client-identifier for a MAC address of d217.0491.1150 defines the client identifier as 01:d2:17:04:91:11:50 (where 01 defines the media type as Ethernet).</td>
<td></td>
</tr>
<tr>
<td>For example, a custom client identifier of 0f:ff:ff:ff:ff:51:04:99:a1 may be entered using the &lt;identifier&gt; option.</td>
<td></td>
</tr>
</tbody>
</table>

Default Values

By default, the client identifier is populated using the following formula:

```
TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS
```

Where TYPE specifies the media type in the form of one hexadecimal byte (refer to hardware-address <hardware-address> <type> on page 1287 for a detailed listing of media types) and the MAC ADDRESS is the MAC address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to Ethernet 0/1 is used in this field.)

INTERFACE SPECIFIC INFO is only used for Frame Relay interfaces and can be determined using the following:

```
FR_PORT# : Q.922 ADDRESS
```

Where the FR_PORT# specifies the label assigned to the virtual Frame Relay interface using four hexadecimal bytes. For example, a virtual Frame Relay interface labeled 1 would have a FR_PORT# of 00:00:00:01.

The Q.922 ADDRESS field is populated using the following:

```
<table>
<thead>
<tr>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLCI (high order)</td>
<td>C/R</td>
<td>EA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLCI (lower)</td>
<td>FECN</td>
<td>BECN</td>
<td>DE</td>
<td>EA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

DLCI (high order) C/R EA
DLCI (lower) FECN BECN DE EA
Where the FECN, BECN, C/R, DE, and high order EA bits are assumed to be 0, and the lower order extended address (EA) bit is set to 1.

The following list provides a few example DLCIs and associated Q.922 addresses:

DLCI (decimal) / Q.922 address (hex)

16 / 0x0401
50 / 0x0C21
60 / 0x0CC1
70 / 0x1061
80 / 0x1401

Functional Notes

DHCP clients use client-identifiers in place of hardware addresses. To create the client identifier, begin with the two-digit numerical code representing the media type and append the client’s MAC address. For example, a Microsoft client with an Ethernet (01) MAC address d2:17:04:91:11:50 uses a client identifier of 01:d2:17:04:91:11:50.

Usage Examples

The following example specifies the client identifier for a Microsoft client with an Ethernet MAC address of d217.0491.1150:

ProCurve(config)#ip dhcp-server pool Microsoft_Clients
ProCurve(config-dhcp)#client-identifier 01:d2:17:04:91:11:50
**client-name <name>**

Use the `client-name` command to specify the name of a Dynamic Host Configuration Protocol (DHCP) client. Use the `no` form of this command to remove the configured client name.

**Syntax Description**

| `<name>` | Identifies the DHCP client (example is `client1`) with an alphanumeric string (up to 32 characters in length). |

**Note**  
*The specified client name should not contain the domain name.*

**Default Values**

By default, there are no specified client names.

**Usage Examples**

The following example specifies a client name of `myclient`:

```
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#client-name myclient
```
**default-router** `<address> <secondary>`

Use the `default-router` command to specify the default primary and secondary routers to use for the Dynamic Host Configuration Protocol (DHCP) client. Use the `no` form of this command to remove the configured router.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Specifies the address (in dotted decimal notation) of the preferred router on the client’s subnet (example: 192.168.4.254).</td>
</tr>
<tr>
<td><code>&lt;secondary&gt;</code></td>
<td>Optional. Specifies the address (in dotted decimal notation) of the second preferred router on the client’s subnet (example: 192.168.4.253).</td>
</tr>
</tbody>
</table>

**Default Values**

By default, there are no specified default routers.

**Functional Notes**

When specifying a router to use as the primary/secondary preferred router, verify that the listed router is on the same subnet as the DHCP client. The SROS allows a designation for two routers, listed in order of precedence.

**Usage Examples**

The following example configures a default router with address **192.168.4.254** and a secondary router with address **192.168.4.253**:

```
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#default-router 192.168.4.254 192.168.4.253
```
**dns-server <address> <secondary>**

Use the `dns-server` command to specify the default primary and secondary Domain Name System (DNS) servers to use for the Dynamic Host Configuration Protocol (DHCP) client. Use the `no` form of this command to remove the configured DNS server.

**Syntax Description**

| `<address>` | Specifies the address (in dotted decimal notation) of the preferred DNS server on the network (example: 192.168.4.254). |
| `<secondary>` | Optional. Specifies the address (in dotted decimal notation) of the second preferred DNS server on the network (example: 192.168.4.253). |

**Default Values**

By default, there are no specified default DNS servers.

**Usage Examples**

The following example specifies a default DNS server with address **192.168.3.254** and a secondary DNS server with address **192.168.4.253**:

```
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#dns-server 192.168.3.254 192.168.4.253
```
domain-name <domain>

Use the domain-name command to specify the domain name for the Dynamic Host Configuration Protocol (DHCP) client. Use the no form of this command to remove the configured domain name.

Syntax Description

| <name>          | Identifies the DHCP client (e.g., procurve.com) with an alphanumeric string (up to 32 characters in length). |

Default Values

By default, there are no specified domain names.

Usage Examples

The following example specifies a domain name of procurve.com:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#domain-name procurve.com
**hardware-address** <hardware-address> <type>

Use the **hardware-address** command to specify the name of a Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove the configured client name.

**Syntax Description**

<table>
<thead>
<tr>
<th><strong>&lt;hardware-address&gt;</strong></th>
<th>Specifies the hardware address (in hexadecimal notation with colon delimiters) of the preferred router on the client’s subnet (example d2:17:04:91:11:50).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>&lt;type&gt;</strong></td>
<td>Optional. Specifies the hardware protocol of the DHCP client. The hardware type field can be entered as follows:</td>
</tr>
<tr>
<td>ethernet</td>
<td>Specifies standard Ethernet networks.</td>
</tr>
<tr>
<td>ieee802</td>
<td>Specifies IEEE 802 standard networks.</td>
</tr>
<tr>
<td>&lt;1-21&gt;</td>
<td>Enter one of the hardware types listed in RFC1700.</td>
</tr>
</tbody>
</table>

The valid hardware types are as follows:

1. 10 Mb Ethernet
2. Experimental 3 Mb Ethernet
3. Amateur Radio AX.25
4. Proteon ProNET Token Ring
5. Chaos
6. IEEE 802 Networks
7. ARCNET
8. Hyperchannel
9. Lanstar
10. Autonet Short Address
11. LocalTalk
12. LocalNet (IBM PCNet or SYTEK LocalNet)
13. Ultra link
14. SMDS
15. Frame Relay
16. Asynchronous Transmission Mode (ATM)
17. HDLC
18. Fibre Channel
19. Asynchronous Transmission Mode (ATM)
20. Serial Line
21. Asynchronous Transmission Mode (ATM)

**Default Values**

By default, the hardware address type is set to 10 Mbps Ethernet.
Usage Examples

The following example specifies an Ethernet client with a MAC address of ae:11:54:60:99:10:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#hardware-address ae:11:54:60:99:10 ethernet
host <address> [ <subnet mask> or <prefix length> ]

Use the host command to specify the IP address and subnet mask for a manual binding to a Dynamic Host Configuration Protocol (DHCP) client. Use the no form of this command to remove the configured client address.

Syntax Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Specifies the IP address (in dotted decimal notation) for a manual binding to a DHCP client.</td>
</tr>
<tr>
<td>&lt;subnet mask&gt;</td>
<td>Optional. Specifies the network mask (subnet) for a manual binding to a DHCP client. If the subnet mask is left unspecified, the DHCP server examines its address pools to obtain an appropriate mask. If no valid mask is found in the address pools, the DHCP server uses the Class A, B, or C natural mask.</td>
</tr>
<tr>
<td>&lt;prefix length&gt;</td>
<td>Optional. Alternately, the prefix length may be used to specify the number of bits that comprise the network address. The prefix length must be preceded by a forward slash ( / ). For example, to specify an IP address with a subnet mask of 255.255.0.0, enter /16 after the address.</td>
</tr>
</tbody>
</table>

Default Values

By default, there are no specified host addresses.

Usage Examples

The following examples show two different ways to specify a client with IP address 10.200.5.99 and a 21-bit subnet mask:

```
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#host 10.200.5.99 255.255.248.0

or

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#host 10.200.5.99/21
```
lease <days> <hours> <minutes>

Use the lease command to specify the duration of the lease for an IP address assigned to a Dynamic Host Configuration Protocol (DHCP) client. Use the no form of this command to return to the default lease value.

Syntax Description

- `<days>`: Specifies the duration of the IP address lease in days.
- `<hours>`: Optional. Specifies the number of hours in a lease. You may only enter a value in the hours field if the days field is specified.
- `<minutes>`: Optional. Specifies the number of minutes in a lease. You may only enter a value in the minutes field if the days and hours fields are specified.

Default Values

By default, an IP address lease is one day.

Usage Examples

The following example specifies a lease of 2 days:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#lease 2

The following example specifies a lease of 1 hour:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#lease 0 1

The following example specifies a lease of 30 minutes:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#lease 0 0 30
netbios-name-server <address> <secondary>

Use the netbios-name-server command to specify the primary and secondary NetBIOS Windows Internet Naming Service (WINS) name servers available for use by the Dynamic Host Configuration Protocol (DHCP) clients. Use the no form of this command to remove a configured NetBIOS name server.

Syntax Description

<address> Specifies the address (in dotted decimal notation) of the preferred NetBIOS WINS name server on the network (example: 192.168.6.99).

<secondary> Optional. Specifies the address (in dotted decimal notation) of the second preferred NetBIOS WINS name server on the network (example: 192.168.8.15).

Default Values

By default, there are no configured NetBIOS WINS name servers.

Usage Examples

The following example specifies a primary NetBIOS WINS name server with an IP address of 192.168.6.99 and a secondary with an IP address of 192.168.8.15:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#netbios-name-server 192.168.6.99 192.168.8.15
netbios-node-type <type>

Use the netbios-node-type command to specify the type of NetBIOS node used with Dynamic Host Configuration Protocol (DHCP) clients. Use the no form of this command to remove a configured NetBIOS node type.

Syntax Description

<table>
<thead>
<tr>
<th>&lt;type&gt;</th>
<th>Specifies the NetBIOS node type used with DHCP clients.</th>
</tr>
</thead>
</table>

Valid node types are as follows:

- **b-node** (1) - Broadcast node
- **p-node** (2) - Peer-to-Peer node
- **m-node** (4) - Mixed node
- **h-node** (8) - Hybrid node (Recommended)

Alternately, the node type can be specified using the numerical value listed next to the nodes above.

Default Values

| <type> | h-node (8) - Hybrid node |

Usage Examples

The following example specifies a client's NetBIOS node type as **h-node**:

```
ProCurve(config)#ip dhcp-server pool MyPool
(ProCurve(config-dhcp)#netbios-node-type h-node
```

Alternately, the following also specifies the client's NetBIOS node type as **h-node**:

```
ProCurve(config-dhcp)#netbios-node-type 8
```
network <address> [<subnet mask> or <prefix length>]

Use the network command to specify the subnet number and mask for an SROS Dynamic Host Configuration Protocol (DHCP) server address pool. Use the no form of this command to remove a configured subnet.

Syntax Description

- **<ip address>**
  - Specifies the IP address (in dotted decimal notation) of the DHCP address pool.

- **<subnet mask>**
  - Optional. Specifies the network mask (subnet) for the address pool. If the subnet mask is left unspecified, the DHCP server uses the Class A, B, or C natural mask.

- **<prefix length>**
  - Optional. Alternately, the prefix length may be used to specify the number of bits that comprise the network address. The prefix length must be preceded by a forward slash (/). For example, to specify an IP address with a subnet mask of 255.255.0.0, enter /16 after the address.

Default Values

By default, there are no configured DHCP address pools.

Usage Examples

The following examples show two different ways to configure an address pool subnet of 192.168.0.0 with a 16-bit subnet mask:

```
ProCurve(config)#ip dhcp-server pool MyPool
(ProCurve(config-dhcp)#network 192.168.0.0 255.255.0.0

or

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#network 192.168.0.0 /16
```
**ntp-server** `<ip address>`

Use the `ntp-server` command to specify the name of the Network Time Protocol (NTP) server published to the client.

**Syntax Description**

| `<ip address>` | Specifies the IP address of the NTP server. |

**Default Values**

By default, no NTP server is defined.

**Usage Examples**

The following example specifies the IP address of the NTP server:

```
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#ntp-server 192.168.1.1
```
option <option value> [ascii | hex | ip] <value>

Use the option command to describe a generic DHCP option to be published to the client. The user may specify any number of generic options to be published to the client.

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;option value&gt;</td>
<td>Specifies the value of the generic DHCP option published to the client. Range: 0 to 255.</td>
</tr>
<tr>
<td>ascii</td>
<td>Specifies the DHCP option information in ascii format.</td>
</tr>
<tr>
<td>hex</td>
<td>Specifies the DHCP option information in hexadecimal format.</td>
</tr>
<tr>
<td>ip</td>
<td>Specifies the DHCP option information in IP format.</td>
</tr>
<tr>
<td>&lt;value&gt;</td>
<td>Specifies the ASCII, hexadecimal, or IP value. The value for ascii is simple text. The value for hex is an 8-digit hexadecimal number (32 bit). The value for ip is a standard IP address in the format A.B.C.D.</td>
</tr>
</tbody>
</table>

Default Values

No default value necessary for this command.

Usage Examples

The following example publishes DHCP options to the client:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#option 100 ascii ascii_value
ProCurve(config-dhcp)#option 101 hex AB458E80
ProCurve(config-dhcp)#option 102 ip 192.168.1.1
**tftp-server <server>**

Use the `tftp-server` command to specify the IP address or DNS name of the TFTP server published to the client.

**Syntax Description**

| <server> | Specifies the DNS name or dotted notation IP address of the server. |

**Default Values**

By default, no tftp server is defined.

**Usage Examples**

The following example specifies the IP address of the TFTP server:

```
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#tftp-server 192.168.1.1
```

The following example specifies the DNS name of the TFTP server:

```
ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#tftp-server MyServer.procurve.com
```
timezone-offset <offset>

Use the timezone-offset command to specify the timezone adjustment (in hours) published to the client.

Syntax Description

| <offset> | Specifies the timezone adjustment (in hours) published to the client. Use an integer from -12 to 12. |

Default Values

No default value necessary for this command.

Usage Examples

The following example sets the timezone adjustment for the client to -3 hours. For example, if the server time is configured for eastern time and the client is configured for Pacific time, you can set the client timezone adjustment to -3 hours:

ProCurve(config)#ip dhcp-server pool MyPool
ProCurve(config-dhcp)#timezone-offset -3
To activate the Radius Group mode, enter the **aaagroup server** command at the Global Configuration mode prompt. For example:

```
ProCurve> enable
ProCurve# configure terminal
ProCurve(config)# aaagroup server radius myServer
ProCurve(config-sg-radius)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `server [acct-port <port number>| auth-port <port number>]` on page 1299
server \[acct-port \textless port number\]| auth-port \textless port number\]}

Use the \texttt{server} command to add a predefined RADIUS server to the current named list of servers. See \texttt{radius-server} on page 452 for more information.

\textbf{Syntax Description}

- \texttt{acct-port \textless port number\} \hfill Defines the accounting port value.
- \texttt{auth-port \textless port number\} \hfill Defines the authorization port value.

\textbf{Default Values}

No defaults necessary for this command.

\textbf{Usage Examples}

The following example adds a server to the \texttt{myServers} list:

\begin{verbatim}
ProCurve(config)#aaa group server radius myServers
ProCurve(config-sg-radius)#server 1.2.3.4 acct-port 786 auth-port 1812
ProCurve(config-sg-radius)#server 4.3.2.1
ProCurve(config-sg-radius)#exit
ProCurve(config)#
\end{verbatim}

or

\begin{verbatim}
ProCurve(config)#aaa group server radius myServers
ProCurve(config-sg-radius)#server 4.3.2.1
ProCurve(config-sg-radius)#exit
ProCurve(config)#
\end{verbatim}
TACACS+ GROUP CONFIGURATION COMMAND SET

To activate the Terminal Access Controller Access Control System Plus (TACACS+) Group Configuration mode, enter the `aaa group server tacacs+` command at the Global Configuration mode prompt. For example:

```
ProCurve>enable
ProCurve#configure terminal
ProCurve(config)#aaa group server tacacs+ TEST GROUP
ProCurve(config-sg-tacacs+)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311

All other commands for this command set are described in this section in alphabetical order.

- `server <host>` on page 1301
server <host>

Use the **server** command to specify a particular TACACS+ server’s IP address or host name.

**Syntax Description**

| <host> | Specifies a TACACS+ server IP address. |

**Default Values**

No default is necessary for this command.

**Usage Examples**

The following example specifies the IP address of the TACACS+ server:

```
ProCurve(config)#aaa group server tacacs+ TEST_GROUP
ProCurve(config-sg-tacacs+)#server 192.168.1.1
ProCurve(config-sg-tacacs+)#
```
The following section contains descriptions of commands which are common across multiple command sets. These commands are listed in alphabetical order.

- `alias <"text">` on page 1303
- `bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>` on page 1304
- `description <text>` on page 1307
- `do` on page 1308
- `end` on page 1309
- `exit` on page 1310
- `ping <address>` on page 1311
- `show running-config` on page 1313
- `shutdown` on page 1315
alias <"text">

Use the alias command to populate the ifAlias OID (Interface Table MIB of RFC2863) for all physical and virtual interfaces when using Simple Network Management Protocol (SNMP) management stations.

Syntax Description

| input  | Alphanumeric character string (no more than 64 characters) describing the interface (for SNMP) — must be encased in quotation marks. |

Default Values

No defaults required for this command.

Functional Notes

The ifAlias OID is a member of the ifXEntry object-type (defined in RFC2863) used to provide a non-volatile, unique name for various interfaces. This name is preserved through power cycles. Enter a string (using the alias command) which clearly identifies the interface.

Usage Examples

The following example defines a unique character string for the T1 interface:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#alias “CIRCUIT_ID_23-908-8887-401”
```

Technology Review

Please refer to RFC2863 for more detailed information on the ifAlias display string.
bind <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>

Use the `bind` command to create a bind map from a created tdm-group on an interface to a virtual interface.

**Caution** Changing `bind` settings could potentially result in service interruption.

### Syntax Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;#&gt;</code></td>
<td>Number descriptor or label for identifying the bind (useful in systems that allow multiple binds)</td>
</tr>
<tr>
<td><code>&lt;from interface&gt;</code></td>
<td>Specifies the interface (physical of virtual) on one end of the bind. Enter <code>bind 1 ?</code> for a list of valid interfaces.</td>
</tr>
<tr>
<td><code>&lt;slot/port&gt;</code></td>
<td>Used when a physical interface is specified in the <code>&lt;from interface&gt;</code> subcommand (For example: specifying the T1 port of a T1 module would be <code>t1 1/1</code>).</td>
</tr>
<tr>
<td><code>&lt;tdm-group#&gt;</code></td>
<td>Specifies which configured tdm-group to use for this bind. This subcommand only applies to T1 physical interfaces.</td>
</tr>
<tr>
<td><code>&lt;to interface&gt;</code></td>
<td>Specifies the virtual interface on the other end of the bind. Use the <code>?</code> to display a list of valid interfaces.</td>
</tr>
<tr>
<td><code>&lt;slot/port&gt;</code></td>
<td>Used when a physical interface is specified in the <code>&lt;to interface&gt;</code> subcommand. (For example, specifying the primary T1 port of a T1 module would be <code>t1 1/1</code>).</td>
</tr>
<tr>
<td><code>&lt;rbs&gt;</code></td>
<td>This optional field is used in order to maintain robbed bit signaling through the bind when voice is being delivered.</td>
</tr>
</tbody>
</table>

### Default Values

By default, there are no configured binds.

### Functional Notes

Binds provide the mechanism for binding a configured virtual (layer 2) endpoint with a physical (layer 1) interface. Supported layer 2 protocols include Frame Relay and point-to-point protocol (PPP).
Usage Examples

The following example creates a Frame Relay endpoint and binds it to the t1 1/1 physical interface:

1. Create the Frame Relay virtual endpoint and set the signaling method:

ProCurve(config)#interface frame-relay 1
ProCurve(config-fr 1)#frame-relay lmi-type cisco

2. Create the sub-interface and configure the PVC parameters (including DLCI and IP address):

ProCurve(config-fr 1)#interface fr 1.1
ProCurve(config-fr 1.1)#frame-relay interface-dlci 17
ProCurve(config-fr 1.1)#ip address 168.125.33.252 255.255.255.252

3. Create the tdm-group of 12 DS0s (64K) on the t1 physical interface:

ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#tdm-group 1 timeslots 1-12 speed 64
ProCurve(config-t1 1/1)#exit

4. Connect the Frame Relay sub-interface with port t1 1/1:

ProCurve(config)#bind 1 t1 1/1 1 fr 1

Technology Review

Creating an endpoint that uses a layer 2 protocol (such as Frame Relay) is generally a four-step process:

Step 1:
Create the Frame Relay virtual endpoint (using the `interface frame-relay` command) and set the signaling method (using the `frame-relay lmi-type` command). Also included in the Frame Relay virtual endpoint are all the applicable Frame Relay timers logging thresholds, encapsulation types, etc. Generally, most Frame Relay virtual interface parameters should be left at their default state. For example, the following creates a Frame Relay interface labeled 7 and sets the signaling method to `ansi`.

ProCurve(config)#interface frame-relay 7
ProCurve(config-fr 7)#frame-relay lmi-type ansi

Step 2:
Create the sub-interface and configure the PVC parameters. Using the sub-interface, apply access policies to the interface, create bridging interfaces, configure backup, assign an IP address, and set the PVC data-link control identifier (DLCI). For example, the following creates a Frame Relay sub-interface labeled 22, sets the DLCI to 30, and assigns an IP address of 193.44.69.253 to the interface.
ProCurve(config-fr 7)#interface fr 7.22
ProCurve(config-fr 7.22)#frame-relay interface-dlci 30
ProCurve(config-fr 7.22)#ip address 193.44.69.253 255.255.255.252

Step 3:
Specify the group of DS0s used for signaling on the T1 interface by creating a `tdm-group`. Group any number of contiguous DS0s together to create a data pipe for layer 2 signaling. Also use the `tdm-group` command to specify the per-DS0 signaling rate on the interface. For example, the following creates a tdm-group labeled `9` containing 20 DS0s (each DS0 having a data rate of 56 kbps).

ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#tdm-group 9 timeslots 1-20 speed 56
ProCurve(config-t1 1/1)#exit

Step 4:
Make the association between the layer 2 endpoint and the physical interface using the `bind` command. Supported layer 2 protocols include Frame Relay and point-to-point protocol (PPP). For example, the following creates a bind (labeled `5`) to make an association between the Frame Relay virtual interface (`fr 7`) and the tdm-group configured on interface t1 1/1 (`tdm-group 9`).

ProCurve(config)#bind 5 t1 1/1 9 fr 7
**description** \(<text>\)

Use the `description` command to identify the specified interface, both physical and virtual (for example, circuit ID, contact information, etc.).

**Syntax Description**

|  | Identifies the specified interface using up to 80 alphanumeric characters. |

**Default Values**

No defaults required for this command.

**Usage Examples**

The following example enters comment information using the `description` command:

```
ProCurve(config)#interface t1 1/1
ProCurve(config-t1 1/1)#description This is the Dallas office T1
```
do

Use the `do` command to execute any command, regardless of the active configuration mode.

**Syntax Description**

No subcommands.

**Default Values**

No defaults required for this command.

**Functional Notes**

Use the `do` command to view configurations or interface states after configuration changes are made without exiting to the Enable mode.

**Usage Examples**

The `do` command provides a way to execute commands in other configuration modes without taking the time to exit the current configuration mode and enter the desired one. The following example shows the `do` command used to remove all dynamic entries from the ARP cache:

```
ProCurve(config)#do clear arp-cache
```
end

Use the `end` command to exit the current configuration mode and enter the Enable Security mode.

**Note** When exiting the Global Configuration mode, remember to perform a `copy running-config startup-config` to save all configuration changes.

**Syntax Description**

No subcommands.

**Default Values**

No defaults necessary for this command.

**Usage Examples**

The following example shows the `end` command being executed in the T1 Configuration mode:

```
ProCurve(config-t1 1/1)#end
#

#- Enable Security mode command prompt
```
exit

Use the `exit` command to exit the current Configuration mode and enter the previous one. For example, using the `exit` command in an interface configuration mode will activate the Global Configuration mode. When using the `exit` command in the Basic mode, the current session will be terminated.

**Note** When exiting the Global Configuration mode, remember to perform a `copy running-config startup-config` to save all configuration changes.

**Syntax Description**
No subcommands.

**Default Values**
No defaults necessary for this command.

**Usage Examples**
The following example shows the `exit` command being executed in the Global Configuration mode:

```
ProCurve(config)#exit
#
# - Enable Security mode command prompt
```
ping <address>

Use the ping command (at the Enable Command mode prompt) to verify IP network connectivity.

Syntax Description

| <address> | Optional. Specifies the IP address of the system to PING. Entering the ping command with no specified address prompts the user with parameters for a more detailed PING configuration. See Functional Notes (below) for more information. |

Default Values

No default value necessary for this command.

Functional Notes

The ping command helps diagnose basic IP network connectivity using the Packet InterNet Groper (PING) program to repeatedly bounce Internet Control Message Protocol (ICMP) Echo_Request packets off a system (using a specified IP address). The SROS allows executing a standard PING request to a specified IP address or provides a set of prompts to configure a more specific PING configuration.

The following is a list of output messages from the ping command:

! Success
- Destination Host Unreachable
$ Invalid Host Address
X TTL Expired in Transit
? Unknown Host
* Request Timed Out

The following is a list of available extended PING fields with descriptions:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target IP Address</td>
<td>Specifies the IP address of the system to PING.</td>
</tr>
<tr>
<td>Repeat Count</td>
<td>Number of PING packets to send to the system (valid range: 1 to 1,000,000).</td>
</tr>
<tr>
<td>Datagram Size</td>
<td>Size (in bytes) of the PING packet (valid range: 1 to 1448).</td>
</tr>
<tr>
<td>Timeout in Seconds</td>
<td>If a PING response is not received within the timeout period, the PING is considered unsuccessful (valid range: 1 to 5 seconds).</td>
</tr>
<tr>
<td>Extended Commands</td>
<td>Specifies whether additional commands are desired for more PING configuration parameters.</td>
</tr>
<tr>
<td>Source Address</td>
<td>Specifies the IP address to use as the source address in the ECHO_REQ (or interface) packets.</td>
</tr>
<tr>
<td>Data Pattern</td>
<td>Specifies an alphanumerical string to use (the ASCII equivalent) as the data pattern in the ECHO_REQ packets.</td>
</tr>
</tbody>
</table>
Sweep Range of Sizes: Varies the sizes of the ECHO_REQ packets transmitted.
Sweep Min Size: Specifies the minimum size of the ECHO_REQ packet (valid range: 0 to 1488).
Sweep Max Size: Specifies the maximum size of the ECHO_REQ packet (valid range: Sweep Min Size to 1448).
Sweep Interval: Specifies the interval used to determine packet size when performing the sweep (valid range: 1 to 1448).
Verbose Output: Specifies an extended results output.

Usage Examples
The following is an example of a successful ping command:

#ping
Target IP address: 192.168.0.30
Repeat count[1-1000000]: 5
Datagram Size [1-1000000]: 100
Timeout in seconds [1-5]: 2
Extended Commands? [y or n]: n
Type CTRL+C to abort.
Legend: '!' = Success '?' = Unknown host '$' = Invalid host address
       '*' = Request timed out '-' = Destination host unreachable
       'x' = TTL expired in transit

Pinging 192.168.0.30 with 100 bytes of data:
!!!!
Success rate is 100 percent (5/5) round-trip min/avg/max = 19/20.8/25 ms
show running-config

Use the `show running-config` command to display a text print of all the non-default parameters contained in the current running configuration file. Use the `verbose` keyword to display a text print of the entire configuration (including parameters in their default state). Specific portions of the running-config may be displayed, based on the command entered.

Variations of this command include the following:

- `show running-config`
- `show running-config access-lists`
- `show running-config access-lists verbose`
- `show running-config checksum`
- `show running-config interface <interface>`
- `show running-config interface <interface> verbose`
- `show running-config interface vlan <vlan id>`
- `show running-config interface vlan <vlan id> verbose`
- `show running-config policy-class`
- `show running-config policy-class verbose`
- `show running-config qos-map`
- `show running-config qos-map verbose`
- `show running-config verbose`

**Syntax Description**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-lists</td>
<td>Displays the current running configuration for all configured IP access lists.</td>
</tr>
<tr>
<td>interface <code>&lt;interface&gt;</code></td>
<td>Displays the current running configuration for a particular interface. Type the <code>show running-config interface ?</code> command to display a list of valid interfaces.</td>
</tr>
<tr>
<td>policy-class</td>
<td>Displays the current running configuration for all configured policy classes.</td>
</tr>
<tr>
<td>qos-map</td>
<td>Displays the current running configuration for all configured QoS maps.</td>
</tr>
<tr>
<td>verbose</td>
<td>Displays the entire running configuration to the terminal screen (versus only the non-default values).</td>
</tr>
<tr>
<td>checksum</td>
<td>Displays the encrypted Message Digest 5 (md5) version of the running configuration.</td>
</tr>
</tbody>
</table>

**Default Values**

No default value necessary for this command.
Usage Examples

The following is a sample output from the `show running-config` command:

```
>enable
#show running-config
Building configuration...
!
no enable password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
!
ip policy-timeout tcp all-ports 600
ip policy-timeout udp all-ports 60
ip policy-timeout icmp 60
!
interface eth 0/1.......
```
**shutdown**

Use the `shutdown` command to disable the interface (both physical and virtual) so that no data will be passed through. Use the `no` form of this command to turn on the interface and allow it to pass data. By default, all interfaces are disabled.

**Syntax Description**

No subcommands.

**Default Values**

By default, all interfaces are disabled.

**Usage Examples**

The following example administratively disables the modem interface:

```
ProCurve(config)#interface modem 1/2
ProCurve(config-modem 1/2)#shutdown
```
Index

A
aaa accounting commands 288
aaa accounting suppress null-username 289
aaa accounting update 290
aaa authentication 291
aaa authentication enable default 293
aaa authentication login 294
aaa authorization 296
aaa authorization commands 295
aaa group server 297, 1298
aaa on 298
aaa processes 300
access-class in 504, 513
access-policy 568, 667, 736, 816, 883, 943, 978, 1055
accounting commands 491, 505, 514
ADSL Interface Configuration command set 519
advertisement-interval 1219
alias 1303
alias link 884, 979
anning-tree mode 483
antireplay 1136, 1147
area default-cost 1236
area range 1237
area stub 1238
arp arpa 301, 569
as-path-list 1220
ATM Interface Configuration command set 662
atm routed-bridged ip 668
ATM Sub-Interface Config command set 665
attribute 1161, 1168
authentication 1162
authorization commands 492, 506, 515
auto cost reference-bandwidth 1239
autosyn 22, 302

B
backup auto-backup 669, 817, 980
backup auto-restore 670, 818, 886, 981, 1057
backup backup-delay 671, 819, 887, 982, 1058
backup call-mode 672, 820, 888, 983, 1059
backup connect-timeout 676, 824, 892, 987, 1062
backup force 677, 825, 893, 988, 1063
backup maximum retry 678, 826, 894, 989, 1064
backup number 679, 827, 895, 990, 1065
backup priority 680, 828, 896, 991, 1066
backup randomize-timers 681, 829, 897, 992, 1067
backup redial-delay 682, 830, 898, 993, 1068
backup restore-delay 683, 831, 899, 994, 1069
backup schedule 684, 832, 900, 995, 1070
backup shutdown 685, 833, 901, 996, 1071
bandwidth 570, 686, 737, 794, 834, 902, 944, 997, 1072, 1266
banner 304
basic 5
Basic Mode command set 12
deterministic-med 1209
deterministic-med 1209
deterministic-med 1209
developer ipsec sa 31
developer sa 30
dump-core 32
event-history 33
host 34
ip bgp 35

C
CA Profile command set 1120
caller-id override 531, 643
call-type 1113
certificate 1132
certificate ca 1133
Certificate Configuration command set 1131
clear access-list 24
clear arp-cache 25
clear arp-entry 26
clear bridge 27
clear counters 29
clear crypto ike sa 30
clear crypto ipsec sa 31
clear dump-core 32
clear event-history 33
clear host 34
clear ip bgp 35
clear ip cache 36
clear ip dhcp-server binding 37
clear ip igmp group 38
clear ip ospf 39
clear ip policy-sessions 40
clear ip policy-stats 42
clear ip prefix-list 43
clear ip route 44
clear lldp counters 45
clear lldp counters interface 46
clear pppoe 48
clear process cpu max 49
clear processes queue 50
clear qos map 51
clear route-map counters 52
clear sip location 53
clear spanning-tree counters 55
clear spanning-tree detected-protocols 56
clear tacacs+ statistics 57
clear user 58
clear user-registration 54

 CLI
  accessing with PC 5
  error messages 9
  introduction 5
  shortcuts 7
client authentication host 1169
client authentication host xauth-type 1170
client authentication server list 1171
client configuration pool 1172
client-identifier 1281
client-name 1283
clock auto-correct-dst 59, 308
clock set 60, 309
clock source 551, 648
clock timezone 61, 310
coding 541, 552, 627, 649
command descriptions 10
command level path 7
Command Line Interface
  accessing with PC 5
  error messages 9
  shortcuts 7
command security levels
  basic 5
  enable 5
commit-bit 1137
common CLI functions 8
common commands 1302
configuration 285
configuration modes
  global 6
  interface 6
  line 6
  router 6
configure 63
connect pri 1114
connected 1244, 1259
console port
  configuring 5
  receiving files 67
copy boot 64
copy cflash 65
copy cflash tftp 66
copy cflash xmodem 67
copy console 68
copy flash 69
  copy flash tftp 70
  copy flash xmodem 71
  copy interface 72
  copy running-config 73, 76
  copy running-config tftp 74
  copy running-config xmodem 75
  copy startup-config tftp 77
  copy startup-config xmodem 78
  copy tftp 79
  copy xmodem 80
crl 1134
crl optional 1121
crypto ca authenticate 312
crypto ca certificate chain 314
crypto ca enroll 315
crypto ca import certificate 316
crypto ca import crl 317
crypto ca profile 318
crypto ike 319
crypto ike policy 1167
crypto ike remote-id 323
crypto ipsec transform-set 325
crypto map 326, 573, 688, 745, 836, 904, 945, 1000, 1073
Crypto Map IKE command set 1135
crypto map ipsec-ike 1135
crypto map ipsec-manual 1146
Crypto Map Manual command set 1146
D
databits 493
data-call 328, 329
debug 9
debug aaa 81
debug access-list 82
debug arp 83
debug atm events 84
debug atm oam 85
debug atm packet 86
debug backup 87
debug chat-interfaces 89
debug crypto 90
debug data-call 91
debug demand-routing 92
debug dialup-interfaces 93
debug dynamic-dns 94
debug firewall 95
d debug firewall alg sip 96
d debug frame-relay 97
d debug frame-relay multilink 98
d debug interface 99
d debug interface adsl events 100
d debug ip bgp 101
d debug ip dhcp-client 102
d debug ip dhcp-server 103
d debug ip dns-client 104
d debug ip dns-proxy 105
d debug ip http 106
d debug ip icmp 107
d debug ip igmp 108
d debug ip mroute 109
d debug ip ospf 110
d debug ip packet 111
d debug ip pim-sparse 113
d debug ip pim-sparse assert 114
d debug ip pim-sparse hello 115
d debug ip pim-sparse join/prune 116
d debug ip pim-sparse packets 117
d debug ip pim-sparse register 118
d debug ip policy 119
d debug ip rip 120
d debug ip routing 121
d debug ip tcp events 122
d debug ip tcp md5 123
d debug ip udp 124
d debug isdn 125
d debug isdn events 126
d debug isdn resource-manager 127
d debug isdn verbose 128
d debug lldp 129
d debug port-auth 130
d debug ppp 131
d debug pppoe client 132
d debug radius 133
d debug sip 134, 135
d debug sip stack 136
d debug snmp 137
d debug spanning-tree 138
d debug spanning-tree bpdu 139
d debug system 140
d debug tacacs+ packets 141
d debug tftp 142
d debug-bridge 88
default-information originate 1240
default-metric 1241, 1255
default-router 1284
deny 1179
description 1307
DHCP Pool command set 1279
dialin 644
dir 143, 144
disable 13
disable, basic mode 13
disable, enable mode 145
distance bgp 1213
distribute-list 1256
dns-server 1158, 1285
do 1308
domain-name 1286
DSX-1 Interface Configuration command set 540
dynamic-dns 575, 690, 748, 838, 906, 947, 1002, 1075
E
E1 Interface Configuration command set 550
ebgp-multihop 1221
e-mail address 1122
enable 13, 146
enable password 330
enable, basic mode 13
enable, enable mode 20
enable, understanding 5
encapsulation 692
encapsulation 802.1q 577
encapsulation frame-relay ietf 795
encryption 1163
end 1309
enrollment retry 1123
enrollment terminal 1124
enrollment url 1125
erase 147
erase cflash 148
erase file-system cflash 149
erase flash 148
et-clock-source 634
Ethernet Interface Configuration command set 565
Ethernet Sub-Interface Configuration command set 565
event-history on 331
event-history priority 332
events 150
exception memory minimum 334
exception report 335
exception report generate 151
exit 1310
fair-queue 693, 750, 796, 908, 1004
fdl 650
flowcontrol 494
fqdn 1126
frame-relay bc 840
frame-relay be 841
frame-relay fragment 842
frame-relay interface-dlci 843
frame-relay intf-type 797
frame-relay lmi-n391dce 798
frame-relay lmi-n391dte 799
frame-relay lmi-n392dce 800
frame-relay lmi-n392dte 801
frame-relay lmi-n393dce 802
frame-relay lmi-n393dte 803
frame-relay lmi-t391dte 804, 805
frame-relay lmi-t392dce 805
frame-relay lmi-type 806
frame-relay multilink 807
framing 542, 553, 628, 651
ftp authentication 336
full-duplex 578
G.703 Interface Configuration command set 626
Global Configuration Mode command set 285
group 1164
half-duplex 579
hardware-address 1287
hash 1165
HDLC Configuration command set 881
hold-timer 694, 752, 809, 909, 1005
hold-timer 1214, 1222
host 1289
hostname 337
ignore dcd 635
IKE Client command set 1157
IKE Policy Attributes command set 1161
IKE Policy command set 1167
ike-policy 1139, 1148
incoming-accept-number 1115
initiate 1173
init-string 645
interface 338
interface atm 339, 662, 665
interface bri 524
interface e1 550
interface ethernet 565
interface ethernet sub 566
interface frame-relay 793, 814
interface G.703 626
interface hdlc 881
interface hdlc 343
interface loopback 345, 942
interface modem 642
interface ppp 346, 734, 976
interface range 348, 565
interface serial 633
interface t1 540, 647
interface tunnel 349, 1053
invert etclock 636
invert rxclock 637
invert txclock 638
ip access-group 580, 695, 754, 844, 910, 949, 1006, 1077
ip access-list extended 350
ip access-list standard 354
ip address 1078
ip address dhcp 581, 696, 845, 1007
ip address negotiated 755, 1009
ip address secondary 583, 698, 756, 847, 911, 950, 1010
ip classless 357
ip community-list 358, 1232
ip crypto 359
ip default-gateway 360
ip dhcp 699, 848
ip dhcp release 584
ip dhcp renew 585
ip dhcp-server database local 361
ip dhcp-server excluded-address 362
ip dhcp-server ping packets 363
ip dhcp-server ping timeout 364
ip dhcp-server pool 365, 1279
ip domain-lookup 366
ip domain-name 367
ip domain-proxy 368
ip firewall 369
ip firewall alg 374
ip firewall alg sip udp 376
ip firewall attack-log threshold 377
ip firewall check reflexive-traffic 378
ip firewall check rst-seq 379
ip firewall check syn-flood 380
ip firewall check winnuke 381
ip firewall fast-nat-fallback 382
ip firewall fin-timeout 383
ip firewall policy-log threshold 384
ip firewall rst-timeout 385
ip firewall stealth 386
ip forward-protocol udp 387
ip ftp agent 390
ip ftp source-interface 391
ip helper-address 586, 700, 757, 849, 912, 951, 1011, 1079
ip host 392
ip http 393
ip igmp 587, 701, 758, 850, 913, 952, 1012, 1080
ip igmp join 394
ip load-sharing per-destination 395
ip mcast-stub downstream 588, 702, 759, 851, 914, 953, 1013, 1081
ip mcast-stub fixed 589, 703, 760, 852, 915, 954, 1014, 1082
ip mcast-stub helper-address 397, 590
ip mcast-stub helper-enable 704, 761, 853, 916, 955, 1015, 1083
ip mcast-stub upstream 591, 705, 762, 854, 917, 956, 1016, 1084
ip multicast routing 398
ip name-server 399
ip ospf 592, 706, 763, 855, 918, 957, 1017, 1085
ip ospf authentication 593, 707, 764, 856, 919, 958, 1018, 1086
ip ospf network 594, 708, 765, 857, 920, 959, 1019, 1087
ip pim sparse-mode 595, 709, 858, 921, 960, 1020, 1088
ip pim-sparse dr-priority 596, 710, 859, 922, 961, 1021, 1089
ip pim-sparse hello-timer 597, 711, 860, 923, 962, 1022, 1090
ip pim-sparse nbr-timeout 598, 712, 861, 924, 963, 1023, 1091
ip pim-sparse override-interval 599, 713, 862, 925, 964, 1024, 1092
ip pim-sparse propagation-delay 600, 714, 863, 926, 965, 1025, 1093
ip policy route-map 601, 715, 766, 864, 927, 966, 1026, 1094
ip policy-class 400
ip policy-class max-sessions
  ip policy-class max-host-sessions 405
ip policy-class rpf-check 406
ip policy-timeout 407
ip prefix-list description 409
ip prefix-list seq 410
ip proxy-arp 602, 716, 767, 865, 928, 967, 1027, 1095
ip radius source-interface 411
ip rip receive version 603, 717, 768, 866, 929, 968, 1028, 1096, 1264
ip rip send version 604, 718, 769, 867, 930, 969, 1029, 1097, 1264
ip rip summary-address 605, 719, 770, 868, 931, 970, 1030, 1098
ip route 412
ip route-cache 606, 720, 771, 869, 932, 971, 1031, 1099
ip routing 413
ip rtp firewall-traversal 414
ip sip 417
ip sip proxy 418
ip sip register 419
ip smtp agent 420
ip snmp source-interface 421
ip ssh-server 422
ip subnet-zero 423
ip tacacs source-interface 424
ip telnet-server 422
ip tftp server 425
ip tftp source-interface 426
ip unnumbered 607, 721, 772, 870, 933, 972, 1032
ip-address 1127
ip-range 1159
isdn ldn 532
isdn spid 533
isdn switch-type 534
isdn-group 427, 1112
isdn-number-template 428
J
join-prune-msg-interval 1250
K
keepalive 773, 934, 1033, 1100
L
line 430
line (console) Interface Configuration command set 490
line (ssh) Interface Configuration command set 512
line (telnet) Interface Configuration command set 503
line console 490
line ssh 512
line telnet 503
line-length 543
line-timeout 495, 507, 516
lldp 432
lldp receive 608, 774, 871, 935, 1034, 1101
lldp send 609, 775, 936, 1035, 1102
lldp send-and-receive 610, 776, 873, 937, 1036, 1103
local-as 1223
local-id 1174
logging console 433
logging email 434
logging email address-list 436
logging email exception-report address list 435
logging email on 436
logging email priority-level 437
logging email receiver-ip 438
logging email sender 439
logging email source-interface 440
logging facility 441
logging forwarding on 442
logging forwarding priority-level 442, 443
logging forwarding receiver-ip 444
logging forwarding source-interface 445
login 496, 508
login authentication 497, 509, 517
login local-userlist 498, 510, 518
logout 14, 153
loop-alarm-detect 554
Loopback Interface Configuration command set 942
loopback local 535
loopback network 536, 544, 555, 629, 653
loopback remote line 654
loopback remote line inband 545
loopback remote payload 655
loopback remote V54 556
M
mac address-table aging-time 446
mac address-table static bridge 447
mac-address 611
maintenance 537
match 1269
match address 1140, 1149
match community 1183
match ip address 1184
match ip address prefix-list 1185
match ip dscp 1186
match ip precedence 1189
match length 1191
max-channels 1117
maximum paths 1242
maximum-paths 1215
max-reserved-bandwidth 612, 722, 778, 810, 938, 1037
min-channels 1118
modem countrycode 448
mtu 613, 723, 779, 874, 939, 973, 1038, 1104
N
nat-traversal 1175
neighbor 1216
netbios-name-server 1160, 1291
netbios-node-type 1292
network 1217, 1257, 1293
network area 1243
next-hop-self 1224
no enable password 330
ntp-server 1294
O
oam retry 725
oam-pvc managed 724
option 1295
P
parity 499
passive-interface 1258
password 500, 511, 1128, 1225
peer 1176
peer default ip address 780, 1039
permit 1180
ping 1311
point-to-point 339, 341
port-auth supplicant 614
port-auth supplicant enable 616
ppoe ac-name 1048
ppp authentication 781, 1040
ppp chap hostname 785, 1043
ppp chap password 786, 1044
ppp chap sent-username/password 788, 1047
PPP Interface Configuration command set 976
ppp multilink 787, 1045, 1046
pppoe service-name 1049
prefix-list 1226
preventing unauthorized users 6
priority 1275
pvc 726
Q
QoS command set 1265
qos map 451
qos-policy out 727, 811, 1050
R
Radius Group command set 1298
radius-server 452
radius-server host 454
redistribute 1245
redistribute connected 1244, 1259
redistribute ospf 1260
redistribute static 1246, 1261
reload 154
remote-alarm 557, 656
remote-as 1227
remote-loopback 546, 558, 657
resource pool-member 538, 646, 1119
respond 1177
retrain 520
rip 1245
route-map 455, 1228
Router (OSPF) Configuration command set 1235
Router (PIM Sparse) Configuration command set 1249
Router (RIP) Configuration command set 1253
router bgp 456
router ospf 457, 1235
router pim-sparse 458
router rip 459
rp-address 1251
S
sa4tx-bit 559
safe-mode 461
send-community standard 1229
Serial Interface Configuration command set 633
serial-mode 639
serial-number 1129
server 1299, 1301
service password-encryption 462
set comm-list 1194
set community 1195
set default interface 1196
set dscp 1277
set interface 1197
set ip default next-hop 1198
set ip df 1199
set ip dscp 1200
set ip next-hop 1201
set ip precedence 1202
set local-preference 1203
set peer 1142, 1151
set pfs 1143
set precedence 1278
set security-association lifetime 1144
set session-key 1152
set transform-set 1145, 1156
shortcuts 7
show access-lists 155
show arp 156
show atm 157
show backup interfaces 159
show bridge 160
show buffers 161
show buffers users 162
show cflash 163
show clock 15, 164
show configuration 165
show connections 167
show crypto ca 168
show crypto ike 169
show crypto ipsec 171
show crypto map 172
show debugging 173
show demand 174
show dialin interfaces 177
show dynamic-dns 178
show event-history 179, 331
show file 180
show flash 181
show frame-relay 182
show frame-relay fragment 184
show frame-relay multilink 185
show hosts 186
show interfaces 187
show interfaces adsl 190
show ip access-lists 192
show ip arp 193
show ip as-path-list 194
show ip bgp community 195
show ip bgp community-list 197
show ip bgp neighbors 198
show ip bgp regexp 201
show ip bgp summary 203
show ip cache 204
show ip community-list 205
show ip dhcp-client lease 206
show ip dhcp-server binding 207
show ip igmp groups 208
show ip igmp interface 209
show ip interfaces 210
show ip local policy 211
show ip mroute 212
show ip ospf 214
show ip ospf database 215
show ip ospf interface 217
show ip ospf neighbor 218
show ip ospf summary-address 219
show ip pim-sparse 220
show ip policy 224
show ip policy-class 225
show ip policy-sessions 227
show ip policy-stats 229
show ip prefix-list 230
show ip protocols 231
show ip route 232
show ip traffic 234
show isdn group 236
show lldp 237
show lldp device 238
show lldp interface 239
show lldp neighbors interface 240
show lldp neighbors statistics 241
show memory 242
show modules 243
show output-startup 244
show port-auth 245
show pppoe 246
show processes 247
show qos map 248
show queue 251
show queuing 252
show radius statistics 253
show route map 254
show running-config 256, 1313
show sip 258
show sip location 260
show snmp 16, 261
show sntp 262
show spanning-tree, status 263
show startup-config 264
show startup-config checksum 266
show system 267
show tacacs+ statistics 268
show tep info 269
show thresholds 273
show udp info 274
show users 275
show version 17, 276
show version cflash 277
show version flash 277
shutdown 1315
signaling-mode 547
sip check-sync 278
snmp trap 616, 640, 663, 812, 974
snmp trap line-status 560, 658
snmp trap link-status 548, 561, 617, 630, 641, 659, 664, 791, 813, 941, 975, 1051
snmp-server chassis-id 463
snmp-server community 464
snmp-server contact 465
snmp-server enable traps 466
snmp-server host traps 467
snmp-server host traps version 468
snmp-server location 469
snmp-server management-url 470
snmp-server management-url-label 471
snmp-server source-interface 472
snmp-server view 473
snr-margin 521, 522
snntp retry-timeout 474
snntp server 475
snntp wait-time 476
soft-reconfiguration inbound 1230
spanning-tree bpduguard 619, 729, 876
spanning-tree bpduguard filter 478
spanning-tree edgeport 620, 730, 877
spanning-tree forward-time 480
spanning-tree hello-time 481
spanning-tree link-type 621, 731, 878
spanning-tree max-age 482
spanning-tree path-cost 732, 879, 1105
spanning-tree pathcost 484
spanning-tree port-priority 622, 880
spanning-tree priority 485, 733
speed 501, 623
spt-threshold 1252
stopbits 502
subject-name 1130
summary-address not-advertise 1246
T
T1 Interface Configuration command set 647
tacacs 424
TACACS Plus Group Configuration command set 1300
tacacs-server 486
tdm-group 562, 660
telnet 18, 279
terminal length 280
test-call 539
test-pattern 549, 563, 631, 661
tftp-server 1296
thresholds 487
timeout-timer 1262
timers Isagroup-pacing 1247
timers spf 1248
timezone-offset 1297
traceroute 19, 281
traffic-shape rate 624
training-mode 523
ts16 564, 632
tunnel checksum 1106
tunnel destination 1107
tunnel key 1108
tunnel mode gre 1109
tunnel sequence-datagrams 1110
tunnel source 1111
U
unauthorized users 6
undebug all 282
update-source 1231
update-timer 1263
username password 489
V
version 1264
vlan-id 625
VT100 configuration 5

W

cell 283

warranty 2
write 284