# DIGITAL GIGAswitch/Router

# Command Line Interface Reference Manual

Part Number: 9032682-01

**December 1998**

This manual provides reference information for the commands in the DIGITAL GIGAswitch/Router (GSR) Command Line Interface (CLI).

**Revision/Update Information:** This is a revised document.

# Notice

Cabletron Systems reserves the right to make changes in specifications and other information contained in this document without prior notice. The reader should in all cases consult Cabletron Systems to determine whether any such changes have been made.

The hardware, firmware, or software described in this manual is subject to change without notice.

IN NO EVENT SHALL CABLETRON SYSTEMS BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS MANUAL OR THE INFORMATION CONTAINED IN IT, EVEN IF CABLETRON SYSTEMS HAS BEEN ADVISED OF, KNOWN, OR SHOULD HAVE KNOWN, THE POSSIBILITY OF SUCH DAMAGES.



**Cabletron Systems**, **clearVISN, GIGAswitch, SPECTRUM**, and **LANVIEW** are registered trademarks and **HubSTACK**, **SecureFast**, **SmartSwitch**, SmartSwitch Router, and **Synthesis** are trademarks of Cabletron Systems, Inc.

DEC, DIGITAL, and the DIGITAL logo are trademarks of Digital Equipment Corporation.

Java and Solaris are trademarks of Sun Microsystems, Inc.

Netscape Navigator is a registered trademark of Netscape Communications Corp.

Pentium is a registered trademark of Intel Corp.

Windows NT is a trademark and Microsoft, Windows, and Windows 95, are registered trademarks of Microsoft Corp.

HP OpenView is a trademark of Hewlett Packard Company.

All other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

# FCC Notice — Class A Computing Device

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference. Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference. Any modifications to this device - unless expressly approved by the manufacturer - can void the user's authority to operate this equipment under part 15 of the FCC rules.

# Industry Canada Notice

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

# VCCI Notice — Class A Computing Device

This equipment is a Class A product (information equipment to be used in commercial and/or industrial areas) and conforms to the standards set by the Voluntary Control Council for Interference by Data Processing Equipment and Electronic Office Machines aimed at preventing radio interference in commercial and/or industrial areas. Consequently, when used in a residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers. Read the instructions for correct handling.

この装置は，情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

**Taiwanese Notice** — Class A Computing Device:

警告使用者:

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

# CE Notice — Class A Computing Device

**Warning!**
This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may be required to take adequate measures.

**Achtung!**
Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

**Avertissement!**
Cet appareil est un appareil de Classe A. Dans un environnement résidentiel cet appareil peut provoquer des brouillages radioélectriques. Dans ce cas, il peut être demandé à l'utilisateur de prendre les mesures appropriées.

# Cabletron Systems, Inc. Program License Agreement

**IMPORTANT**: Before utilizing this product, carefully read this License Agreement.

This document is an agreement between you, the end user, and Cabletron Systems, Inc. ("Cabletron") that sets forth your rights and obligations with respect to the Cabletron software program (the "Program") contained in this package. The Program may be contained in firmware, chips or other media. BY UTILIZING THE ENCLOSED PRODUCT, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES THE LICENSE AND THE LIMITATION OF WARRANTY AND DISCLAIMER OF LIABILITY. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND.

## Cabletron Software Program License

1. LICENSE. You have the right to use only the one (1) copy of the Program provided in this package subject to the terms and conditions of this License Agreement.

   You may not copy, reproduce or transmit any part of the Program except as permitted by the Copyright Act of the United States or as authorized in writing by Cabletron.

2. OTHER RESTRICTIONS. You may not reverse engineer, decompile, or disassemble the Program.

3. APPLICABLE LAW. This License Agreement shall be interpreted and governed under the laws and in the state and federal courts of New Hampshire. You accept the personal jurisdiction and venue of the New Hampshire courts.

## Exclusion of Warranty and Disclaimer of Liability

1. EXCLUSION OF WARRANTY. Except as may be specifically provided by Cabletron in writing, Cabletron makes no warranty, expressed or implied, concerning the Program (including its documentation and media).

   CABLETRON DISCLAIMS ALL WARRANTIES, OTHER THAN THOSE SUPPLIED TO YOU BY CABLETRON IN WRITING, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE PROGRAM, THE ACCOMPANYING WRITTEN MATERIALS, AND ANY ACCOMPANYING HARDWARE.

2. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. IN NO EVENT SHALL CABLETRON OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS, PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR RELIANCE DAMAGES, OR OTHER LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THIS CABLETRON PRODUCT, EVEN IF CABLETRON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR ON THE DURATION OR LIMITATION OF IMPLIED WARRANTIES, IN SOME INSTANCES THE ABOVE LIMITATIONS AND EXCLUSIONS MAY NOT APPLY TO YOU.

## United States Government Restricted Rights

The enclosed product (a) was developed solely at private expense; (b) contains "restricted computer software" submitted with restricted rights in accordance with Section 52227-19 (a) through (d) of the Commercial Computer Software - Restricted Rights Clause and its successors, and (c) in all respects is proprietary data belonging to Cabletron and/or its suppliers.

For Department of Defense units, the product is licensed with "Restricted Rights" as defined in the DoD Supplement to the Federal Acquisition Regulations, Section 52.227-7013 (c) (1) (ii) and its successors, and use, duplication, disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Cabletron Systems, Inc., 35 Industrial Way, Rochester, New Hampshire 03867-0505.

# SAFETY INFORMATION

## CLASS 1 LASER TRANSCEIVERS

**The DGSRF-AA 100Base-FX Module, DGSRS-AA 1000BASE-SX Module, and DGSRL-AA 1000BASE-LX Module use Class 1 Laser transceivers. Read the following safety information before installing or operating these modules.**

The Class 1 laser transceivers use an optical feedback loop to maintain Class 1 operation limits. This control loop eliminates the need for maintenance checks or adjustments. The output is factory set, and does not allow any user adjustment. Class 1 Laser transceivers comply with the following safety standards:

- 21 CFR 1040.10 and 1040.11 U.S. Department of Health and Human Services (FDA).
- IEC Publication 825 (International Electrotechnical Commission).
- CENELEC EN 60825 (European Committee for Electrotechnical Standardization).

When operating within their performance limitations, laser transceiver output meets the Class 1 accessible emission limit of all three standards. Class 1 levels of laser radiation are not considered hazardous.

### Laser Radiation and Connectors

When the connector is in place, all laser radiation remains within the fiber. The maximum amount of radiant power exiting the fiber (under normal conditions) is -12.6 dBm or $55 \times 10^{-6}$ watts.

Removing the optical connector from the transceiver allows laser radiation to emit directly from the optical port. The maximum radiance from the optical port (under worst case conditions) is $0.8\ W\ cm^{-2}$ or $8 \times 10^{3}\ W\ m^{2}\ sr\text{-}1$.

**Do not use optical instruments to view the laser output. The use of optical instruments to view laser output increases eye hazard. When viewing the output optical port, power must be removed from the network adapter.**

# DECLARATION OF CONFORMITY ADDENDUM

| | |
|---|---|
| Application of Council Directive(s): | **89/336/EEC**<br>**73/23/EEC** |
| Manufacturer's Name: | **Cabletron Systems, Inc.** |
| Manufacturer's Address: | **35 Industrial Way**<br>PO Box 5005<br>Rochester, NH 03867 |
| European Representative Name: | **Mr. J. Solari** |
| European Representative Address: | **Cabletron Systems Limited**<br>**Nexus House, Newbury Business Park**<br>**London Road, Newbury**<br>**Berkshire RG13 2PZ, England** |
| Conformance to Directive(s)/Product Standards: | **EC Directive 89/336/EEC**<br>**EC Directive 73/23/EEC**<br>**EN 55022**<br>**EN 50082-1**<br>**EN 60950** |
| Equipment Type/Environment: | **Networking Equipment, for use in a Commercial or Light Industrial Environment.** |

We the undersigned, hereby declare, under our sole responsibility, that the equipment packaged with this notice conforms _to the above directives.

| Manufacturer | Legal Representative in Europe |
|---|---|
| Mr. Ronald Fotino | Mr. J. Solari |
| Full Name | Full Name |
| Principal Compliance Engineer | Managing Director - E.M.E.A. |
| Title | Title |
| Rochester, NH, USA | Newbury, Berkshire, England |
| Location | Location |

# Contents

# About This Manual

This manual provides reference information for the commands in the DIGITAL GIGAswitch/Router (GSR) Command Line Interface (CLI). For product information not available in this manual, see the manuals listed in "Related Documentation" on page xxii.

**Note:** If you plan to use clearVISN CoreWatch to configure or manage the GSR, see the *DIGITAL clearVISN CoreWatch User's Guide* and the clearVISN CoreWatch online help for information.

## Who Should Read This Manual?

Read this manual if you are a network administrator responsible for configuring or managing the GSR.

## How to Use This Manual

The CLI commands and facilities are organized alphabetically in this manual. To locate information about a command, go to the chapter for the command or for the facility that contains the command. For example, to find information about the **configure** command, go to "configure Command" on page 135. To find information about the **interface add** command, go to "interface Commands" on page 213, then locate the description of the **interface add** command within that chapter.

# Related Documentation

The GSR documentation set includes the following items. Refer to these other documents to learn more about your product.

| For Information About... | See the... |
|---|---|
| Installing and setting up the GSR | *DIGITAL GIGAswitch/Router Getting Started Guide* |
| Managing the GSR using the clearVISN CoreWatch Web-based management application | *DIGITAL clearVISN CoreWatch User's Guide* and the clearVISN CoreWatch online help |
| How to use CLI (Command Line Interface) commands to configure and manage the GSR | *DIGITAL GIGAswitch/Router User Reference Manual* |
| SYSLOG messages and SNMP traps | *DIGITAL GIGAswitch/Router Error Reference Manual* |

# CLI Parameter Types

The following table describes all the parameter types you can use with the CLI.

| Data Type | Descripton | Example |
|---|---|---|
| conditional | A numerical conditional expression. Special symbols are used to describe a numerical condition: > (greater than), < (less than) and != (not equal to). | <1024 or >2048 or !=4096 |
| hexadecimal | A hexadecimal number | a7 or 0xa7 |
| hostname | Hostname of an IP host | gauguin or john-pc |
| hostname/IP | Hostname or IP address of a host | nagasaki or 10.43.1.4 |
| keyword | A keyword described in the list of acceptable keywords in the online help | on or off |
| interface name or IP address | Name of an interface or its IP address | GSR1 or 10.1.4.33 |
| interface name list | A list of one or more interface names delimited by commas | GSR1 or GSR1,GSR2,GSR3 |

| Data Type | Descripton | Example |
|---|---|---|
| IP address | An IP address of the form x.x.x.x. Some commands may explicitly require a unicast or multicast address. | 10.1.2.3 |
| IP address/mask | A pair of IP address and mask values. Depending on the command, the mask may be a network mask or filtering mask. The mask can be described using the traditional IP address syntax (255.0.0.0) or a CIDR syntax (/8). | 10.1.4.0/255.255.255.0 or 10.1.4.0/24 |
| IP address list | A list of IP addresses separated by spaces but enclosed in quotes. | "10.1.4.4 10.1.5.5 10.1.6.6" |
| IPX network address | An IPX network address in hexadecimal | |
| IPX<br><br>network.node address | An IPX network and node address of the form <netaddr>.<macaddr> where <netaddr> is the network address of a host and <macaddr> is the node or MAC address of the IPX host. For some commands, if the node address is not given, the node address is assumed to be a wildcard. | a1b2c3d4.0820a1:f3:38:11<br><br>or aa89f383 |
| IPX SAP<br><br>server name | An alphanumeric string representing a valid IPX SAP server name where the following characters are illegal: "*./:;<=>?[]\ \| | server1 |
| MAC address | A MAC address specified in one of two forms: xx:xx:xx:xx:xx:xx or xxxxxx:xxxxxx | 08:00:50:1a:2b:c3 or 080050:1a2bc3 |
| number | An integer number | 100 |
| numerical range | A number or a range of numbers | 5 or 7-10 |
| port | A single port | et.1.4, gi.2.1, hs.3.1.100, or se.4.2.200 |

| Data Type | Descripton | Example |
|---|---|---|
| port list | A list of one or more ports. To specify a range of ports within a module, describe the range in parenthesis. You can also specify non-consecutive ports by using commas to separate them. The wildcard character (*) can also be used to specify all modules or all ports within a module | et.1.(3-8) or et.1.(1,3,5), hs.(1-2).1.100, or se.4.(1-3).200, gi.2.* |
| slot number | A list of one or more occupied slots in the GSR | 1 or 7 |
| string | A character string. To include spaces in a string, specify the entire string in double quotes (“). | abc or “abc def” |
| URL | A Uniform Resource Locator. The type of URL depends on the command where the URL is used. Currently, two URLs are supported:<br><br>TFTP: *tftp://host/pathname*<br><br>RCP: *rcp://username@host/pathname* | tftp://10.1.4.5/test/abc.txt<br><br>rcp://dave@gsr/test/abc.txt |

# Correspondence

## Documentation Comments

If you have comments or suggestions about this manual, send them to the DIGITAL Network Products Organization as follows:

Attn.:Documentation Project Manager

E-MAIL: doc_quality@lkg.mts.dec.com

## Online Services

To locate product-specific information, refer to the DIGITAL Network Products Home Page on the World Wide Web located at the following addresses:

| | |
|---|---|
| **Americas:** | http://www.networks.digital.com |
| **Europe:** | http://www.networks.europe.digital.com |
| **Asia Pacific:** | http://www.networks.digital.com.au |

# Getting Help

To expedite your inquiry when you contact your DIGITAL representative, please provide the following information:

- Your Name
- Your Company Name
- Address
- Email Address
- Phone Number
- FAX Number
- Detailed description of the issue (including history, what you've tried, and conditions under which you see this occur)
- Hardware module number, software version, and switch configuration (that is, what part types are in what slots)

# Chapter 1

# acl Commands

The acl commands allow you to create ACLs (Access Control Lists) and apply them to IP and IPX interfaces on the GSR. An ACL permits or denies switching of packets based on criteria such as the packet's source address and destination address, TCP or UDP port number, and so on. When you apply an ACL to an interface, you can specify whether the ACL affects incoming traffic or outgoing traffic. You also can enable a log of the ACL's use.

## Command Summary

Table 1 lists the acl commands. The sections following the table describe the command syntax.

**Table 1. acl commands**

| |
|---|
| **acl** <*name*> **apply interface** <*InterfaceName*> **input** \| **output [logging [on** \| **off]]** |
| **acl** <*name*> **apply service** <*ServiceName*> **[logging [on** \| **off]]** |
| **acl** <*name*> **permit** \| **deny icmp** <*SrcAddr/Mask*> <*DstAddr/Mask*> |
| **acl** <*name*> **permit** \| **deny igmp** <*SrcAddr/Mask*> <*DstIP/mask*> |
| **acl** <*name*> **permit** \| **deny ip** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> |
| **acl** <*name*> **permit** \| **deny ip-protocol** <*proto-num*> <*SrcAddr/Mask*> <*DstAddr/Mask*> <*tos*> |
| **acl** <*name*> **permit** \| **deny ipx** <*SrcAddr*> <*SrcSocket*> <*DstAddr*> <*DstSocket*> <*SrcNetMask*> <*DstNetMask*> |
| **acl** <*name*> **permit** \| **deny ipxgns** <*ServerAddr*> <*ServiceType*> <*ServiceName*> |

**Table 1. acl commands (Continued)**

| |
|---|
| **acl** <*name*> **permit** \| **deny ipxrip** <*FromNetwork*> <*ToNetwork*> |
| **acl** <*name*> **permit** \| **deny ipxsap** <*ServerAddr*> <*ServiceType*> <*ServiceName*> |
| **acl** <*name*> **permit** \| **deny ipxtype20** |
| **acl** <*name*> **permit** \| **deny tcp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> |
| **acl** <*name*> **permit** \| **deny udp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> |
| **acl-policy enable external** |

# acl apply interface

## Purpose

Apply an ACL to an interface.

## Format

**acl** <*name*> **apply interface** <*InterfaceName*> **input | output [logging [on | off]]**

## Mode

Configure

## Description

The **acl apply interface** command applies a previously defined ACL to an interface. When you apply an ACL to an interface, you implicitly enable access control on that interface. You can apply an ACL to filter out inbound traffic, outbound traffic, or both inbound and outbound traffic. Inbound traffic is packets coming into the interface while outbound traffic is packets going out of that interface.

When you apply an ACL, you also can enable ACL Logging by using the **logging** keyword. When you enable ACL Logging on an interface, the GSR displays ACL Logging messages on the console. The ACL log provides information such as the interface name, the ACL name, whether the packet is forwarded or not, and the internal details of the packet.

## Parameters

| | |
|---|---|
| <*name*> | Name of the ACL. The ACL must already be defined. To define an ACL, use one of the commands described in the previous sections in this chapter. |
| <*InterfaceName*> | Name of the interface to which you are applying the ACL. |
| **input** | Applies the ACL to filter out inbound traffic. |
| **output** | Applies the ACL to filter out outbound traffic. |

**logging [on | off]** Enables or disables ACL logging for this interface. You can specify one of the following keywords:

**off** Disables logging.

**on** Enables logging.

## Restrictions

You can apply only one ACL of each type (IP or IPX) to an interface at one time. For example, although you can define two ACLs, "ipacl1" and "ipacl2", you cannot apply them both to the same interface.

You can apply IP ACLs only to IP interfaces. Likewise, you can apply IPX ACLs only to IPX interfaces.

## Examples

To apply ACL "100" to interface gs/r*4* to filter out inbound traffic:

```
gs/r(config)# acl 100 apply interface gs/r4 input
```

To apply ACL "nonfs" to interface gs/r *16* to filter out outbound traffic and enable logging:

```
gs/r(config)# acl nonfs apply interface gs/r16 output logging on
```

# acl apply service

## Purpose

Apply an ACL to a service on the GSR.

## Format

**acl** <*name*> **apply service** <*ServiceName*> **[logging [on | off]]**

## Mode

Configure

## Description

The **acl apply service** command applies a previously defined ACL to a service provided by the GSR. A service is typically a server or agent running on the GSR, for example, a Telnet server or SNMP agent. By applying an ACL to a service, you can control which host can access individual services on the GSR. This type of ACL is known as a Service ACL. It does not control packets going *through* the GSR. It only controls packets that are *destined* for the GSR, specifically, one of the services provided by the GSR. As a result, a Service ACL, by definition, is applied only to check for inbound traffic to the GSR. In addition, if a Service ACL is defined with destination address and port information, that information is ignored. The destination host of a Service ACL is by definition the GSR. The destination port is the well-known port of the service.

When you apply an ACL, you also can enable ACL Logging by using the **logging** keyword. When you enable ACL Logging on an interface, the GSR displays ACL Logging messages on the console. The ACL log provides information such as the interface name, the ACL name, whether the packet is forwarded or not, and the internal details of the packet.

## Parameters

| | |
|---|---|
| <*name*> | Name of the Service ACL. The ACL must already be defined. To define an ACL, use one of the commands described in the previous sections in this chapter. |
| <*ServiceName*> | Name of the service on the GSR to which you are applying the ACL. Currently, the following services are supported: |

**http** HTTP web server

**snmp** SNMP agent

**telnet** Telnet server

**[logging [on | off]]** Enables or disables ACL logging for this interface. You can specify one of the following keywords:

**off** Disables logging.

**on** Enables logging.

## Restrictions

You can apply only one ACL of each type (IP or IPX) to a service at one time. For example, although you can define two ACLs, "ipacl1" and "ipacl2", you cannot apply them both to the same service.

## Examples

To permit access to the SNMP agent only from the host 10.4.3.33 (presumably an SNMP management station):

```
gs/r(config)# acl 100 permit udp 10.4.3.33
gs/r(config)# acl 100 apply service snmp
```

The following commands permit access to the Telnet server from hosts on the subnet 10.4.7.0/24 with a privileged source port. In addition, with logging enabled, all incoming Telnet accesses are logged to the console.

```
gs/r(config)# acl 120 permit tcp 10.4.7.0/24 <1024
gs/r(config)# acl 120 apply service telnet logging on
```

The following commands permit access to the HTTP web server from subnet 10.12.4.0/24. Notice that even though the destination address and port are specified for this ACL (*10.12.7.44* and *any* port), they are ignored. This service ACL will match only packets destined for the GSR itself and the well-known port of the service (port 80 for HTTP).

```
gs/r(config)# acl 140 permit ip 10.12.4.0/24 any 10.12.7.44 any
gs/r(config)# acl 120 apply service http
```

# acl permit|deny icmp

## Purpose

Create an ICMP ACL.

## Format

**acl** <*name*> **permit | deny icmp** <*SrcAddr/Mask*> <*DstAddr/Mask*>

## Mode

Configure

## Description

The **acl permit icmp** and **acl deny icmp** commands define an ACL to allow or block ICMP traffic from entering or leaving the GSR. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the GSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*SrcAddr/Mask*> | The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16"). |
| <*DstAddr/Mask*> | The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>. |

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To deny ICMP traffic from the subnet 10.24.5.0 (with a 24 bit netmask) to any destination:

```
gs/r(config)# acl 310 deny icmp 10.24.5.0/24 any
```

To create an ACL to permit ICMP traffic from the host 10.12.28.44 to subnet 10.43.21.0:

```
gs/r(config)# acl 312 permit icmp 10.12.28.44 10.43.21.0/24
```

# acl permit|deny igmp

## Purpose

Create an IGMP ACL.

## Format

**acl** <*name*> **permit | deny igmp** <*SrcAddr/Mask*> <*DstAddr/Mask*>

## Mode

Configure

## Description

The **acl permit igmp** and **acl deny igmp** commands define an ACL to allow or block IGMP traffic from entering or leaving the GSR. For each of the values describing a flow, you can use the keyword **any** to specify a wildcard ("don't care") condition. If you do not specify a value for a field, the GSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword.

<*name*> Name of this ACL. You can use a string of characters or a number.

<*SrcAddr/Mask*> The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").

<*DstAddr/Mask*> The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To create an ACL to deny IGMP traffic from the subnet 10.1.5.0 (with a 24 bit netmask) to any destination:

```
gs/r(config)# acl 410 deny igmp 10.1.5.0/24 any
```

To create an ACL to permit IGMP traffic from the host 10.33.34.44 to subnet 10.11.21.0:

```
gs/r(config)# acl 714 permit igmp 10.33.34.44 10.11.21.0/24
```

# acl permit|deny ip

## Purpose

Create an IP ACL.

## Format

**acl <***name***> permit | deny ip** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*>

## Mode

Configure

## Description

The **acl permit ip** and **acl deny ip** commands define an Access Control List to allow or block IP traffic from entering or leaving the router. Unlike the more specific variants of the acl commands for **tcp** and **udp**, the **ip** version of the command includes IP-based protocols such as **tcp**, **udp**, **icmp** and **igmp**. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the GSR assumes that the value is a wildcard (as if you had specified the **any** keyword).

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*SrcAddr/Mask*> | The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16"). |
| <*DstAddr/Mask*> | The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>. |
| <*SrcPort*> | For TCP or UDP, the number of the source TCP or UDP port. This field applies only to TCP or UDP traffic. If the incoming packet is ICMP or |

another non-TCP or non-UDP packet and you specified a source or destination port, the GSR does not check the port value. The GSR checks only the source and destination IP addresses in the packet.

You can specify a range of port numbers using operator symbols; for example, 10-20 (between 10 and 20 inclusive), >1024 (greater than 1024), <1024 (less than 1024), !=1024 (not equal to 1024). The port numbers of some popular services are already defined as keywords. For example, for Telnet, you can enter the port number 23 as well as the keyword **telnet**.

<*DstPort*> For TCP or UDP, the number of the destination TCP or UDP port. This field applies only to incoming TCP or UDP traffic. The same requirements and restrictions for <*SrcPort*> apply to <*DstPort*>.

<*tos*> IP TOS (Type of Service) value. You can specify a TOS from 0 – 15.

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To create an ACL to permit IP traffic from the subnet 10.1.0.0 (with a 16 bit netmask) to any destination:

```
gs/r(config)# acl 100 permit ip 10.1.0.0/16 any
```

The following command creates an ACL to deny any incoming TCP or UDP traffic coming from a privileged port (less than 1024). If the incoming traffic is not TCP or UDP, then the GSR check only the source and destination addresses, not the port number. Therefore, this ACL will deny all non-TCP and non-UDP traffic.

```
gs/r(config)# acl 120 deny ip any any 1-1024 any
```

To create an ACL to permit Telnet traffic (port 23) from the host 10.23.4.8 to the subnet 10.2.3.0:

```
gs/r(config)# acl 130 permit ip 10.23.4.8 10.2.3.0/24
```

The following command creates an ACL to permit all IP traffic. Since none of the ACL fields are specified, they are all assumed to be wildcards.

```
gs/r(config)# acl allip permit ip
```

The above command is equivalent to the following:

```
gs/r(config)# acl allip permit ip any any any any any
```

# acl permit|deny ip-protocol

## Purpose

Create an ACL for any IP protocol type.

## Format

**acl <***name***> permit | deny ip-protocol** <*proto-num*> <*SrcAddr/Mask*> <*DstAddr/Mask*> <*tos*>

## Mode

Configure

## Description

The **acl permit ip-protocol** and **acl deny ip-protocol** commands define an Access Control List to allow or block IP traffic from entering or leaving the router for any protocol type. Unlike the more specific variants of the acl commands such as **ip**, **tcp** and **udp**, the **ip-protocol** version of the command allows the user to specify any valid IP protocol type. This command allows the user to specify an IP protocol other than the ones available with other **acl permit | deny** commands. For example, to specify an ACL for IP encapsulation in IP, one can use the IPinIP protocol type, 4, in the ACL. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the GSR assumes that the value is a wildcard (as if you had specified the **any** keyword).

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*proto-num*> | IP protocol number of this flow. |
| <*SrcAddr/Mask*> | The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16"). |

<*DstAddr/Mask*> The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.

<*tos*> IP TOS (Type of Service) value. You can specify a TOS from 0 – 15.

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To create an ACL to permit VRRP traffic (IP protocol type 112) from the subnet 10.14.0.0 (with a 16 bit netmask) to any destination:

```
gs/r(config)# acl 100 permit ip-protocol 112 10.14.0.0/16 any
```

The following command has the same function as **acl 120 deny igmp** since the protocol type for IGMP is 2.

```
gs/r(config)# acl 120 deny ip-protocol 2
```

# acl permit|deny ipx

## Purpose

Create an IPX ACL.

## Format

**acl** <*name*> **permit | deny ipx** <*SrcAddr*> <*SrcSocket*> <*DstAddr*> <*DstSocket*> <*SrcNetMask*> <*DstNetMask*>

## Mode

Configure

## Description

The **acl permit ipx** and **acl deny ipx** commands define an ACL to allow or block IPX traffic from entering or leaving the GSR.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*SrcAddr*> | The source IPX address in <*network*>**.**<*node*> format, where <*network*> *is the network address and* <*node*> is the MAC address. The GSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition.<br><br>To specify any network, enter **FFFFFFFF**.<*node*>; to specify any node, enter <*network*>.**FF:FF:FF:FF:FF:FF**. |
| <*SrcSocket*> | Source IPX socket. The GSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition. |
| <*DstAddr*> | The destination IPX address in <*network*>**.**<*node*> format. The syntax for the destination address is the same as the syntax for the source address <*SrcAddr*>. The GSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition. |

<*DstSocket*> Destination IPX socket. The GSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition.

<*SrcNetmask*> Source network mask. This field specifies a group of networks for which the ACL applies. This mask field is ANDed with the network portion of <*SrcAddr*> and the source network of the incoming packets to determine a hit. The GSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix.

This is an optional argument and if you omit the argument, the GSR uses the hexadecimal value FFFFFFFF.

<*DstNetmask*> Destination network mask. This field specifies a group of networks for which the ACL applies. This mask field is ANDed with the network portion of <*DstAddr*> and the destination network of the incoming packets to determine a hit. The GSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix.

This is an optional argument and if you omit the argument, the GSR uses the hexadecimal value FFFFFFFF.

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

The following command creates an ACL to permit IPX traffic from the host with IPX address AAAAAAAA.01:20:0A:F3:24:6D, any socket, to any other IPX address (network.node), any socket.

```
gs/r(config)# acl 100 permit ipx AAAAAAAA.01:20:0A:F3:24:6D any any any
```

The following command creates an ACL to deny IPX traffic from the host with IPX address F6D5E4.01:20:0A:F3:24:6D, with socket address 451, to any other IPX address (network.node), any socket.

```
gs/r(config)# acl 200 deny ipx F6D5E4.01:20:0A:F3:24:6D 451 any any
```

# acl permit|deny ipxgns

## Purpose

Create an IPX GNS (Get Nearest Server) ACL.

## Format

**acl** <*name*> **permit | deny ipxgns** <*ServerAddr*> <*ServiceType*> <*ServiceName*>

## Mode

Configure

## Description

The **acl permit ipxgns** and **acl deny ipxgns** commands define an ACL to allow or block replying to GNS requests.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*ServerAddr*> | The SAP server's IPX address in <*network*>**.**<*node*> format, where <*network*> *is the network address and* <*node*> is the MAC address. You can use the keyword **any** to specify a wildcard ("don't care") condition. |
| <*ServiceType*> | The SAP service type. Express the service type in hexadecimal. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition. |
| <*ServiceName*> | The SAP service name. This is an optional argument and if you omit the argument, the GSR applies a wildcard condition to the field. |

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit

all traffic. You can only apply the **acl permit ipxgns** and **acl deny ipxgns** commands to output.

## Examples

To create a GNS ACL to permit the GSR to reply with the server "FILESERVER", whose IPX address is F6D5E4.01:20:0A:F3:24:5D, to get nearest server requests:

```
gs/r(config)# acl 100 permit ipxgns F6D5E4.01:20:0A:F3:24:5D 0004
FILESERVER
```

To create a GNS ACL to prevent the GSR from replying with the server "ARCHIVESERVER", whose IPX address is F6D5E4.01:20:0A:F3:24:5C, to a get nearest server request:

```
gs/r(config)# acl 200 deny ipxgns F6D5E4.01:20:0A:F3:24:5C 0009
ARCHIVESERVER
```

# acl permit|deny ipxrip

## Purpose

Create an IPX RIP (Route Information Protocol) ACL.

## Format

**acl** <*name*> **permit | deny ipxrip** <*FromNetwork*> <*ToNetwork*>

## Mode

Configure

## Description

The **acl permit ipxrip** and **acl deny ipxrip** commands define an ACL to allow or block IPX RIP traffic from entering or leaving the GSR.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*FromNetwork*> | The "from" IPX network address. You can use the **any** keyword to specify a wildcard condition. If you use **any**, the GSR uses the value 0 for <*FromNetwork*> *and FFFFFFFE for* <*ToNetwork*>. |
| <*ToNetwork*> | The "to" IPX network address. This is an optional parameter. If you omit this parameter, the value that the GSR assumes depends on whether you specified **any** for <*FromNetwork*>.<br>–If you omit the <*ToNetwork*> value and you used the value **any** for <*FromNetwork*>, the GSR sets the <*ToNetwork*> to FFFFFFFE.<br>–If you omit the <*ToNetwork*> value but do not use the value **any** for <*FromNetwork*>, the GSR sets <*ToNetwork*> to the same value you specified for <*FromNetwork*>. |

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To create an ACL to permit IPX RIP traffic from networks AA000001 to AAFFFFFF:

```
gs/r(config)# acl 100 permit ipxrip AA000001 AAFFFFFF
```

# acl permit|deny ipxsap

## Purpose

Create an IPX SAP (Service Advertisement Protocol) ACL.

## Format

**acl** <*name*> **permit | deny ipxsap** <*ServerAddr*> <*ServiceType*> <*ServiceName*>

## Mode

Configure

## Description

The **acl permit ipxsap** and **acl deny ipxsap** commands define an ACL to allow or block IPX SAP traffic from entering or leaving the GSR.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*ServerAddr*> | The SAP server's IPX address in <*network*>**.**<*node*> format, where <*network*> *is the network address and* <*node*> is the MAC address. You can use the keyword **any** to specify a wildcard ("don't care") condition.<br><br>To specify any network, enter **FFFFFFFF**.<*node*>; to specify any node, enter <*network*>.**FF:FF:FF:FF:FF:FF**. |
| <*ServiceType*> | The SAP service type. Express the service type in hexadecimal. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition. |
| <*ServiceName*> | The SAP service name. This is an optional argument and if you omit the argument, the GSR applies a wildcard condition to the field. |

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To create a SAP ACL to permit SAP information related to the server "FILESERVER" whose IPX address is F6D5E4.01:20:0A:F3:24:5D:

```
gs/r(config)# acl 100 permit ipxsap F6D5E4.01:20:0A:F3:24:5D 0004
FILESERVER
```

To create a SAP ACL to deny SAP information related to the server "ARCHIVESERVER" whose IPX address is F6D5E4.01:20:0A:F3:24:5C:

```
gs/r(config)# acl 200 deny ipxsap F6D5E4.01:20:0A:F3:24:5C 0009
ARCHIVESERVER
```

# acl permit|deny ipxtype20

## Purpose

Create an IPX type 20 ACL.

## Format

**acl** <*name*> **permit | deny ipxtype20**

## Mode

Configure

## Description

The **acl permit ipxtype20** and **acl deny ipxtype20** commands define an ACL to allow or block IPX type 20 packets from entering or leaving the GSR.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To create an ACL to deny IPX type 20 packets:

```
gs/r(config)# acl 100 deny ipxtype20
```

# acl permit|deny tcp

## Purpose

Create a TCP ACL.

## Format

**acl** <*name*> **permit | deny tcp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*>

## Mode

Configure

## Description

The **acl permit tcp** and **acl deny tcp** commands define an ACL to allow or block TCP traffic from entering or leaving the GSR. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the GSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*SrcAddr/Mask*> | The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16"). |
| <*DstAddr/Mask*> | The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>. |
| <*SrcPort*> | For TCP or UDP, the number of the source TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* You can specify a range of port numbers using operator symbols; for example, 10-20 (between 10 and 20 inclusive), >1024 (greater than 1024), <1024 (less than 1024), |

| | |
|---|---|
| | !=1024 (not equal to 1024). The port numbers of some popular services are already defined as keywords. For example, for Telnet, you can enter the port number 23 as well as the keyword **telnet**. |
| <*DstPort*> | For TCP or UDP, the number of the destination TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* The same requirements and restrictions for <*SrcPort*> apply to <*DstPort*>. |
| <*tos*> | IP TOS (Type of Service) value. You can specify a TOS from 0 – 15. |

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

To create an ACL to permit TCP traffic from the subnet 10.21.33.0 (with a 24 bit netmask) to any destination:

```
gs/r(config)# acl 100 permit tcp 10.21.33.0/255.255.255.0 any
```

To create an ACL to deny any incoming HTTP traffic:

```
gs/r(config)# acl noweb deny tcp any any http any
```

To create an ACL to permit FTP traffic (both command and data ports) from subnet 10.31.34.0 to 10.31.60.0:

```
gs/r(config)# acl ftp100 permit tcp 10.31.34.0/24 10.31.60.0/24 20-21
any
```

# acl permit|deny udp

## Purpose

Create a UDP ACL.

## Format

**acl** <*name*> **permit | deny udp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*>

## Mode

Configure

## Description

The **acl permit udp** and **acl deny udp** commands define an ACL to allow or block UDP traffic from entering or leaving the GSR. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the GSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword.

## Parameters

| | |
|---|---|
| <*name*> | Name of this ACL. You can use a string of characters or a number. |
| <*SrcAddr/Mask*> | The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16"). |
| <*DstAddr/Mask*> | The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>. |
| <*SrcPort*> | For TCP or UDP, the number of the source TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* You can specify a range of port numbers using operator symbols; for example, 10-20 (between 10 |

| | |
|---|---|
| | and 20 inclusive), >1024 (greater than 1024), <1024 (les than 1024), !=1024 (not equal to 1024). The port numbers of some popular services are already defined as keywords. For example, for Telnet, you can enter the port number 23 as well as the keyword **telnet**. |
| <*DstPort*> | For TCP or UDP, the number of the destination TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* The same requirements and restrictions for <*SrcPort*> apply to <*DstPort*>. |
| <*tos*> | IP TOS (Type of Service) value. You can specify a TOS from 0 – 15. |

## Restrictions

When you apply an ACL to an interface, the GSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

## Examples

Here are some examples of ACL commands for permitting and denying UDP traffic flows.

```
gs/r(config)# acl 100 permit udp 10.1.3.0/24 any
```

Creates an ACL to permit UDP traffic from the subnet 10.1.3.0 (with a 24 bit netmask) to any destination.

```
gs/r(config)# acl notftp deny udp any any tftp any
```

Creates an ACL to deny any incoming TFTP traffic.

```
gs/r(config)# acl udpnfs permit udp 10.12.0.0/16 10.7.0.0/16 any nfs
```

Creates an ACL to permit UDP based NFS traffic from subnet 10.12.0.0 to subnet 10.7.0.0.

# acl-policy enable external

## Purpose

Allow an external server to create and delete ACLs.

## Format

**acl-policy enable external**

## Mode

Configure

## Description

The **acl-policy enable external** command allows ACLs to be configured by an external agent, such as the Policy Manager. If this command is in the active configuration, an external server can create, modify, and delete ACLs on the GSR. If this command is not in the active configuration, then ACLs can only be created, modified, and deleted using the CLI.

## Parameters

None.

## Restrictions

The only action allowed by the **acl-policy enable external** command is to allow an external server to create, modify, and delete ACLs. Once entered, this command must be negated in order to prohibit an external server from creating, altering, or deleting ACLs. An external server can only modify ACLs that it created, or ACLs that were created using the CLI with the “external” flag. It cannot modify an ACL that was created using the CLI with the “local” flag.

# Chapter 2

# acl-edit Commands

The acl-edit command activates the ACL Editor mode. The ACL Editor provides a user-friendly interface for maintaining and manipulating rules in an ACL. Using the editor, you can add, delete or re-order ACL rules. In addition, if the modified ACL is currently applied to an interface, the ACL is automatically "re-applied" to the interface and takes effect immediately. To edit an ACL, you enter the **acl-edit** command in Configure mode. The command must also specify the name of the ACL you want to edit. Only one ACL can be edited at one time.

## Command Summary

Table 2 lists the commands available with the ACL Editor. The sections following the table describe the command syntax.

**Table 2. acl-edit commands**

| |
|---|
| **acl-edit** <*aclname*> |
| **acl permit** \| **deny** |
| **delete** <*rule#*> |
| **exit** |
| **move** <*rule#*> **after** <*rule#*> |
| **save** |
| **show** |

# acl-edit

## Purpose

Enter ACL Editor to edit the specified ACL.

## Format

**acl-edit** <*aclname*>

## Mode

Configure

## Description

The **acl-edit** command enters the ACL Editor to edit an ACL specified by the user. Once inside the ACL editor, the user can then add, delete or re-order ACL rules for that ACL. If the ACL happens to be applied to an interface, changes made to that ACL will automatically take effect when the changes are committed to the running system.

## Parameters

<*aclname*> Name of the ACL to edit.

## Restrictions

Inside the ACL Editor, you can only add rules for the ACL you specified in the acl-edit command. You cannot add rules for other ACLs. Basically, each ACL editing session works only on one ACL at a time. For example, if you start with *acl-edit 110*, you cannot add rules for ACL *121*.

## Example

To edit ACL 111:

```
gs/r(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any

gs/r(acl-edit)> ?
acl                     - Configure L3 Access Control List
delete                  - Delete an ACL rule
exit                    - Exit current mode
move                    - Move an ACL rule
save                    - Save changes made to this ACL
show                    - Show contents of this ACL
```

# acl permit|deny

## Purpose

Create an ACL rule to permit or deny traffic.

## Format

**acl** <*name*> **permit | deny**

## Mode

ACL Editor

## Description

The **acl permit | deny** commands are equivalent to the same commands in the Configuration mode. You can use these commands to create rules for the ACL that you are editing. Just like the acl commands in Configuration mode, new rules are appended to the end of the rules. You can use the move command to re-order the rules.

## Restrictions

You can only add rules for the ACL you specified in the acl-edit command. You cannot add rules for other ACLs. For example, if you start with *acl-edit 110*, you cannot add rules for ACL *121*.

## Example

To add a new rule (deny all UDP traffic) to ACL 111:

```
gs/r(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any

gs/r(acl-edit)> acl 111 deny udp
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp
```

# delete

## Purpose

Deletes a rule from an ACL.

## Format

**delete** <*rule#*>

## Mode

ACL Editor

## Description

The **delete** commands allows the administrator to delete a specific rule from an ACL. When in the ACL Editor, each rule is displayed with its rule number. One can delete a specific rule from an ACL by specifying its rule number with the delete command.

## Parameters

<*rule#*> Number of the ACL rule to delete.

## Restrictions

None

## Example

To delete ACL rule number 2 from the ACL:

```
gs/r(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp

gs/r(acl-edit)> delete 2
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 deny udp
```

# exit

## Purpose

Exit ACL Editor.

## Format

**exit**

## Mode

ACL Editor

## Description

The **exit** command allows the user to exit the ACL Editor. Before exiting, if changes are made to this ACL, the system will prompt the user to see if the changes should be committed to the running system or discarded. If the user commits the changes then changes made to this ACL will take effect immediately. If the ACL is applied to an interface, the ACL is automatically re-applied to the interface. Packets going through this interface will be matched against the new rules in this ACL. If the user chooses not to commit the changes, the changes will be discarded. The next time the user edits this ACL, changes from the previous edit session will be lost.

## Parameters

None

## Restrictions

None

## Example

To create an ACL to deny IGMP traffic from the subnet 10.1.5.0 (with a 24 bit netmask) to any destination:

```
gs/r(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp

gs/r(acl-edit)> delete 2
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 deny udp

gs/r(acl-edit)> exit

gs/r(config)# acl 410 deny igmp 10.1.5.0/24 any
```

# move

## Purpose

Re-order ACL rules by moving a rule to another position.

## Format

**move** <*src-rule#*> **after** <*dst-rule#*>

## Mode

ACL Editor

## Description

The **move** command provides the user with the ability to re-order rules within an ACL. When new rules are entered in the ACL Editor, they are appended to the end of the rules. One can move these rules to the desired location by using the move command. The move command can also be used on existing ACL rules created in Configuration mode instead of the ACL Editor.

## Parameters

<*src-rule#*> Rule number of the rule you want to move.

<*dst-rule#*> Rule number of the rule after which you want the source rule to move to.

## Restrictions

None

## Examples

To move rule #2 to the end of the list:

```
gs/r(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 permit udp 10.1.17.0/24 10.1.22.0/24 2000-2002 any
4*: acl 111 permit udp 10.1.18.0/24 10.1.34.0/24 2003-2005 any

gs/r(acl-edit)> move 2 after 4
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit udp 10.1.17.0/24 10.1.22.0/24 2000-2002 any
3*: acl 111 permit udp 10.1.18.0/24 10.1.34.0/24 2003-2005 any
4*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
```

# save

## Purpose

Save any changes made by the ACL Editor.

## Format

**save**

## Mode

ACL Editor

## Description

The **save** command saves any non-committed changes made by the ACL Editor. If changes are made to this ACL, the changes will be saved and will take effect immediately. If the ACL is applied to an interface, the ACL is automatically re-applied to the interface. Packets going through this interface will be matched against the new rules in this ACL. The **save** command also contains an implicit exit command. Regardless of whether changes were made by the ACL Editor or not, upon completion of the **save** command, the user exits the ACL Editor and returns to Configuration mode. Consequently, one should issue the **save** command after all the changes are made.

## Parameters

None

## Restrictions

None

## Examples

To save and commit the changes made by the ACL Editor.

```
gs/r(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp

gs/r(acl-edit)> delete 2
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 deny udp

gs/r(acl-edit)> save
```

# show

## Purpose

Displays the contents of the ACL in the current editing session.

## Format

**show**

## Mode

ACL Editor

## Description

The **show** command displays the contents of the ACL currently being edited.

## Parameters

*None*

## Restrictions

None

## Examples

To display the contents of the ACL currently being edited:

```
gs/r(acl-edit)# show
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
```

# Chapter 3

# aging Commands

The aging commands control aging of learned MAC address entries in the GSR's L2 lookup tables. Using the aging commands, you can show L2 aging information, disable L2 aging on specific ports, and set the aging time on specific ports.

## Command Summary

Table 3 lists the L2 aging commands. The sections following the table describe the command syntax.

**Table 3. aging commands**

| |
|---|
| **aging l2 disable** <*port-list*> \| **all-ports** |
| **aging l2 set aging-timeout** <*seconds*> **port** <*port-list*> \| **all-ports** |
| **aging l2 show status** |

# aging l2 disable

## Purpose

Disable aging of MAC addresses.

## Format

**aging l2 disable** <*port-list*> | **all-ports**

## Mode

Configure

## Description

By default, the GSR ages learned MAC addresses in the L2 lookup tables. Each port has its own L2 lookup table. When a learned entry ages out, the GSR removes the aged out entry. You can disable this behavior by disabling aging on all ports or on specific ports.

## Parameters

<*port-list*> | **all-ports**
: The port(s) on which you want to disable aging. You can specify a single port or a comma-separated list of ports. If you use the **all-ports** keyword, aging is disabled on all ports.

## Restrictions

Unknown.

## Examples

To disable aging on slot 1, port 3:

```
gs/r(config)# aging l2 disable et.1.3
```

To disable aging on slot 4, port 2, and slots 1 through 3, ports 4, 6, 7, and 8:

```
gs/r(config)# aging l2 disable et.4.2 et.(1-3).(4 6-8)
```

To disable aging on all ports:

```
gs/r(config)# aging l2 disable all-ports
```

# aging l2 set aging-timeout

## Purpose

Set the aging time for learned MAC entries.

## Format

**aging l2 set** <*port-list*> | **all-ports aging-timeout** <*seconds*>

## Mode

Configure

## Description

The **aging l2 set aging-timeout** command sets the aging time for learned MAC entries. When the aging time expires for a MAC address, the GSR removes the MAC address from the specified port(s). The aging time is specified in seconds.

## Parameters

<*port-list*> | **all-ports**
: The port(s) on which you want to set the aging time. You can specify a single port or a comma-separated list of ports. If you use the **all-ports** keyword, the aging time is set on all ports.

<*seconds*>
: The number of seconds the GSR allows a learned MAC address to remain in the L2 lookup table (for the specified port). You can specify from 15 to 1000000 seconds. The default is 300 seconds.

## Restrictions

None.

## Example

To set the aging time to 15 seconds on all ports:

```
gs/r(config)# aging l2 set all-ports aging-timeout 15
```

# aging l2 show status

## Purpose

Show the L2 aging status for GSR ports.

## Format

aging l2 show status

## Mode

User

## Description

The **aging l2 show status** command shows whether L2 aging is enabled or disabled on GSR ports. For ports on which L2 aging is enabled, this command also shows the aging time.

## Parameters

None.

## Restrictions

None.

# Chapter 4

# arp Commands

The **arp** commands enable you to add, display, and clear ARP entries on the GSR.

## Command Summary

Table 4 lists the arp commands. The sections following the table describe the command syntax.

**Table 4. arp commands**

| |
|---|
| **arp add** <*host*> **mac-addr** <*MAC-addr*> **exit-port** <*port*> **keep-time** <*seconds*> |
| **arp clear** <*host*> \| **all** |
| **arp set interface** <*name*> \| **all keep-time** <*number*> |
| **arp show** <*IPaddr*> \| **all** |
| **statistics show arp** |

# arp add

## Purpose

Add an ARP entry.

## Format

**arp add** <*host*> **mac-addr** <*MAC-addr*> **exit-port** <*port*> **keep-time** <*seconds*>

## Mode

Enable and Configure

## Description

The **arp add** command lets you manually add ARP entries to the ARP table. Typically, the GSR creates ARP entries dynamically. Using the **arp add** command, you can create an ARP entry to last a specific amount of time or as a permanent ARP entry. This command exists in both Enable and Configure mode with a slight variation. The **keep-time** option is valid only in Enable mode. The **keep-time** option allows you to create an ARP entry to last a specific amount of time. The Configure mode version of the **arp add** command does not use the **keep-time** option. ARP entries created in the Configure mode are permanent ARP entries and they do not have an expiration time. If the exit port is not specified, then packets to the IP address for which the ARP entry is created are transmitted on all ports of the interface. If an ARP request is received from the host for which the ARP entry was created, then the exit port is updated with the port on which the ARP request was received, so that subsequent packets are transmitted on one port only.

## Parameters

| | |
|---|---|
| <*host*> | Hostname or IP address of this ARP entry. |
| **mac-addr** <*MAC-addr*> | MAC address of the host. |
| **exit-port** <*port*> | The port for which you are adding the entry. Specify the port to which the host is connected. |
| **keep-time** <*seconds*> | The number of seconds this ARP entry should remain in the ARP table. A value of 0 means this is a permanent ARP entry. |

**Note:** This option is valid only for the Enable mode **arp add** command.

## Restrictions

If you enter the **arp add** command while in the Configure mode, you can add only permanent ARP entries.

## Examples

To create an ARP entry for the IP address 10.8.1.2 at port et.4.7 for 15 seconds:

```
gs/r# arp add 10.8.1.2 mac-addr 08:00:20:a2:f3:49 exit-port et.4.7 keep-time
15
```

To create a permanent ARP entry for the host *nfs2* at port et.3.1:

```
gs/r(config)# arp add nfs2 mac-addr 080020:13a09f exit-port et.3.1
```

# arp clear

## Purpose

Remove an ARP entry from the ARP table.

## Format

**arp clear** <*host*> | **all**

## Mode

Enable

## Description

The **arp clear** command lets you manually remove entries from the ARP table. The command can remove both dynamic and permanent entries.

## Parameters

| | |
|---|---|
| <*host*> | Hostname or IP address of the ARP entry to remove. |
| **all** | Remove all ARP entries, thus clearing the entire ARP table. |

## Examples

To remove the ARP entry for the host 10.8.1.2 from the ARP table:.

```
gs/r# arp clear 10.8.1.2
```

To clear the entire ARP table.

```
gs/r# arp clear all
```

If the Startup configuration file contains **arp add** commands, the Control Module re-adds the ARP entries even if you have cleared them using the **arp clear** command. To

permanently remove an ARP entry, use the **negate** command or **no** command to remove the entry. Here is an example of the **no** command:

```
gs/r# no arp add nfs2 macaddr 080020:13a09f exit-port et.3.1
```

This command removes the ARP entry for "nfs2".

# arp set interface

## Purpose

Set the lifetime of ARP entries in seconds.

## Format

**arp set interface** <*name*> | **all keep-time <*number*>**

## Mode

Configure

## Description

The **arp set interface ... keep-time** command lets you specify the lifespan (inseconds) for any or all ARP interface entries.

## Parameters

| | |
|---|---|
| **interface** <*name*> \| **all** | Name of the interface(s) for which you will define the lifespan. |
| **keep-time** <*number*> | number of seconds determining lifespan of ARP interfaces. The default value is 1200 seconds (20 minutes). |

# arp show

## Purpose

Display the ARP table.

## Format

**arp show** <*IPaddr*> | **all**

## Mode

Enable

## Description

The **arp show** command displays the entire ARP table.

## Parameters

| | |
|---|---|
| <*IPaddr*> | Shows the ARP entry for the specified IP address. |
| **all** | Shows all entries in the ARP table. |

# statistics show arp

## Purpose

Display ARP statistics.

## Format

**statistics show arp** <*Interface Name*> | **all**

## Mode

Enable

## Description

The **arp show statistics** command displays ARP statistics, such as the total number of ARP requests and replies.

## Parameters

| | |
|---|---|
| <*Interface Name*> | Displays ARP statistics for the specified interface. |
| **all** | Displays ARP statistics for all router interfaces. |

# Chapter 5

# bgp Commands

The **bgp** commands let you display and set parameters for the Border Gateway Protocol (BGP).

## Command Summary

Table 5 lists the **bgp** commands. The sections following the table describe the command syntax.

**Table 5. bgp commands**

| |
|---|
| **bgp add network** <*ipaddr-mask*> \| **all group** <*number-or-string*> |
| **bgp add peer-host** <*ipaddr*> **group** <*number-or-string*> |
| **bgp create peer-group** <*number-or-string*> |
| **bgp set DampenFlap** <*option*> |
| **bgp set default-metric** <*num*> |
| **bgp set cluster-id** <*ipaddr*> |
| **bgp set peer-group** <*number-or-string*> |
| **bgp set peer-host** <*ipaddr*> |
| **bgp set preference** <*num*> |
| **bgp show aspaths** <*aspath*> \| **all [to-terminal \| to-file]** |
| **bgp show cidr-only** <*ip-addr-mask*> \| **default \| all [to-terminal \| to-file]** |

**Table 5. bgp commands (Continued)**

| |
|---|
| **bgp show community community-id** <*number*> **autonomous-system** <*number*> **\| well-known-community [no-export \| no-advertise \| no-export-subconfed] \| reserved-community** <*number*>**] [to-terminal \| to-file]** |
| **bgp show peer-as** <*number*> **[to-terminal \| to-file]** |
| **bgp show peer-group-type external \| internal \| igp \| routing [to-terminal \| to-file]** |
| **bgp show peer-host** <*ipaddr*> **received-routes \| all-received-routes \| advertised-routes [to-terminal \| to-file]** |
| **bgp show routes** <*ip-addr-mask*> **\| default \| all [to-terminal \| to-file]** |
| **bgp show summary [to-terminal \| to-file]** |
| **bgp show sync-tree** |
| **bgp start \| stop** |
| **bgp trace** <*option*> |

# bgp add network

## Purpose

Adds a network to a BGP peer group.

## Format

**bgp add network** <*ip-addr-mask*> | **all group** <*number-or-string*>

## Mode

Configure

## Description

The **bgp add network** command lets you add a BGP peer network, thus allowing peer connections from any addresses in the specified range of network and mask pairs.

## Parameters

**network** <*ip-addr-mask*> | **all**
Specifies a network from which peer connections are allowed. Specify an IP address and Mask value. Example: 1.2.3.4/255.255.0.0 or 1.2.3.4/16. Specify **all** to add all networks.

**group** <*number-or-string*>
Specifies the group ID associated with this network range.

## Restrictions

None.

# bgp add peer-host

## Purpose

Add a BGP peer by adding a peer host.

## Format

**bgp add peer-host** <*ipaddr*> **group** <*number-or-string*>

## Mode

Configure

## Description

The **bgp add peer-host** command adds a peer-host to a BGP group.

## Parameters

**peer-host** <*ipaddr*>
Specifies the peer host's IP address.

**group** <*number-or-string*>
Specifies the group ID of the group to which the peer host belongs.

## Restrictions

None.

# bgp create peer-group

## Purpose

Create a BGP Group based on type or the autonomous system of the peers. You can create any number of groups, but each group must have a unique combination of type and peer autonomous system.

## Format

**bgp create peer-group** <*number-or-string*> **type external | internal | igp | routing**
**[autonomous-system** <*number*>**]**
**[proto any | rip | ospf | static]**
**[interface** <*interface-name-or-ipaddr*> **| all]**

## Mode

Configure

## Description

The **bgp create peer-group** command creates a BGP peer group.

## Parameters

**peer-group** <*number-or-string*>
Is a group ID, which can be a number or a character string.

**type** Specifies the type of BGP group you are adding. Specify one of the following:

**external** In the classic external BGP group, full policy checking is applied to all incoming and outgoing advertisements. The external neighbors must be directly reachable through one of the machine's local interfaces.

**internal** An internal group operating where there is no IP-level IGP, for example an SMDS network. Type internal groups expect all peers to be directly attached to a shared subnet so that, like external peers, the next hops received in BGP advertisements may be used directly for forwarding.All internal group peers should be L2 adjacent.

**igp** An internal group operating where there is no IP-level IGP, for example an SMDS network.

**routing** An internal group which uses the routes of an interior protocol to resolve forwarding addresses. Type routing groups will determine the immediate next hops for routes by using the next hop received with a route from a peer as a forwarding address, and using this to look up an immediate next hop in an IGP's routes. Such groups support distant peers, but need to be informed of the IGP whose routes they are using to determine immediate next hops. This implementation comes closest to the IBGP implementation of other router vendors.

**autonomous-system**
Specifies the autonomous system of the peer group. Specify a number from 1 – 65534.

**proto** Specifies the interior protocol to be used to resolve BGP next hops. Specify one of the following:

**any** Use any igp to resolve BGP next hops.

**rip** Use RIP to resolve BGP next hops.

**ospf** Use OSPF to resolve BGP next hops.

**static** Use static to resolve BGP next hops.

**interface** <*name-or-IPaddr*>
Interfaces whose routes are carried via the IGP for which third-party next hops may be used instead. Use only for type ROUTING group. Specify the interface or **all** for all interfaces.

## Restrictions

None.

# bgp set cluster-id

## Purpose

Specifies the route reflection cluster ID for BGP.

## Format

**bgp set cluster-id** <*ipaddr*>

## Mode

Configure

## Description

The **bgp set cluster-id** command specifies the route reflection cluster ID for BGP. The cluster ID defaults to the same as the router-id. If a router is to be a route reflector, then a single cluster ID should be selected and configured on all route reflectors in the cluster. If there is only one route reflector in the cluster, the cluster ID setting may be omitted, as the default will suffice.

## Parameters

**cluster-id** <*ipaddr*>
Is the cluster ID.

## Restrictions

The only constraints on the choice of cluster ID are (a) IDs of clusters within an AS must be unique within that AS, and (b) the cluster ID must not be 0.0.0.0. Choosing the cluster ID to be the router ID of one router in the cluster will always fulfill these criteria.

# bgp set peer-group

## Purpose

Set parameters for the specified BGP Peer Group.

## Format

**bgp set peer-group** <*number-or-string*> **[med | reflector-client | no-client-reflect | [metric-out** <*num*>**] | [set-pref** <*num*>**][local-as** <*num*>**] | ignore-first-as-hop | [generate-default enabled | disabled] | [gateway** <*ipaddr*>**] | next-hop-self | [preference** <*num*>**][preference2** <*num*>**][local-address** <*ipaddr*>**] | [hold-time** <*num*>**] | [version 2 | 3 | 4] | passive | [send-buffer** <*num*>**] | [recv-buffer** <*num*>**] | [in-delay** <*num*>**] | [out-delay** <*num*>**] | [keep all | none] | show-warnings | no-aggregator-id | keep-alives-always | v3-asloop-okay | no-v4-asloop | [as-count** <*num*>**] | log-up-down | [ttl** <*num*>**] | [optional-attributes-list** <*number-or-string*>**]]**

## Mode

Configure

## Description

The **bgp set peer-group** command sets parameters for the specified BGP group.

## Parameters

**group** <*number-or-string*>
Specifies the group.

**med**
Forces med to be used for route selection process. By default, any metric (Multi_Exit_Disc, or MED) received on a BGP connection is ignored. If it is desired to use MEDs in route selections, the **med** option must be specified in this (**create peer-group**) command. By default, MEDs are not sent on external connections. To send MEDs, use the **metric** option of the **create bgp-export-destination** statement or the **metric-out** option of the **set peer-group** or **set peer-host** commands.

**reflector-client**
The **reflector-client** option specifies that GateD will act as a route reflector for this group. All routes received from any group member will be sent to all other internal

neighbors, and all routes received from any other internal neighbors will be sent to the reflector clients. Since the route reflector forwards routes in this way, the reflector-client group need not be fully meshed. Use only for INTERNAL, ROUTING and IGP groups.

**no-client-reflect**
If the no-client-reflect option is specified, routes received from reflector clients will only be sent to internal neighbors which are not in the same group as the sending reflector client. In this case the reflector-client group should be fully meshed. In all cases, routes received from normal internal peers will be sent to all reflector clients.

Note that it is necessary to export routes from the local AS into the local AS when acting as a route reflector. The reflector-client option specifies that GateD will act as a route reflector for this group. All routes received from any group member will be sent to all other internal neighbors, and all routes received from any other internal neighbors will be sent to the reflector clients. Since the route reflector forwards routes in this way, the reflector-client group need not be fully meshed.

**metric-out** <*num*>
Specifies the primary metric used on all routes sent to the specified peer group. Specify a number from 0 - 65535.

**set-pref** <*num*>
Routes propagated by IBGP must include a Local_Pref attribute. By default, BGP sends the Local_Pref path attribute as 100, and ignores it on receipt. GateD BGP does not use Local_Pref as a route-preference decision maker unless the setpref option has been set. For Routing- or Internal-type groups, the setpref option allows GateD's global protocol preference to be exported into Local_Pref and allows Local_Pref to be used for GateD's route selection preference. Note that the setpref option is the only way for GateD to send a route with a given local_pref. The local_pref is never set directly, but rather as a function of the GateD preference and setpref metrics. Allows BGP's LOCAL_PREF attribute to be used to set the GateD preference on reception, and allows the GateD preference to set the LOCAL_PREF on transmission. The set-pref metric works as a lower limit, below which the imported LOCAL_PREF may not set the GateD preference. Use only for INTERNAL, ROUTING, and IGP groups. Specify a number from 0 - 255.

**local-as** <*num*>
Identifies the autonomous system which the router is representing to this group of peers. The default is the one configured by the **set autonomous_system** command. Specify a number from 1 - 65534.

**ignore-first-as-hop**
Some routers, known as Route Servers, are capable of propagating routes without appending their own AS to the AS path. By default, GateD will drop such routes. Specifying ignore-first-as-hop here or on either the **create peer-group** or **set peer-host** CLI commands disables this feature. This option should only be used if it is positively known that the peer is a route server and not a normal router.

**generate-default enabled | disabled**
Specifies whether the router should generate a default route when BGP receives a valid update from its peer. If this option is not specified, then the generation of default route is enabled.

**gateway** <*ipaddr*>
If a network is not shared with a peer, this option specifies a router on an attached network to be used as the next hop router for routes received from this neighbor. This field is used for EBGP Multihop. **The IP address must be a host address on a locally attached network**.

**next-hop-self**
This option causes the next hop in route advertisements set to this peer or group of peers to be set to our own router's address even if it would normally be possible to send a third-party next hop. Use of this option may cause efficient routes to be followed, but it may be needed in some cases to deal with broken bridged interconnect media (in cases where the routers on the shared medium do not really have full connectivity to each other) or broken political situations. Use only for EXTERNAL groups.

**preference** <*num*>
Specifies the preference used for routes learned from these peers. Specify a number from 0 - 255.

**preference2** <*num*>
In case of a preference tie, this option (the second preference), may be used to break the tie. The default value is 0. Specify a number from 0 - 255.

**local-address** <*ipaddr*>
Specifies the address to be used on the local end of the TCP connection with the peer or with the peer's gateway when the gateway option is used. A session with an external peer will only be opened when an interface with the appropriate local address (through which the peer or gateway address is directly reachable). In either case incoming connections will only be recognized as matching a configured peer if they are addressed to the configured local address. Use only for INTERNAL, ROUTING, and IGP groups. **It should be one of the interface addresses.**

**hold-time** <*num*>
Specifies the hold time value to use when negotiating the connection with this peer, in seconds. If BGP does not receive a keepalive, update, or notification message from a peer within the period specified in the Hold Time field of the BGP Open message, then the BGP connection will be closed. The value must be either 0 (no keepalives will be sent) or at least 6.

**version 2 | 3 | 4**
Specifies the version of the BGP protocol to use with this peer. If not specified, only the specified version will be offered. Specify 2, 3, or 4.

**passive**
Specifies that active OPENs to this peer should not be attempted. BGP would wait for

the peer to issue an OPEN. By default, all explicitly configured peers are active, they periodically send OPEN messages until the peer responds. Note that if it is applied to both sides of a peering session, it will prevent the session from ever being established.

**send-buffer** <*num*>
Controls the amount of send buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 – 65535.

**recv-buffer** <*num*>
Controls the amount of receive buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 – 65535.

**in-delay** <*num*>
Used to dampen route fluctuations. In delay specifies the amount of time in secs a route learned from a BGP peer must be stable before it is accepted into the routing database. Specify a number equal to or greater than 0. The default value is 0, meaning that this feature is disabled.

**out-delay** <*num*>
Used to dampen route fluctuations. Out delay is the amount of time in secs a route must be present in the routing table before it is exported to BGP. Specify a number equal to or greater than 0. The default value is 0, meaning that this feature is disabled.

**keep all | none**
Used to retain routes learned from a peer even if the routes' AS paths contain one of our exported AS numbers.

**show-warnings**
This option causes GateD to issue warning messages when receiving questionable BGP updates such as duplicate routes and/or deletions of non-existing routes. Normally these events are silently ignored.

**no-aggregator-id**
This option causes GateD to specify the router ID in the aggregator attribute as zero (instead of its router ID) in order to prevent different routers in an AS from creating aggregate routes with different AS paths.

**keep-alives-always**
This option causes GateD to always send keepalives, even when an update could have correctly substituted for one. This allows interoperability with routers that do not completely obey the protocol specifications on this point.

**v3-asloop-okay**
By default GateD will not advertise routes whose AS path is looped (i.e. with an AS appearing more than once in the path) to version 3 external peers. Setting this flag removes this constraint. Ignored when set on internal groups or peers.

**no-v4-asloop**
Prevents routes with looped AS paths from being advertised to version 4 external peers. This can be useful to avoid advertising such routes to peer which would incorrectly forward the routes on to version 3 neighbors.

**as-count** <*num*>
This option determines how many times the GSR will insert its own AS number when we send the AS path to an external neighbor.

Specify a number between 1 and 25. The default is 1. Higher values typically are used to bias upstream neighbors' route selection. (All else being equal, most routers will prefer to use routes with shorter AS Paths. Using **ascount**, the AS Path the GSR sends can be artificially lengthened.)

Note that ascount supersedes the **no-v4-asloop** option–regardless of whether **no-v4-asloop** is set, we will still send multiple copies of our own AS if the **as-count** option is set to something greater than one. Also, note that if the value of ascount is changed and GateD is reconfigured, routes will not be sent to reflect the new setting. If this is desired, it will be necessary to restart the peer session.

**log-up-down**
This option causes a message to be logged via the SYSLOG mechanism whenever a BGP peer enters or leaves the ESTABLISHED state.

**ttl** <*num*>
By default, BGP sets the IP TTL for local peers to ONE and the TTL for non-local peers to 255. This option is provided when attempting to communicate with improperly functioning routers that ignore packets sent with a TTL of ONE. Specify a number between 1 and 255.

**optional-attributes-list** <*number-or-string*>
Specifies the ID of the optional-attributes-list to be associated with this peer-group.

## Restrictions

None.

# bgp set DampenFlap

## Purpose

Configures parameters for Weighted Route Dampening.

## Format

**bgp set dampenflap [state enable | disable] | [suppress-above** <*num*>**] | [reuse-below** <*num*>**] | [max-flap** <*num*>**] | [unreach-decay** <*num*>**] | [reach-decay** <*num*>**] | [keep-history** <*num*>**]**

## Mode

Configure

## Description

The **bgp set dampenflap** command configures the state of Weighted Route Dampening.

## Parameters

**state enable | disable**
Causes the Route Instability History to be maintained (**enable** option) or not (**disable** option).

**suppress-above** <*num*>
Is the value of the instability metric at which route suppression will take place. A route will not be installed in the FIB or announced even if it is reachable during the period that it is suppressed. The default is 3.0.

**reuse-below** <*num*>
Is the value of the instability metric at which a suppressed route will become unsuppressed, if it is reachable but currently suppressed. The value must be less than that for the suppress-above option. The default is 2.0.

**max-flap** <*num*>
Is the upper limit of the instability metric. This value must be greater than the larger of 1 and that for suppress-above. The default is 16.0.

**unreach-decay** <*num*>
Specifies the time in seconds for the instability metric value to reach one-half of its

current value when the route is *unreachable*. This half-life value determines the rate at which the metric value is decayed. The default is 900.

**reach-decay** <*num*>
Specifies the time in seconds for the instability metric value to reach one half of its current value when the route is *reachable*. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value will make a suppressed route reusable sooner than a larger value. The default is 300.

**keep-history** <*num*>
Specifies the period in seconds over which the route flapping history is to maintained for a given route. The size of the configuration arrays is directly affected by this value. The default is 1800.

## Restrictions

None.

# bgp set default-metric

## Purpose

Set the metric used when advertising routes through BGP.

## Format

**bgp set default-metric** <*num*>

## Mode

Configure

## Description

The **bgp set default-metric** command lets you set the default metric BGP uses when it advertises routes. If this command is not specified, no metric is propagated. This metric may be overidden by a metric specified on the neighbor or group statements or in an export policy.

## Parameters

<*num*> Specifies the default cost. Specify a number from 0 - 65535.

## Restrictions

None.

# bgp set peer-host

## Purpose

Set parameters for a BGP peer host.

## Format

**bgp set peer-host** <*ipaddr*> **[group** <*number-or-string*> | **[metric-out** <*num*>**]** |
**[set-pref** <*num*>**][local-as** <*num*>**]** | **ignore-first-as-hop** |
**[generate-default enabled** | **disabled]** | **[gateway** <*ipaddr*>**]** | **next-hop-self** |
**[preference** <*num*>**][preference2** <*num*>**][local-address** <*ipaddr*>**]** |
**[hold-time** <*num*>**]** | **[version 2** | **3** | **4]** | **passive** | **[send-buffer** <*num*>**]** |
**[recv-buffer** <*num*>**]** | **[in-delay** <*num*>**]** | **[out-delay** <*num*>**]** | **[keep all** | **none]** |
**show-warnings** | **no-aggregator-id** | **keep-alives-always** | **v3-asloop-okay** |
**no-v4-asloop** | **[as-count** <*num*>**]** | **[ttl** <*num*>**]** |
**[optional-attributes-list** <*number-or-string*>**]]**

## Mode

Configure

## Description

The **bgp set peer-host** command lets you set various parameters for the specified BGP peer hosts.

## Parameters

**group** <*number-or-string*>
Specifies the group ID

**metric-out** <*num*>
Specifies the primary metric used on all routes sent to the specified peer group. The metric hierarchy is as follows, starting from the most preferred: 1)The metric specified by export policy. 2) Peer-level metricout. 3) Group-level metricout 4) Default metric. For INTERNAL, IGP, and ROUTING hosts use the **group** command to set the metric-out. Specify a number from 0 - 65535.

**set-pref** <*num*>
Allows BGP's LOCAL_PREF attribute to be used to set the GateD preference on reception, and allows the GateD preference to set the LOCAL_PREF on transmission.

The set-pref metric works as a lower limit, below which the imported LOCAL_PREF may not set the GateD preference. For INTERNAL, IGP, and ROUTING hosts, use the **group** command to set the metric-out. Specify a number from 0 - 255. **This parameter applies only to INTERNAL, IGP, and ROUTING hosts only.**

**local-as** <*num*>
Identifies the autonomous system which the router is representing to this group of peers. The default is the one configured using the **set autonomous_system** command. Specify a number from 1 - 65534.

**ignore-first-as-hop**
Some routers, known as Route Servers, are capable of propagating routes without appending their own AS to the AS path. By default, GateD will drop such routes. Specifying ignore-first-as-hop here or on either the **create peer-group** or **set peer-host** CLI commands disables this feature. This option should only be used if it is positively known that the peer is a route server and not a normal router.

**generate-default enabled | disabled**
Specifies whether the router should generate a default route when BGP receives a valid update from its peer. If this option is not specified, then the generation of default route is enabled.

**gateway** <*IPaddr*>
if a network is not shared with a peer, this option specifies a router on an attached network to be used as the next hop router for routes received from this neighbor. This is used for **EBGP multihop**. **The IP address must be a host address on a locally attached network.**

**next-hop-self**
This option causes the next hop in route advertisements set to this peer or group of peers to be set to our own router's address, even if it would normally be possible to send a third-party next hop. Use of this option may cause inefficient routes to be followed, but it may be needed in some cases to deal with broken bridged interconnect media (in cases where the routers in the shared medium do not really have full connectivity to each other) or broken political situations. **Use only for external peer hosts.**

**preference** <*num*>
Specifies the preference used for routes learned from these peers. This can differ from the default BGP preference set in the bgp set preference statement, so that GateD can prefer routes from one peer, or group of peer, over others. This preference may be explicitly overridden by import policy. Specify a number from 0 - 255.

**preference2** <*num*>
In case of preference tie, this option (the second preference), may be used to break the tie. The default value is 0. Specify a number from 0 - 255.

**local-address** <*IPaddr*>
Specifies the address to be used on the local end of the TCP connection with the peer or with the peer's gateway when the gateway option is used. A session with an

external peer will only be opened when an interface with the appropriate local address (through which the peer or gateway address is directly reachable). In either case incoming connections will only be recognized as matching a configured peer if they are addressed to the configured local address. For INTERNAL, IGP and ROUTING, hosts use the **group** command to set the local-address. **It should be one of the interface addresses.**

**hold-time** <*num*>
Specifies the hold time value to use when negotiating the connection with this peer, in seconds. If BGP does not receive a keepalive, update, or notification message from a peer within the period specified in the Hold Time field of the BGP Open message, then the BGP connection will be closed. The value must be either 0 (no keepalives will be sent) or at least 6.

**version 2 | 3 | 4**
Specifies the version of the BGP protocol to use with this peer. If not specified, only the specified version will be offered. Specify 2, 3, or 4.

**passive**
Specifies that active OPENs to this peer should not be attempted. BGP would wait for the peer to issue an OPEN. By default, all explicitly configured peers are active, they periodically send OPEN messages until the peer responds. Note that if it is applied to both sides of a peering session, it will prevent the session from ever being established.

**send-buffer** <*num*>
Controls the amount of send buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 - 65535.

**recv-buffer** <*num*>
Controls the amount of receive buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 – 65535.

**in-delay** <*num*>
Used to dampen route fluctuations. In delay specifies the amount of time in secs a route learned from a BGP peer must be stable before it is accepted into the routing database. The default value is 0, meaning that this feature is disabled. Specify a number equal to or greater than 0.

**out-delay** <*num*>
Used to dampen route fluctuations. Out delay is the amount of time in secs a route must be present in the routing table before it is exported to BGP. The default value is 0, meaning that this feature is disabled. Specify a number equal to or greater than 0.

**keep all | none**
Used to retain routes learned from a peer even if the routes' AS paths contain one of our exported AS numbers.

**show-warnings**
This option causes GateD to issue warning messages when receiving questionable BGP updates such as duplicate routes and/or deletions of non-existing routes. Normally these events are silently ignored.

**no-aggregator-id**
This option causes GateD to specify the router ID in the aggregator attribute as zero (instead of its router ID) in order to prevent different routers in an AS from creating aggregate routes with different AS paths.

**keep-alives-always**
This option causes GateD to always send keepalives, even when an update could have correctly substituted for one. This allows interoperability with routers that do not completely obey the protocol specifications on this point.

**v3-asloop-okay**
By default GateD will not advertise routes whose AS path is looped (i.e. with an AS appearing more than once in the path) to version 3 external peers. Setting this flag removes this constraint. Ignored when set on internal groups or peers.

**no-v4-asloop**
Prevents routes with looped AS paths from being advertised to version 4 external peers. This can be useful to avoid advertising such routes to peer which would incorrectly forward the routes on to version 3 neighbors.

**as-count** <*num*>
This option determines how many times we will insert our own AS number when we send the AS path to an external neighbor. Specify a number equal to or greater than 0. The default is 1. Higher values are typically used to bias upstream neighbors' route selection. (All things being equal most routers will prefer to use routes with shorter AS Paths.

Using **ascount**, the AS Path the GSR sends can be artificially lengthened.) Note that ascount supersedes the **no-v4-asloop** option--regardless of whether **no-v4-asloop** is set, the GSR will still send multiple copies its own AS if the **as-count** option is set to something greater than one.

Also, note that if the value of **ascount** is changed and GateD is reconfigured, routes will not be sent to reflect the new setting. If this is desired, it will be necessary to restart the peer session. Use only for external peer_hosts. Specify a number from 1-25.

**log-up-down**
Causes a message to be logged via the SYSLOG mechanism whenever a BGP peer enters or leaves the ESTABLISHED state.

**ttl** <*num*>
By default, BGP sets the IP TTL for local peers to ONE and the TTL for non-local peers to 255. This option is provided when attempting to communicate with improperly functioning routers that ignore packets sent with a TTL of ONE. Specify a number from 1-255.

**optional-attributes-list** *<num-or-string>*
Specifies the ID of the optional-attributes-list to be associated with this peer-group.

## Restrictions

None.

# bgp set preference

## Purpose

Set BGP preference.

## Format

**bgp set preference** <*num*>

## Mode

Configure

## Description

The **bgp set preference** command lets you set the BGP preference for the GSR.

## Parameters

<*num*> Specifies the preference of routes learned from BGP. Specify a number from 0 - 255. The default preference is 170.

## Restrictions

None.

# bgp show aspaths

## Purpose

Displays BGP AS path information

## Format

**bgp show aspaths** <*aspath*> | **all [to-terminal | to-file]**

## Mode

Enable

## Description

The **bgp show aspaths** command displays information about a specified AS path or all AS paths. The AS path is listed along with the number of routes that use it.

## Parameters

| | |
|---|---|
| <*aspath*> | Displays information about the specified AS path. |
| **all** | Displays information about all AS paths. |
| **to-terminal** | Causes output to be displayed on the terminal. This is the default. |
| **to-file** | Causes output to be saved in the file **/gatedtrc/gated.dmp**. |

## Restrictions

None.

## Example

To display information about all AS paths:

```
gs/r# bgp show aspaths all
Hash  Ref   Path
0     5     IGP (Id 1)
2     1     (64900) 64901 64902 IGP (Id 3)
7     4     (64900) 64901 IGP (Id 2)
```

# bgp show cidr-only

## Purpose

Display routes in the BGP routing table with CIDR network masks

## Format

**bgp show cidr-only** <*ip-addr-mask*> | **default** | **all** [**to-terminal** | **to-file**]

## Mode

Enable

## Description

The **bgp show cidr-only** command displays the same type of route information as the bgp show routes command. The difference is that the **bgp show cidr-only** command limits the display to CIDR routes only.

## Parameters

| | |
|---|---|
| <*ip-addr-mask*> | Displays information about the specified CIDR route. |
| **default** | Displays information about the default route. |
| **all** | Displays information about all CIDR routes. |
| **to-terminal** | Causes output to be displayed on the terminal. This is the default. |
| **to-file** | Causes output to be saved in the file **/gatedtrc/gated.dmp**. |

## Restrictions

None.

## Example

To display information all CIDR routes in the GSR's BGP route table:

```
gs/r# bgp show cidr-only all
Proto       Route/Mask NextHop         ASPath
BGP          12.2.19/25 207.135.89.65   (64800) 64753 64752 64751 6379 3561 11277 IGP (Id 13805)
BGP         12.5.172/22 207.135.89.65   (64800) 64753 64752 64751 6379 3561 1 IGP (Id 173)
BGP         12.5.252/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 1 7018 6301 IGP (Id 926)
BGP          12.6.42/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 1 7018 11090 IGP (Id 979)
BGP         12.6.134/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 1 701 7314 10562 IGP (Id 388)
BGP         12.7.214/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 7018 4129 IGP (Id 31004)
```

# bgp show community

## Purpose

Displays routes that belong to a specified community.

## Format

**bgp show community community-id** <*number*> **autonomous-system** <*number*> **| well-known-community [no-export | no-advertise | no-export-subconfed] | reserved-community** <*number*>**] [to-terminal | to-file]**

## Mode

Enable

## Description

The **bgp show community** command displays routes that belong to a specified community in a specified autonomous system.

## Parameters

**community-id** <*number*>
Is the community identifier portion of a community split. This is combined with the autonomous-system value entered to create a value for the community attribute.

**autonomous-system** <*number*> Is an autonomous system number.

**well-known-community**
Is one of the well-known communities. Specify one of the following:

**no-export**
Is a special community that indicates the routes associated with this attribute must not be advertised outside a BGP confederation boundary. Since the GSR's implementation does not support confederations, this boundary is an AS boundary.

**no-advertise**
is a special community indicating that the routes associated with this attribute must not be advertised to other BGP peers.

**no-export-subconfed**
Is a special community indicating the routes associated with this attribute must not be advertised to external BGP peers. (This includes peers in other members' autonomous systems inside a BGP confederation.)

**reserved-community <*number*>**
This option specifies one of the reserved communities that is not well-known. A reserved community is one that is in one of the following ranges (0x0000000 - 0x0000FFFF) or (0xFFFF0000 - 0xFFFFFFFF).

**to-terminal**
Causes output to be displayed on the terminal. This is the default.

**to-file**
Causes output to be saved in the file **/gatedtrc/gated.dmp**.

## Restrictions

None.

## Example

To display routes that belong to community 160 in AS 64900:

```
gs/r# bgp show community community-id 160 autonomous-system 64900
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
*> 192.68.20/24       172.16.20.2                   64901 i
*> 192.68.222/24      172.16.20.2                   64901 64902 i
```

# bgp show peer-as

## Purpose

Displays information about TCP and BGP connections to an autonomous system.

## Format

**bgp show peer-as** <*number*> **[to-terminal | to-file]**

## Mode

Enable

## Description

The **bgp show peer-as** command displays information about routers in a specified autonomous system that are peered with the GSR.

## Parameters

**peer-as** <*number*> Is the AS number of a peer autonomous system.

**to-terminal** Causes output to be displayed on the terminal. This is the default.

**to-file** Causes output to be saved in the file **/gatedtrc/gated.dmp**.

## Restrictions

None.

## Example

To display information about TCP and BGP connections to autonomous system 64901:

```
gs/r# bgp show peer-as 64901
group type External AS 64901 local 64900 flags <>
 peer 172.16.20.2 version 4 lcladdr (null) gateway (null)
  flags 0x20
  state 0x6 <Established>
  options 0x0 <>
  metric_out -1  preference 170  preference2 0
  recv buffer size 0  send buffer size 0
  messages in 10039 (updates 5  not updates 10034) 190863 octets
  messages out 10037 (updates 1  not updates 10036) 190743 octets
```

# bgp show peer-group-type

## Purpose

Displays status information about BGP peers by group.

## Format

**bgp show peer-group-type external | internal | igp | routing [to-terminal | to-file]**

## Mode

Enable

## Description

The **bgp show peer-group-type** command displays status information about BGP peers according to their group.

## Parameters

| | |
|---|---|
| **external** | Displays status information about external peers. |
| **internal** | Displays status information about internal peers. |
| **igp** | Displays status information about igp peers. |
| **routing** | Displays status information about routing peers. |
| **to-terminal** | Causes output to be displayed on the terminal. This is the default. |
| **to-file** | Causes output to be saved in the file **/gatedtrc/gated.dmp**. |

## Restrictions

None.

## Example

To display status information about external peers:

```
gs/r# bgp show peer-group-type external
Group    Neighbor        V    AS MsgRcvd MsgSent State
external 172.16.20.2     4 64901   10045   10044 Established
BGP summary  1 peers in group type "external"
```

# bgp show peer-host

## Purpose

Displays status information about BGP peer hosts.

## Format

**bgp show peer-host** <*ipaddr*> **received-routes | all-received-routes | advertised-routes [to-terminal | to-file]**

## Mode

Enable

## Description

The **bgp show peer-host** command displays information related to a specified BGP peer host. Three types of information can be displayed: routes received and accepted from a BGP peer host, all BGP routes (both accepted and rejected) from a peer host, and all routes the GSR has advertised to a peer host.

## Parameters

| | |
|---|---|
| <*ipaddr*> | Is the IP address of a BGP peer host |
| **received-routes** | Displays all valid BGP routes received and accepted from the specified peer host. |
| **all-received-routes** | Displays all BGP routes (both accepted and rejected) from the specified peer host. |
| **advertised-routes** | Displays all routes the **GSR** has advertised to the specified peer host. |
| **to-terminal** | Causes output to be displayed on the terminal. This is the default. |
| **to-file** | Causes output to be saved in the file **/gatedtrc/gated.dmp**. |

## Restrictions

None.

## Examples

To display all valid BGP routes received and accepted from peer host 172.16.20.2:

```
gs/r# bgp show peer-host 172.16.20.2 received-routes
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
*> 172.16.70/24       172.16.20.2                   64901 i
*> 172.16.220/24      172.16.20.2                   64901 i
*> 192.68.20/24       172.16.20.2                   64901 i
*> 192.68.222/24      172.16.20.2                   64901 64902 i
```

To display all BGP routes (both accepted and rejected) from peer host 172.16.20.2:

```
gs/r# bgp show peer-host 172.16.20.2 all-received-routes
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
   172.16.20/24       172.16.20.2                   64901 i
*> 172.16.70/24       172.16.20.2                   64901 i
*> 172.16.220/24      172.16.20.2                   64901 i
*> 192.68.20/24       172.16.20.2                   64901 i
*> 192.68.222/24      172.16.20.2                   64901 64902 i
```

Displays all routes the GSR has advertised to peer host 172.16.20.2:

```
gs/r# bgp show peer-host 172.16.20.2 advertised-routes
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
*> 172.16.20/24       172.16.20.1                   i
*> 192.68.11/24       192.68.11.1                   i
```

# bgp show routes

## Purpose

Displays entries in the BGP routing table.

## Format

**bgp show routes** <*ip-addr-mask*> | **default** | **all [to-terminal** | **to-file]**

## Mode

Enable

## Description

The **bgp show routes** command displays the IP address/netmask, next hop, and AS path for each BGP route.

## Parameters

| | |
|---|---|
| <*ip-addr-mask*> | Displays information about the specified route. |
| **default** | Displays information about the default route. |
| **all** | Displays information about all routes. |
| **to-terminal** | Causes output to be displayed on the terminal. This is the default. |
| **to-file** | Causes output to be saved in the file **/gatedtrc/gated.dmp**. |

## Restrictions

None.

## Example

To display the BGP routing table:

```
gs/r# bgp show routes all
Proto       Route/Mask NextHop         ASPath
BGP       172.16.70/24 172.16.20.2     (64900) 64901 IGP (Id 2)
BGP      172.16.220/24 172.16.20.2     (64900) 64901 IGP (Id 2)
BGP       192.68.20/24 172.16.20.2     (64900) 64901 IGP (Id 2)
BGP      192.68.222/24 172.16.20.2     (64900) 64901 64902 IGP (Id 3)
```

# bgp show summary

## Purpose

Displays the status of all BGP connections.

## Format

**bgp show summary [to-terminal | to-file]**

## Mode

Enable

## Description

The **bgp show summary** command displays the status of all BGP peers of the GSR.

## Parameters

**to-terminal** Causes output to be displayed on the terminal. This is the default.

**to-file** Causes output to be saved in the file **/gatedtrc/gated.dmp**.

## Restrictions

None.

## Example

To display the status of all BGP connections:

```
gs/r# bgp show summary
Neighbor        V    AS MsgRcvd MsgSent      Up/Down State
172.16.20.2     4 64901   10033   10031    6d23h8m1s Established
BGP summary  1 groups  1 peers
```

# bgp show sync-tree

## Purpose

Displays the BGP synchronization tree.

## Format

**bgp show sync-tree**

## Mode

Enable

## Description

The **bgp show sync-tree** command displays the BGP synchronization tree. The synchronization tree is used by IBGP peers to resolve the next hop (forwarding address). It gives information about routes that are orphaned because the next hop could not be resolved.

## Parameters

None.

## Restrictions

None.

## Examples

The following example shows the next hops for some of the routes that are not resolved (by showing orphaned routes):

```
gs/r# bgp show sync tree
Task BGP_Sync_64805:
        IGP Protocol: Any       BGP Group: group type Routing AS 64805

        Sync Tree (* == active  + == active with alternate  - ==
inactive with alternate:
        Orphaned routes
                Forwarding address 172.23.1.18
                        3/255 peer 172.23.1.26 preference 170
                        128.36/255.255 peer 172.23.1.26 preference 170
                        128.152/255.255 peer 172.23.1.26 preference 170
                        129.200/255.255 peer 172.23.1.26 preference 170
                        129.253/255.255 peer 172.23.1.26 preference 170
                        130.44/255.255 peer 172.23.1.26 preference 170
                        130.50/255.255 peer 172.23.1.26 preference 170
                        130.132/255.255 peer 172.23.1.26 preference 170
                        134.54/255.255 peer 172.23.1.26 preference 170
                        134.120/255.255 peer 172.23.1.26 preference 170
                        134.173/255.255 peer 172.23.1.26 preference 170
                        134.217/255.255 peer 172.23.1.26 preference 170
                        134.244/255.255 peer 172.23.1.26 preference 170
                        136.1/255.255 peer 172.23.1.26 preference 170
                        137.49/255.255 peer 172.23.1.26 preference 170
                        137.159/255.255 peer 172.23.1.26 preference 170
                        138.239/255.255 peer 172.23.1.26 preference 170
```

The following example shows the next hop for all the routes that are resolved.:

```
gs/r# bgp show sync-tree
Task BGP_Sync_64805:
        IGP Protocol: Any       BGP Group: group type Routing AS 64805

        Sync Tree (* == active  + == active with alternate  - ==
inactive with alternate:
        Node 3/8388608 route 3/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 4/8388608 route 4/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 6/8388608 route 6/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 9.2/32768 route 9.2/255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 9.20/16384 route 9.20/255.255.128 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 10.12.1/2 route 10.12.1/255.255.255.252 metric 0 interface
        Node 10.12.1.4/2 route 10.12.1.4/255.255.255.252 metric 2 next hop 172.23.1.22
        Node 10.200.12/128 route 10.200.12/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 10.203.12/128 route 10.203.12/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 10.204.12/128 route 10.204.12/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12/8388608 route 12/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.2.19/64 route 12.2.19/255.255.255.128 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.2.97/128 route 12.2.97/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.3.123/128 route 12.3.123/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.4.5/128 route 12.4.5/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.4.164/128 route 12.4.164/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.5.164/128 route 12.5.164/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.5.172/512 route 12.5.172/255.255.252 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.5.252/256 route 12.5.252/255.255.254 metric -1 next hops 172.23.1.6 172.23.1.22
```

# bgp start|stop

## Purpose

Start or stop Border Gateway Protocol (BGP).

## Format

**bgp start** | **stop**

## Mode

Configure

## Description

The **bgp** start command starts BGP on the GSR.

## Parameters

**start** Starts BGP.

**stop** Stops BGP.

## Restrictions

None.

# bgp trace

## Purpose

Set BGP trace options.

## Format

**bgp trace [packets | open | update | keep-alive [detail | send | receive | [group** <*number*> **[peer-host** <*ipaddr*>**]]] [aspath] [local-options**
**all | general | state | normal | policy | task | timer | route]**

## Mode

Configure

## Description

The **bgp trace** command lets you set BGP trace options for the GSR.

## Parameters

| | |
|---|---|
| **packets** | Traces all BGP packets. |
| **open** | Traces BGP OPEN packets, which are used to establish a peer relationship. |
| **update** | Traces BGP update packets, which are used to pass network reachability information. |
| **keep-alive** | Traces BGP KEEPALIVE packets, which are used to verify reachability. |
| **detail** | Shows detailed information about the specified packets. |
| **send** | Shows the specified packets sent by the router. |
| **receive** | Shows the specified packets received by the router. |
| **local-options** | Sets trace options for this protocol only. You can specify the following: |
| | **aspath** Traces aspath related events. |

| | |
|---|---|
| **all** | Traces all additions, changes, and deletions to the GateD routing table. |
| **general** | Activates normal and route tracing. |
| **state** | Traces state machine transitions in the protocol |
| **normal** | Traces normal protocol occurences. (Abnormal protocol occurences are always traced.) |
| **policy** | Traces the application of protocol and user-specified policies to routes being imported and exported |
| **task** | Traces system interface and processing associated with this protocol or peer |
| **timer** | Traces timer usage by this with this protocol or peer |
| **route** | Traces routing table changes for routes installed by this protocol or peer |
| **group** | Is the group ID of the group for which tracing needs to be enabled. |
| **peer-host** | peer-host ip address for which tracing needs to be enabled.The peer-host has to be qualified by the group to which it belongs |

If neither the group nor peer-host is specified then tracing is enabled for all groups and peers. If the group is specified and the peer-host is not specified then the tracing is enabled for that group. If both the peer-host and group are specified than the tracing is enabled for that peer-host in the specified group

## Restrictions

None.

# Chapter 6

# cli Commands

The **cli** commands allows you to change the behavior of the CLI in terms of command completion and command history recall.

## Command Summary

Table 6 lists the **cli** commands. The sections following the table describe the command syntax.

**Table 6. cli commands**

| |
|---|
| **cli set command completion on** \| **off** |
| **cli set history size** <*num*> \| **default** \| **maxsize** |
| **cli set terminal rows** <*num*> **columns** <*num*> |
| **cli show history** |
| **cli show terminal** |

# cli set command completion

## Purpose

Turn on or off command completion support.

## Format

**cli set command completion on | off**

## Mode

User and Configure

## Description

The **cli set command completion** command lets you enable or disable command completion support. This command works in both User and Configure mode. When executed in Configure mode, it turns on or off command completion support for the entire system. When executed in User mode, the command effects only the current login session of the user issuing that command.

## Parameters

**on** Turn on command completion

**off** Turn off command completion

## Restrictions

None

# cli set history

## Purpose

Modify command history recall characteristics.

## Format

**cli set history size** <*num*> | **default** | **maxsize**

## Mode

User and Configure

## Description

The **cli set history** command lets you to set the size of the command history buffer. Each command stored in this buffer can be recalled without having the user type in the same, complete command again. By setting the size of this history buffer, one tells the router how many of the most recently executed commands should be stored. When the buffer is full, the oldest command is pushed out to make space for the newest command. The **cli set history** command works in both user and Configure mode. When executed in Configure mode, it sets the history size of the entire system. When executed in user mode, the command effects only the current login session of the user issuing that command.

## Parameters

**size** A number specifying how many of the most recently executed commands should be kept. To disable history support, specify a size of 0. The **size** option can also take the following two keywords:

**default** Sets the history size to the system default

**maxsize** Sets the history size to the system maximum

## Restrictions

None

## Examples

To set the history buffer size to 100 commands:

```
gs/r# system set history size 100
```

# cli set terminal

## Purpose

Modify current session's terminal settings.

## Format

**cli set terminal [columns** <*num*>**] [rows** <*num*>**]**

## Mode

User

## Description

The **cli set terminal** command lets you modify the terminal screen size of the current session. By telling the system the number of rows available on your terminal, the system will automatically pause when screen output fills the entire screen.

## Parameters

**columns** Number of columns for your terminal. Minimum acceptable value is 20.

**rows** Number of rows for your terminal. The default row size is 25. To prevent output from pausing after one screen full, set the value to 0.

## Restrictions

None

## Examples

To set the number of rows to 50 lines:

```
gs/r# system set terminal rows 50
```

# cli show history

## Purpose

Display the command history from the current CLI session.

## Format

**cli show history**

## Mode

User

## Description

The **cli show history** command shows the commands you have issued during the current CLI session. A number is associated with each command. A command's number is useful for re-entering, modifying, or negating the command.

**Note:** You also can perform a command history recall by entering **!*** at any command prompt.

## Parameters

None.

## Restrictions

None

# cli show terminal

## Purpose

Display information about the current terminal settings.

## Format

**cli show terminal**

## Mode

User

## Description

The **cli show terminal** command shows information about the terminal settings. The terminal settings affect the display characteristics of your CLI session.

## Parameters

None.

## Restrictions

None.

# Chapter 7

# configure Command

The **configure** command places the CLI session in Configure mode. Configure mode allows you to set and change GSR parameters.

## Purpose

Enter the CLI's Configure mode.

## Format

configure

## Mode

Enable

## Description

Enters Configure mode. To exit Configure mode, use the **exit** command.

## Parameters

None.

## Restrictions

To enter Configure mode, you must already be in Enable mode.

# Chapter 8

# copy Command

The **copy** command lets you copy a file.

## Purpose

Copy configuration information or files.

## Format

**copy active | scratchpad | tftp-server | rcp-server | startup |** <*filename*> | <*url*> **to backup-CM | active | scratchpad | tftp-server | rcp-server | startup |** <*filename*> | <*url*>

## Mode

Enable

## Description

The **copy** command is primarily for transferring configuration information. You can copy configuration information between the GSR and external hosts using protocols such as TFTP or RCP. Within the GSR, you can copy configuration information between the GSR file system, the scratchpad (configuration database), the active (running) configuration or the Startup configuration. You also can use the **copy** command to make backup copies of a configuration file.

If the GSR has two Control Modules, you can copy the startup configuration of the primary Control Module to the secondary Control Module.

## Parameters

**active** Specifies information from the active configuration database (the running system configuration).

**scratchpad** Specifies configuration changes from the scratchpad.

**tftp-server** Downloads or uploads a file on a TFTP server.

**rcp-server** Downloads or uploads a file on an RCP server.

**startup** Copies the Startup configuration information stored in the Control Module's NVRAM.

<*filename*> Specifies the name of a file on the GSR's local file system (NVRAM or PCMCIA card).

<*url*> Specifies a URL. You can specify one of the following types of URLs:

**tftp** For example, **tftp://**<*hostname*>**/**<*path*>

**rcp** For example, **rcp://**<*username*>**@**<*hostname*>**/**<*path*>

**backup-CM** Specifies that the startup configuration be copied to the secondary Control Module. You can specify the **backup-CM** parameter only as the destination and only with **startup** as the source. When **startup** is the destination, information is copied to the secondary Control Module as well.

## Restrictions

The GSR does not allow some combinations of source and destination pair. Typically, you cannot have the same location for both source and destination; for example, you cannot copy from one TFTP server directly to another TFTP server or copy from scratchpad to scratchpad.

In addition, you cannot copy directly into the active configuration from anywhere except the scratchpad. All changes to the running system must come through the scratchpad.

## Examples

To copy configuration information from the scratchpad to the active database, enter the following command. This command activates all the uncommitted changes, thus immediately placing the changes into effect.

```
gs/r# copy scratchpad to active
```

To copy the file `config.john` to `config.debi`:.

```
gs/r# copy config.john to config.debi
```

To copy the Startup configuration to a TFTP server for backup purposes, enter the following command. The CLI prompts for the TFTP server's IP address or hostname and the filename:

```
gs/r# copy startup to tftp-server
```

To copy a previously saved configuration from a TFTP server to the Startup configuration, enter the following command. Note the use of an URL to specify the TFTP server and the filename.

```
gs/r# copy tftp://10.1.2.3/backup/config.org to startup
```

To copy the active configuration to a remote server using RCP, enter the following command. Notice that in this example a URL specifies the RCP user name, server, and filename.

```
gs/r# copy active to rcp://john@server1/config/config.dec25
```

To copy the startup configuration of the primary Control Module to the secondary Control Module:

```
gs/r# copy startup to backup-CM
```

# Chapter 9

# dvmrp Commands

The dvmrp commands let you configure and display information about Distance Vector Multicast Routing Protocol (DVMRP) interfaces.

## Command Summary

Table 7 lists the dvmrp commands. The sections following the table describe the command syntax.

**Table 7. dvmrp commands**

| |
|---|
| **dvmrp accept** \| **noaccept route** <*IPaddr/mask*> **[exact] [interface** <*IPaddr*> **[router** <*IPaddr*>**]]** |
| **dvmrp advertise** \| **noadvertise route** <*IPaddr/mask*> **[exact] [interface** <*IPaddr*>**]** |
| **dvmrp create tunnel** <*name*> **local** <*IPaddr*> **remote** <*IPaddr*> |
| **dvmrp enable no-pruning** |
| **dvmrp enable interface** <*IPaddr*> \| <*interface-name*> \| <*tunnel-name*> |
| **dvmrp set interface** <*IPaddr*> \| <*hostname*> **[metric** <*num*>**] [neighbor-timeout** <*seconds*>**] [prunetime** <*seconds*>**] [rate** <*num*>**] [scope** <*IPaddr/mask*>**] [threshold** <*num*>**]** |
| **dvmrp show interface [**<*IPaddr*>**]** |
| **dvmrp show routes host** <*IPaddr*> \| **interface** <*IPaddr*> \| **net** <*netaddr*> \| **router** <*IPaddr*> |
| **dvmrp show rules** |
| **dvmrp start** |

# dvmrp accept route

## Purpose

Specifies routes to be accepted from DVMRP neighbor routers.

## Format

**dvmrp accept | noaccept route** <*IPaddr/mask*> **[exact] [interface** <*IPaddr*> **[router** <*IPaddr*>**]]**

## Mode

Configure

## Description

The **dvmrp accept route** command allows you to specify particular routes that can be learned from DVMRP neighbors.

A route is always accepted from a DVMRP neighbor unless you use the **dvmrp noaccept route** to prevent it from being accepted. You can use the **dvmrp accept route** command along with the **dvmrp noaccept route** command to filter the routes accepted from DVMRP neighbor routers.

## Parameters

**accept**
Allows the specified route to be accepted from DVMRP neighbor routers.

**noaccept**
Prevents the specified route from being accepted from DVMRP neighbor routers.

**route** <*IPaddr/mask*>
Is the IP address and mask of the route prefix to be accepted.

**exact**
Causes only routes exactly matching the prefix to be accepted.

**interface** <*ipAddr*>
Is the IP address of the interface to which you are applying this filter.

**router** <*IPaddr*>
Is the IP address of a DVMRP neighbor router.

## Restrictions

None.

## Examples

To cause the GSR to accept only prefix 20.30.40.0/24, and filter out all other routes:

```
gs/r(config)# dvmrp noaccept route 0/0 interface customer1
gs/r(config)# dvmrp accept route 20.30.40.0/24 interface customer1
```

If interface customer1 breaks subnet 20.30.40.0/24 into smaller subnets, you can filter out routes from these subnets with the following commands:

```
gs/r(config)# dvmrp noaccept route 0/0 interface customer1
gs/r(config)# dvmrp accept route 20.30.40.0/24 interface customer1
exact
```

# dvmrp advertise route

## Purpose

Specifies routes to be advertised to DVMRP neighbor routers.

## Format

**dvmrp advertise | noadvertise route** <*IPaddr/mask*> **[exact] [interface** <*IPaddr*>**]**

## Mode

Configure

## Description

The **dvmrp advertise route** command allows you to specify particular routes that can be advertised to DVMRP neighbors. A route is always advertised to a DVMRP neighbor unless you use the **dvmrp noadvertise route** command to prevent it from being advertised. You can use the **dvmrp advertise route** command along with **dvmrp noadvertise route** to filter the routes advertised to DVMRP neighbor routers.

## Parameters

**advertise**
Allows the specified route to be advertised to DVMRP neighbor routers.

**noadvertise**
Prevents the specified route from being advertised to DVMRP neighbor routers.

**route** <*IPaddr/mask*>
Is the IP address and mask of the route prefix to be advertised.

**exact**
Causes only routes exactly matching the prefix to be advertised.

**interface** <*ipAddr*>
Is the IP address of the interface to which you are applying this filter.

## Restrictions

None.

## Examples

To prevent route 10.0.0.0/8 from being advertised on interface mbone (all other routes are advertised):

```
gs/r(config)# dvmrp noadvertise route 10/8 interface mbone
```

To advertise only route 20.20.20.0/24 to its neighbors on interface mbone:

```
gs/r(config)# dvmrp noadvertise route 0/0 interface mbone
gs/r(config)# dvmrp advertise route 20.20.20.0/24 interface mbone
```

# dvmrp create tunnel

## Purpose

Creates a DVMRP tunnel.

## Format

**dvmrp create tunnel** <*name*> **local** <*ipAddr*> **remote** <*ipAddr*>

## Mode

Configure

## Description

The **dvmrp create tunnel** command creates a DVMRP tunnel for sending multicast traffic between two end points.

## Parameters

| | |
|---|---|
| <*name*> | Name of this DVMRP tunnel. |
| **local** <*ipAddr*> | IP address of the local end point of this tunnel.<br>**Note:** The local IP address must already be configured on the GSR. |
| **remote** <*ipAddr*> | IP address of the remote end point of this tunnel. |

## Restrictions

- Tunnels use unicast routing principles. Make sure a route exists between the tunnel source and destination (***local*** <*ipAddr*> and ***remote*** <*ipAddr*>) you specify.
- An IP interface has to exist before a tunnel can be created from it.

  **Note:** A good way to confirm that a tunnel exists is to ping the other end of the tunnel.
- Tunnels cannot be created between two endpoints (that is, on the same subnet).
- A maximum of eight tunnels are allowed.

- 

## Example

To create a DVMRP tunnel called *tun12* between 10.3.4.15 (the local end of the tunnel) and 10.5.3.78 (the remote end of the tunnel):

```
gs/r(config)# dvmrp create tunnel tun12 local 10.3.4.15 remote 10.5.3.78
```

# dvmrp enable no-pruning

## Purpose

Disables DVMRP pruning.

**Note:** Pruning is enabled by default. The current DVMRP specification requires pruning capability. Unless you have a good reason for disabling pruning, DIGITAL recommends that you leave it enabled.

## Format

**dvmrp enable no-pruning**

## Mode

Configure

## Description

Disable DVMRP pruning.

## Parameters

None.

## Restrictions

None.

# dvmrp enable interface

## Purpose

Enables DVMRP on an interface.

## Format

**dvmrp enable interface** <*ipAddr/name*> | <*tunnel-name*>

## Mode

Configure

## Description

The **dvmrp enable interface** command enables DVMRP on the specified interface.

## Parameters

<*ipAddr/name*> | <*tunnel-name*>
IP address or tunnel name of the interface on which you are enabling DVMRP.

- If you are enabling DVMRP on an interface that does not have a tunnel, specify its name or IP address.
- If you are enabling DVMRP on an interface that has a tunnel, specify the tunnel name.

## Restrictions

**Note:** The Control Module's en0 interface is never used for multicast traffic.

DVMRP does not run on multiple IP subnets if created on an interface. Currently, the GSR automatically picks up the first subnet to run DVMRP on it. However any one particular subnet can be picked up by enabling it. But before doing that, no subnet should already be enabled on that interface. The GSR supports a maximum of 64 DVMRP and IGMP interfaces.

**Note:** The **igmp enable interface** command has a similar restriction of using only one subnet.

## Examples

To enable DVMRP on the IP interface with IP address 10.50.78.2:

```
gs/r(config)# dvmrp enable interface 10.50.78.2
```

To enable tunnel tun12:

```
gs/r(config)# dvmrp enable interface tun12
```

# dvmrp set interface

## Purpose

Configures various DVMRP parameters on an interface.

## Format

**dvmrp set interface** <*IPaddr/name*> **[metric** <*num*>**] [neighbor-timeout** <*seconds*>**] [prunetime** <*seconds*>**] [rate** <*num*>**] [scope** <*IPaddr/mask*>**] [threshold** <*num*>**]**

## Mode

Configure

## Description

The **dvmrp set interface** command sets DVMRP parameters on an IP interface.

## Parameters

<*ipAddr/name.*
IP address or name of the interface on which you are configuring DVMRP parameters.

**metric** <*num*>
The metric (cost) of this interface. Specify a number in the range 1 – 16. The default is 1. Normally you should not change this setting unless the network topology requires it.

**neighbor-timeout** <*num*>
The number of seconds after which the GSR will consider the neighbor to be down. Specify a number in the range 40 – 400. The default is 35.

**Note:** If you have some old routers, this value should be increased to accommodate them because they don't send probes or route updates at 40-second intervals.

**prunetime** <*seconds*>
The multicast prunetime of this interface. Specify a number in the range 300 – 7200. The default is 3600 seconds (one hour).

**rate** <*num*>
The multicast rate of this interface in kbps. Specify a number in the range 1 – 10000. The default is 500.

**Note:** The option applies only to tunnels.

**scope** <*IPaddr/mask*>
The multicast scope of this interface. The purpose of this option is to disallow the groups specified by a scope from being forwarded across an interface. This option therefore is a filtering mechanism. The threshold and the scope are two common mechanisms for implementing local simple filtering of a multicasting data.

Specify an IP address and network mask. Examples: 230.2.3.4/255.255.0.0 or 230.2.3.4/16.

**threshold** <*num*>
The multicast threshold of this interface. The purpose of this option is to allow forwarding of a packet on a multicast interface only if the packet's threshold is at least the configured value. The threshold and the scope are two common mechanisms for implementing local simple filtering of a multicasting data.

Specify a number in the range 1 – 255. The default is 1.

## Restrictions

None.

## Examples

To configure the interface 10.50.89.90 to have a metric of 5 and a threshold of 16:

```
gs/r(config)# dvmrp set interface 10.50.89.90 metric 5 threshold 16
```

# dvmrp show interface

## Purpose

Displays DVMRP interfaces.

## Format

dvmrp show interface [<*IPaddr*>]

## Mode

Enable

## Description

The **dvmrp show interface** command displays the state of an interface running DVMRP, along with other neighbor-related information. Neighbors are displayed with their DVMRP version and capability flags and Generation IDs; this information can help in debugging. If rules are in effect for an interface, they are indicated by `ExportPol` or the `ImportPol` flags.

## Parameters

<*IPaddr*> Displays DVMRP information for the specified interface.

## Restrictions

None.

## Examples

Here is an example of the **dvmrp show interface** command.

```
gs/r# dvmrp show interface
Address: 10.50.1.1            Subnet: 10.50.1/24        Met: 1   Thr: 1
Name   : pc                    State: Dn   Igmp  Dvmrp

Address: 207.135.89.10        Subnet: 207.135.89.0/27   Met: 1   Thr: 1
Name   : corp                  State: Up   Igmp  Dvmrp Querier  ExportPol
Peer   : 207.135.89.1        Version:   3.255         Flags:0xe  GID: 0x31a

Address: 10.55.89.101         Subnet: 10.55.89/24       Met: 1   Thr: 1
Name   : lab                   State: Up   Dvmrp
Peer   : 10.55.89.100        Version:   3.255         Flags:0xe  GID: 0x179

Address: 207.135.89.10       Remote: 207.137.137.1     Met: 1   Thr: 1  Rate: 1000
Name   : mbone                 State: Tunnel  Up   Dvmrp ExportPol
Peer   : 207.137.137.1       Version:   3.8           Flags:0xe  GID: 0x6c19d135
```

# dvmrp show routes

## Purpose

Displays DVMRP unicast routing table.

## Format

**dvmrp show routes host** <*IPaddr*> | **interface** <*IPaddr*> | **net** <*netaddr*> | **router** <*IPaddr*> **subordinates** | **permission**

## Mode

Enable

## Description

The **dvmrp show routes** command displays the contents of DVMRP unicast routing table.

DVMRP routes show the topology information for the internet multicasting sites. It is independent of IP unicast routing table or protocol. In this table, the information is presented about a address prefix (in form of network-address/network-mask length), the interface and the uplink (parent) router through which this subnet can be reached. This table also shows information about any routers/interfaces which consider this router as their uplink (that is, those routers which depend on this router if traffic were to originate from this subnet). These routers/interfaces are shown as children of the parent router.

**Note:** The **dvmrp show routes** command can search on the basis of subnet and on the basis of those routes whose parent is a particular interface and/or a particular router.

**Note:** This command only shows DVMRP routes and not information about current multicast sessions. For information about current multicast sessions, use the **multicast show mroutes** command.

## Parameters

**host** <*IPaddr*> Displays the route to the specified uplink host address.

**interface** <*IPaddr*> Displays the interface address of the specified uplink interface.

| | |
|---|---|
| **net** <*netaddr*> | Displays the route to the specified prefix (or subnets falling within the prefix). |
| **router** <*IPaddr*> | Displays the route to the specified router. |
| **subordinates** | Displays the downstream routers list. |
| **permissions** | Indicates whether a route is affected by any rules. Routes marked NoAdv are not advertised. |

## Restrictions

None.

## Examples

To display DVMRP routes offered by the next-hop router 207.137.137.1:

```
gs/r# dvmrp show routes router 207.137.137.1
DVMRP Routing Table (4232 routes    8 hold-down-routes)
Net: 128.119.3.16/29         Gateway: 207.137.137.1     Met:  9   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 128.119.3.8/29          Gateway: 207.137.137.1     Met:  9   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 209.12.162.16/28        Gateway: 207.137.137.1     Met: 26   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.197.171.112/28      Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.151.215.240/28      Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.151.215.192/28      Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.151.215.96/28       Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
```

To show non-advertised routes on interface lab:

```
gs/r# dvmrp show routes interface lab permission
DVMRP Routing Table (4232 routes    5 hold-down-routes)
Net: 100.100.100/24          Gateway: 10.55.89.100      Met:  2   Age:  25
Parent: lab                 Children: corp
                                      mbone             leaf NoAdv

Net: 20.20.20/24             Gateway: 10.55.89.100      Met:  2   Age:  25
Parent: lab                 Children: corp
                                      mbone             leaf NoAdv

Net: 10.55.89/24             Gateway: ----              Met:  1   Age:  --
Parent: lab                 Children: corp              leaf NoAdv
                                      mbone             leaf NoAdv

Total Routes Printed: 3
```

# dvmrp show rules

## Purpose

Displays the rules in effect for filtering routes from DVMRP neighbor routers.

## Format

**dvmrp show rules**

## Mode

Enable

## Description

The **dvmrp show rules** command displays the filtering rules in effect for DVMRP routes. Once you have set rules with the **dvmrp accept** and **dvmrp advertise** commands, you can display the active rules by entering the **dvmrp show rules** command.

## Parameters

None.

## Restrictions

None.

## Example

In this example, the following rules are in effect:

```
dvmrp advertise route 207.135.89.0/24 interface mbone
dvmrp noadvertise route 0/0 interface mbone
dvmrp advertise route 207.135.88.0/24 interface mbone
dvmrp noadvertise route 10/8 interface corp
```

To display information about these rules:

```
# dvmrp show rules
NoAdvertise: 10.0.0.0/8                IF: corp
Advertise  : 207.135.89.0/24           IF: mbone
Advertise  : 207.135.88.0/24           IF: mbone
NoAdvertise: default                   IF: mbone
```

These rules would affect the routing table as follows:

```
# dvmrp show route net 10/8 permissions

Net: 10.55.89/24           Gateway: ----            Met:  1   Age:  --
Parent: lab                Children: corp             leaf NoAdv
                                     mbone            leaf NoAdv
```

These rules prevent a directly connected route on this router from being visible to interface corp and mbone. The `leaf` flag indicates there is no downstream neighbor on the interface.

# dvmrp start

## Purpose

Starts DVMRP multicast routing.

## Format

**dvmrp start**

## Mode

Configure

## Description

The **dvmrp start** command starts DVMRP multicast routing on the configured multicast-enabled interfaces and tunnels.

**Note:** Because DVMRP is the only multicasting protocol on the GSR, IGMP starts and stops along with DVMRP. If you want to start IGMP on local interfaces, you still must use this command.

DVMRP is by default not running. DVMRP does not interact with any unicast protocol. However if you need to run a tunnel, make sure that the tunnel is reachable by a unicast routing mechanism.

## Parameters

None.

## Restrictions

None.

# Chapter 10

# enable Command

The **enable** command switches the CLI session from User mode to Enable mode.

## Format

enable

## Mode

User

## Description

The **enable** command switches your CLI session from User mode to Enable mode. After you issue the command, the CLI will prompt you for a password if a password is configured. If no password is configured, a warning message advising you to configure a password is displayed.

If a password is configured and you do not know your password or pressing Return does not work, see the administrator for the GSR.

To exit from the Enable mode and return to the User mode, use the **exit** command. To proceed form the Enable mode into the Configure mode, use the **configure** command.

## Parameters

None.

## Restrictions

None.

# Chapter 11

# erase Command

The **erase** command erases the contents of the scratchpad or Startup configuration files.

## Format

**erase scratchpad | startup**

## Mode

Configure

## Description

The **erase scratchpad** command erases the contents of the GSR's command scratchpad. The **erase startup** command erases the Startup configuration from the Control Module's NVRAM.

## Parameters

| | |
|---|---|
| **scratchpad** | Erases the contents of the scratchpad. The scratchpad contains configuration commands that you have issued but have not yet activated. |
| **startup** | Erases the contents of the Startup configuration. The Startup configuration is the configuration the GSR uses to configure itself when you reboot it. When you erase the Startup configuration, then reboot immediately, the GSR restarts without any configuration information. |

## Restrictions

The erase commands do not delete other types of files. To delete a file, use the **file del** command.

# Chapter 12

# exit Command

The **exit** command exits the current CLI mode to the previous mode. For example, if you are in the Enable mode, **exit** returns you to the User mode. If you are in Configure mode, **exit** returns you to Enable mode. If you are in User mode, **exit** closes your CLI session and logs you off the GSR.

## Format

exit

## Mode

All modes.

## Parameters

None.

## Restrictions

None.

# Chapter 13

# file Commands

The **file** commands enable you to display a directory of the files on a storage device, display the contents of a file on the console, and delete a file.

## Command Summary

Table 8 lists the **file** commands. The sections following the table describe the command syntax.

**Table 8. file commands**

| |
|---|
| **file delete** <*file-name*> |
| **file dir** <*device-name*> |
| **file type** <*file-name*> |

# file delete

## Purpose

Delete a file.

## Format

**file delete** <*file-name*>

## Mode

Enable

## Description

The **file delete** command deletes the specified file. The filename can include a device name. By default, if a device name is not specified, it is assumed to be the **bootflash:** device which is where all configuration files are stored.

## Parameters

<*file-name*> Name of the file to delete. The filename can include a device name using this format: <*device*>**:**<*file-name*>. By default, if a device name is not specified, it is assumed to be the **bootflash** device. The **bootflash** device is the default device for storing configuration files.

## Restrictions

None.

## Examples

To delete the file `config.old`:

```
gs/r# file delete config.old
```

# file dir

## Purpose

Display contents of a file system.

## Format

**file dir** <*device-name*>

## Mode

User.

## Description

Displays a directory of the files on the specified storage device.

## Parameters

<*device-name*> Device name. You can specify one of the following:

**bootflash:** The Control Module's NVRAM.

**slot0:** The PCMCIA flash card in slot 0 (the upper slot).

**slot1:** The PCMCIA flash card in slot 1(the lower slot).

## Restrictions

None.

## Examples

To display the contents of the **bootflash** device:

```
gs/r# file dir bootflash:
```

# file type

## Purpose

Display contents of a file.

## Format

file type <*file-name*>

## Mode

Enable.

## Description

Displays the contents of a file.

## Parameters

<*file-name*> Name of the file to display. The filename can include a device name using this format: <*device*>**:**<*file-name*>. By default, if a device name is not specified, it is assumed to be the **bootflash** device. The **bootflash** device is the default device for storing configuration files.

## Restrictions

None.

## Examples

To display the contents of the file `startup` (the startup configuration file):

```
gs/r# file type startup
```

# Chapter 14

# filters Commands

The **filters** commands let you create and apply the following types of security filters:

- **Address filters.** Address filters block traffic based on a frame's source MAC address, destination MAC address, or both. Address filters are always configured and applied on the input port.
- **Static entry filters.** Static entry filters allow or force traffic to go to a set of destination ports based on a frame's source MAC address, destination MAC address, or both. Static entry filters are always configured and applied on the input port. You can configure source static entry filters, destination static entry filters, and flow static entry filters. Source static entry filters allow or disallow frames based on their source MAC address; destination static entry filters allow or disallow frames based on their destination MAC address. Flow static entries allow or disallow traffic based on their source *and* destination MAC addresses.
- **Port-to-address locks.** Port-to-address lock filters "lock" a user to a port or set of ports, disallowing them access to other ports.
- **Secure ports.** Secure port filters shut down Layer 2 access to the GSR from a specific port or drop all Layer 2 packets received by a port. Used by themselves, secure ports secure unused GSR ports. When used in conjunction with static entry filters, secure ports drop all received or sent traffic (depending on the static entry filter) except traffic forced to or from the port by the static entry filter.

## Command Summary

Table 9 lists the filters commands. The sections following the table describe the command syntax.

**Table 9. filters commands**

| |
|---|
| **filters add address-filter name** <*name*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*><br>**vlan** <*VLAN-num*> **in-port-list** <*port-list*> |
| **filters add port-address-lock name** <*name*> **source-mac** <*MACaddr*><br>**vlan** <*VLAN-num*> **in-port-list** <*port-list*> |
| **filters add secure-port name** <*name*><br>**direction source \| destination vlan** <*VLAN-num*><br>**in-port-list** <*port-list*> |
| **filters add static-entry name** <*name*><br>**restriction allow \| disallow \| force source-mac** <*MACaddr*><br>**dest-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*><br>**out-port-list** <*port-list*> |
| **filters show address-filter [all-source \| all-destination \| all-flow]**<br>**[source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>**] [ports** <*port-list*>**]**<br>**[vlan** <*VLAN-num*>**]** |
| **filters show port-address-lock ports [ports** <*port-list*>**]**<br>**[vlan** <*VLAN-num*>**] [source-mac** <*MACaddr*>**]** |
| **filters show secure-port** |
| **filters show static-entry [all-source \| all-destination \| all-flow]**<br>**ports** <*port-list*> **vlan** <*VLAN-num*><br>**[source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>**]** |

# filters add address-filter

## Purpose

Applies an address filter.

## Format

**filters add address-filter name** <*name*> **source-mac** <*MACaddr*>
**dest-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>

## Mode

Configure

## Description

The **filters add address-filter** command blocks traffic based on a frame's source MAC address (**source-mac**), destination MAC address (**dest-mac**), or a flow (specified using both a source MAC address and a destination MAC address).

## Parameters

| | |
|---|---|
| **name** <*name*> | Specifies the name of the filter. |
| **source-mac** <*MACaddr*> | Specifies the source MAC address. Use this option for source or flow address filters. |
| **dest-mac** <*MACaddr*> | Specifies the destination MAC address. Use this option for destination or flow static entries. |
| **vlan** <*VLAN-num*> | Specifies the VLAN. |
| **in-port-list** <*port-list*> | Specifies the ports to which you want to apply the filter. |

## Restrictions

You should apply flow filters (specified using both a source MAC address and a destination MAC address) only to ports that are using flow-based bridging.

# filters add port-address-lock

## Purpose

Applies a port address lock.

## Format

**filters add port-address-lock name** <*name*> **source-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>

## Mode

Configure

## Description

The **filters add port-address-lock** command locks a user (identified by the user's MAC address) to a specific port or set of ports. The source MAC address will be allowed to reach only those stations and other ports that are connected to a port specified by **in-port-list**.

## Parameters

| | |
|---|---|
| **name** <*name*> | Specifies the name of the lock filter. |
| **source-mac** <*MACaddr*> | Specifies the source MAC address. |
| **vlan** <*VLAN-num*> | Specifies the VLAN. |
| **in-port-list** <*port-list*> | Specifies the ports to which you want to apply the lock. |

## Restrictions

None.

# filters add secure-port

## Purpose

Applies a port security filter.

## Format

**filters add secure-port name** <*name*> **direction source | destination vlan** <*VLAN-num*> **in-port-list** <*port-list*>

## Mode

Configure

## Description

The **filters add secure-port** command shuts down Layer 2 access to the GSR from the ports specified by **in-port-list**. The GSR drops all traffic received from these ports.

**Note:** You can use port-to-address lock filters to force traffic to a port secured by the **filters add secure-port** command.

## Parameters

**name** <*name*>
Specifies the name of the filter.

**direction source | destination**
Specifies whether the filter is to secure a source port or a destination port.

**vlan** <*VLAN-num*>
Specifies the VLAN.

**in-port-list** <*port-list*>
Specifies the ports to which you want to apply the filter.

## Restrictions

None.

# filters add static-entry

## Purpose

Applies a static entry.

## Format

**filters add static-entry name** <*name*>
**restriction allow | disallow | force source-mac** <*MACaddr*>
**dest-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>
**out-port-list** <*port-list*>

## Mode

Configure

## Description

The **filters add static-entry** command allows, disallows, or forces traffic to go to a set of destination ports based on a frame's source MAC address (**source-mac**), destination MAC address (**dest-mac**), or a flow (specified using both a source MAC address and a destination MAC address).

## Parameters

**name** <*name*>
Specifies the name of the static-entry filter.

**restriction allow | disallow | force**
Specifies the forwarding behavior of the static entry, which can be one of the following keywords:

**allow** Allows packets to go to the set of ports specified by out-port-list.

**disallow** Prohibits packets from going to the set of ports specified by out-port-list.

**force** Forces packets to go to the set of ports specified by out-port-list, despite any port locks in effect on the ports.

**source-mac** <*MACaddr*>
Specifies the source MAC address. Use this option for source or flow static entries.

**dest-mac** <*MACaddr*>
Specifies the destination MAC address. Use this option for destination or flow static entries.

**in-port-list** <*port-list*>
Specifies the ports to which you want to apply the static entry.

**out-port-list** <*port-list*>
Specifies the ports to which you are allowing, disallowing, or forcing packets.

## Restrictions

You should apply flow filters (specified using both a source MAC address and a destination MAC address) only to ports that are using flow-based bridging.

# filters show address-filter

## Purpose

Displays the address filters.

## Format

**filters show address-filter**
**[all-source | all-destination | all-flow]**
**[source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>**] [ports** <*port-list*>**]**
**[vlan** <*VLAN-num*>**]**

## Mode

Enable

## Description

The **filters show address-filter** command displays the address filters currently configured on the GSR.

## Parameters

**all-source | all-destination | all-flow**
Specifies the types of filters you want to display.

**source-mac** <*MACaddr*>
Restricts the display to only those address filters that have been applied to this source MAC address.

**dest-mac** <*MACaddr*>
Restricts the display to only those address filters that have been applied to this destination MAC address.

**ports** <*port-list*>
Restricts the display to only those address filters that have been applied to the specified ports.

**vlan** <*VLAN-num*>
Restricts the display to only those address filters that have been applied to the specified VLANs.

## Restrictions

None.

# filters show port-address-lock

## Purpose

Display the port address locks.

## Format

**filters show port-address-lock [ports** <*port-list*>**]**
**[vlan** <*VLAN-num*>**] [source-mac** <*MACaddr*>**]**

## Mode

Enable

## Description

The **filters show port-address-lock** command displays the port-address-lock filters currently configured on the GSR.

## Parameters

**ports** <*port-list*>
Restricts the display to only those port address locks that have been applied to the specified ports.

**vlan** <*VLAN-num*>
Restricts the display to only those port address locks that have been applied to the specified VLANs.

**source-mac** <*MACaddr*>
Restricts the display to only those port address locks that have been applied to this source MAC address.

## Restrictions

None.

# filters show secure-port

## Purpose

Display the port security filters.

## Format

filters show secure-port

## Mode

Enable

## Description

The **filters show secure-port** command displays the secure-port filters currently configured on the GSR.

## Parameters

None.

## Restrictions

None.

# filters show static-entry

## Purpose

Displays the static entry filters.

## Format

**filters show static-entry [all-source | all-destination | all-flow]**
**ports** <*port-list*> **vlan** <*VLAN-num*>
**[source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>**]**

## Mode

Configure

## Description

The **filters show static-entry** command displays the static-entry filters currently configured on the GSR.

## Parameters

**all-source | all-destination | all-flow**
Specifies the types of static entries you want to display.

**ports** <*port-list*>
Restricts the display to only those static entries that have been applied to the specified ports.

**vlan** <*VLAN-num*>
Restricts the display to only those static entries that have been applied to the specified VLANs.

**source-mac** <*MACaddr*>
Restricts the display to only those static entries that have been applied to this source MAC address.

**dest-mac** <*MACaddr*>
Restricts the display to only those static entries that have been applied to this destination MAC address.

## Restrictions

None.

# Chapter 15

# frame relay Commands

The following commands allow you to define frame relay service profiles, and specify and monitor frame relay High-Speed Serial Interface (HSSI) and standard serial ports.

## Command Summary

Table 10 lists the frame relay commands. The sections following the table describe the command syntax.

**Table 10. frame relay commands**

| |
|---|
| **frame-relay apply service** <*service name*> **ports** <*port list*> |
| **frame-relay create vc** <*port*> |
| **frame-relay define service** <*service name*> **[Bc** <*number*>**] [Be** <*number*>**] [becn-adaptive-shaping** <*number*>**] [cir** <*number*>**] [high-priority-queue-depth** <*number*>**] [low-priority-queue-depth** <*number*>**] [med-priority-queue-depth** <*number*>**] [red on \| off] [red-maxTh-high-prio-traffic** <*number*>**] [red-maxTh-low-prio-traffic** <*number*>**] [red-maxTh-med-prio-traffic** <*number*>**] [red-minTh-high-prio-traffic** <*number*>**] [red-minTh-low-prio-traffic** <*number*>**] [red-minTh-med-prio-traffic** <*number*>**] [rmon on \| off]** |
| **frame-relay set fr-encaps-bgd ports** <*port list*> |
| **frame-relay set lmi [error-threshold** <*number*>**] [full-enquiry-interval** <*number*>**] [monitored-events** <*number*>**] [polling-interval** <*number*>**] [state enable \| disable] [type ansi617d-1994 \| q933a \| rev1] port** <*port list*> |
| **frame-relay set peer-addr** <*IP address*> **ports** <*port list*> |

**Table 10. frame relay commands (Continued)**

| |
|---|
| **frame-relay show service** <*service name*> **\| all** |
| **frame-relay show stats port** <*port name*> **[last-error] [lmi] [mibII]** |
| **frame-relay show stats port** <*port name*> **summary** |

# frame-relay apply service ports

## Purpose

Apply a pre-defined service profile to a frame relay virtual circuit (VC).

## Format

**frame-relay apply service** <*service name*> **ports** <*port list*>

## Mode

Configure

## Description

Issuing the **frame-relay apply service** command allows you to apply a previously defined service profile to a given frame relay VC.

## Parameters

<*service name*> The name of the previously defined service profile you wish to apply to the given port(s) or interfaces.

<*port list*> The port(s) to which you wish to apply the pre-defined service profile. You can specify a single VC or a comma-separated list of VCs.

## Restrictions

Usage is restricted to frame relay VCs only.

## Example

To apply the service "s1" to slot 2, VC 100 on serial ports 1 and 2:

```
gs/r(config)# frame-relay apply service s1 ports se.2.1.100 se.2.2.100
```

# frame-relay create vc

## Purpose

Create frame relay virtual circuits (VCs).

## Format

**frame-relay create vc** <*port*>

## Mode

Configure

## Description

The **frame-relay create vc** command allows you to create a frame-relay virtual circuit on a slot and port location specified in the command line.

## Parameters

<*port*> The port on which you wish to create a frame relay virtual circuit.

## Restrictions

Usage is restricted to frame relay ports only.

## Example

To create a frame relay virtual circuit with a DLCI of 100 on serial port 1 of slot 3:

```
gs/r(config)# frame-relay create vc port se.3.1.100
```

# frame-relay define service

## Purpose

Configure service profiles for frame relay ports.

## Format

**frame-relay define service** <*service name*> **[bc** <*number*>**] [be** <*number*>**]**
**[becn-adaptive-shaping** <*number*>**] [cir** <*number*>**] [high-priority-queue-depth** <*number*>**] [low-priority-queue-depth** <*number*>**] [med-priority-queue-depth** <*number*>**]**
**[red on | off] [red-maxTh-high-prio-traffic** <*number*>**] [red-maxTh-low-prio-traffic** <*number*>**] [red-maxTh-med-prio-traffic** <*number*>**]**
**[red-minTh-high-prio-traffic** <*number*>**] [red-minTh-low-prio-traffic** <*number*>**]**
**[red-minTh-med-prio-traffic** <*number*>**] [rmon on | off]**

## Mode

Configure

## Description

The **frame-relay define service** command allows you to specify the following attributes for a newly created service profile:

- Number of bits per second contained in a committed burst for frame relay virtual circuits.
- Number of bits per second contained in an excessive burst for frame relay virtual circuits.
- Whether or not to simultaneously enable and specify the threshold at which adaptive shaping will activate when receiving BECN frames
- The committed information rate (in bits per second) for frame relay virtual circuits.
- The allowable queue depth for high-, low-, and medium-priority items on frame relay VCs.
- Activation or deactivation of Random Early Discard (RED) for frame relay circuits.

- The maximum and minimum threshold values for RED high-, low-, and medium-priority traffic.

  In general, DIGITAL recommends that the maximum threshold values be less than or equal to the respective high-, low-, or medium-priority queue depth. The minimum threshold values should be one-third of the respective maximum threshold.

- Activation and deactivation of RMON for frame relay VCs.

## Parameters

*<service name>*
The name you wish to assign to the newly created service profile.

**Bc** *<number>*
The number of bits per second contained in a committed burst for a frame relay virtual circuit. You can specify a number between 1 and 2,147,483,646 bits per second.

**Be** *<number>*
The number of bits per second contained in an excessive burst for a frame relay virtual circuit. You can specify a number between 1 and 2,147,483,646 bits per second.

**becn-adaptive-shaping** *<number>*
The threshold (number of frames) at which adaptive shaping will activate when receiving BECN frames. You can specify a number between 1 and 100,000 frames.

**cir** *<number>*
The committed information rate (in bits per second) for frame relay virtual circuits. You can specify a number between 1 and 2,147,483,646 bits.

**high-priority-queue-depth** *<number>*
The number of high-priority items allowed in the frame relay queue. You can specify a number between 1 and 65,535. DIGITAL recommends a value within the 5 - 100 item range. The default value is 20.

**low-priority-queue-depth** *<number>*
The number of low-priority items allowed in the frame relay queue. You can specify a number between 1 and 65,535. DIGITAL recommends a value within the 5 - 100 item range. The default value is 20.

**med-priority-queue-depth** *<number>*
The number of medium-priority items allowed in the frame relay queue. You can specify a number between 1 and 65,535. DIGITAL recommends a value within the 5 - 100 item range. The default value is 20.

**red on | off**
Specifying the **on** keyword enables RED for frame relay ports. Specifying the **off** keyword disables RED for frame relay ports.

**red-maxTh-high-prio-traffic** <*number*>
The maximum allowable threshold for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

**red-maxTh-low-prio-traffic** <*number*>
The maximum allowable threshold for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

**red-maxTh-med-prio-traffic** <*number*>
The maximum allowable threshold for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

**red-minTh-high-prio-traffic** <*number*>
The minimum allowable threshold for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

**red-minTh-low-prio-traffic** <*number*>
The minimum allowable threshold for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

**red-minTh-med-prio-traffic** <*number*>
The minimum allowable threshold for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

**rmon on | off**
Specifying the **on** keyword enables RMON for frame relay VCs. Specifying the **off** keyword disables RMON for frame relay VCs.

## Restrictions

When defining a value for **bc**, you *must* also be sure to define an appropriate value for **cir**, and vice-versa.

## Examples

Suppose you wish to specify a frame relay virtual circuit with the following attributes:

- Committed burst value of 35 million and excessive burst value of 30 million
- BECN active shaping at 65 thousand frames
- Committed information rate (CIR) of 120 million bits per second
- Leave high-, low-, and medium-priority queue depths set to factory defaults
- Random Early Discard (RED) disabled
- RMON enabled

The command line necessary to set up a service profile with the above attributes would be as follows:

```
gs/r(config)# frame-relay define service profile1 Bc 35000000 Be
   30000000 becn-adaptive-shaping 65000 cir 120000000 red off rmon on
```

# frame-relay set fr-encaps-bgd

## Purpose

Force the ingress packets to be encapsulated in bridged format.

## Format

**frame-relay set fr-encaps-bgd ports** *<port list>*

## Mode

Configure

## Description

Issuing the **frame-relay set fr-encaps-bgd** command allows you to use bridged format encapsulation on a given frame relay VC.

## Parameters

*<port list>* The port(s) to which you wish to use bridged encapsulation. You can specify a single VC or a comma-separated list of VCs.

## Restrictions

Usage is restricted to frame relay VCs only.

## Example

To force the bridged encapsulation to slot 2, VC 100 on serial ports 1 and 2:

```
gs/r(config)# frame-relay fr-encaps-bgd ports se.2.1.100 se.2.2.100
```

# frame-relay set lmi

## Purpose

Set frame relay LMI parameters.

## Format

**frame-relay set lmi [error-threshold** <*number*>**] [full-enquiry-interval** <*number*>**] [monitored-events** <*number*>**] [polling-interval** <*number*>**] [state enabled | disabled] [type ansi617d-1994 | q933a | rev1] port** <*port list*>

## Mode

Configure

## Description

The **frame-relay set lmi** command allows you to specify the following attributes:

- The number of times the router will attempt to poll an LMI interface before declaring it down. You can define a value between 1 and 10, inclusive.
- The number of status enquiries that will be sent before a full status enquiry is requested. You can define a value between 1 and 255, inclusive.
- The number of status enquiries over which various pieces of LMI information can be collected and tabulated. For example, you can tabulate the number of times an interface was declared down/lost due to a lack of proper responses to status enquiries. You can define a value between 1 and 10, inclusive.
- The number of seconds that pass between successive status enquiry messages. You can define a value between 5 and 30, inclusive.
- Whether or not LMI messages are sent. LMI messages are not sent by default.
- The LMI type for frame relay WAN ports.

## Parameters

**error-threshold** <*number*>
The number of unanswered status enquiries that the router will make before declaring an interface to be down.

**full-enquiry-interval** <*number*>
The number of status enquiries that will be sent before a full report on status is compiled and transmitted.

**monitored-events** <*number*>
The number of status enquiries over which collection and tabulation of various pieces of LMI information will take place.

**polling-interval** <*number*>
The amount of time (in seconds) that will pass before a subsequent status enquiry takes place.

**state enabled | disabled**
Enables the sending and receiving of LMI messages. If LMI messages are enabled, the operational status of each VC is determined by the LMI messages. If LMI messages are disabled, each VC is assumed to be operationally "up". LMI messages are disabled by default.

**type ansi617d-1994 | q933a | rev1**
The LMI type for frame relay WAN ports. You can only specify the **ansi617d-1994**, **q933a**, or **rev1** keywords to define as the LMI type for WAN ports.

**port** <*port list*>
The port or ports that will assume the LMI service profile behavior.

## Restrictions

None.

## Examples

To set the number of status enquiries that will be sent before compilation and transmission of a full status report for serial port 2 of slot 2 to 75 enquiries:

```
gs/r(config)# frame-relay set lmi full-enquiry-interval 75 port
se.2.2.100
```

# frame-relay set peer-addr

## Purpose

Set the peer address in case that InArp is not supported on the remote device.

## Format

**frame-relay set peer-addr** <*IP address*> **ports** <*port list*>

## Mode

Configure

## Description

Issuing the **frame-relay set peer-addr** command allows you to set the peer address if it can't be resolved by InArp.

## Parameters

<*IP address*> The IP or IPX address you wish to use.

<*port list*> The location of the port to which you wish to assign the address.

## Restrictions

Usage is restricted to frame relay VCs only.

## Example

To assign an IP address 10.1.1.1/16 to slot 2, VC 100 on serial port 1:

```
gs/r(config)# frame-relay set peer-addr ip-addr 10.1.1.1/16 ports
    se.2.1.100
```

# frame-relay show service

## Purpose

Displays frame relay service profiles.

## Format

**frame-relay show service** <*service name*> | **all**

## Mode

Enable

## Description

The **frame-relay show service** command allows you to display the available frame relay service profiles.

## Parameters

| | |
|---|---|
| <*service name*> | The name of a particular pre-defined service profile. |
| **all** | Displays all of the available frame relay service profiles. |

## Restrictions

None.

## Example

To display the available frame relay service profiles named "prof1":

```
gs/r# frame-relay show service prof1
```

# frame-relay show stats

## Purpose

Displays frame relay statistics.

## Format

**frame-relay show stats port** <*port name*> **[last-error] [lmi] [mibII]**

## Mode

Enable

## Description

The **frame-relay show stats** command allows you to display the following frame relay port statistics for the given port:

- The last reported frame relay error.
- The active frame relay LMI parameters.
- The MIBII statistics for frame relay WAN ports.

## Parameters

**port** <*port name*>
The port or ports for which you want to display statistics.

**last-error**
Specifying the **last-error** keyword allows you to display the last reported frame relay error for the given port.

**lmi**
Specifying the **lmi** keyword allows you to displays the active frame relay LMI parameters.

**mibII**
Specifying the **mibII** keyword allows you to displays the MIBII statistics for frame relay WAN ports.

## Restrictions

The **last error**, **mibii**, and **lmi** commands are for ports only (no VC designators allowed). Otherwise, the port name may have the "VC" designator.

## Examples

To display the last recorded error and MIB II statistics and for serial port 1 of slot 3:

```
gs/r# frame-relay show stats port se.3.1 last-error mibII
```

To display the VC statistics for serial port 1, slot 3, VCs 1-10:

```
gs/r# frame-relay show stats port se.3.1.1-10
```

# frame-relay show stats summary

## Purpose

Displays a summary of all VC statistics.

## Format

**frame-relay show stats summary port** <*port name*>

## Mode

Enable

## Description

The **frame-relay show stats summary** command allows you to display all of the summary information for VC statistics.

## Parameters

<*port name*> The port or ports for which you wish to display summary statistics.

## Restrictions

None.

## Example

To display summary statistics for serial port 1 of slot 4, VC 100:

```
gs/r# frame-relay show stats summary port se.4.1.100
```

# Chapter 16

# igmp Commands

The **igmp** commands let you display and set IGMP parameters.

## Command Summary

Table 11 lists the **igmp** commands. The sections following the table describe the command syntax.

**Table 11. igmp commands**

| |
|---|
| **igmp enable interface** <*ipAddr*> |
| **igmp set interface** <*ipAddr*> **[allowed-groups** <*group-list*> **\| not-allowed-groups** <*group-list*>**] [use-all-ports]** |
| **igmp set queryinterval** <*num*> |
| **igmp set responsetime** <*num*> |
| **igmp show interfaces [group** <*IPaddr*> **\| interface** <*IPaddr*>**]** |
| **igmp show memberships [group** <*ipAddr*> **\|** *port* <*num*>**]** |
| **igmp show timers** |

# igmp enable interface

## Purpose

Enable IGMP on an interface.

## Format

**igmp enable interface** <*ipAddr*>

## Mode

Configure

## Description

The **igmp enable interface** command enable IGMP on the specified interface.

## Parameters

| | |
|---|---|
| <*ipAddr*> | IP address of the interface on which you are enabling IGMP. |

## Restrictions

IGMP is not enabled on tunnels.

## Example

To enable IGMP on interface 10.50.1.2:

```
gs/r(config)# igmp enable interface 10.50.1.2
```

# igmp set interface

## Purpose

Configure IGMP interface.

## Format

**igmp set interface** <*ipAddr*>
**[allowed-groups** <*group-list*> **| not-allowed-groups** <*group-list*>**] [use-all-ports]**

## Mode

Configure

## Description

Sets the IGMP Host Membership Query time interval. The interval you set applies to all ports on the GSR. The GSR does port based optimization in a vlan, also named by some vendors as: layer 2 IGMP snooping. The GSR does this optimization, whenever its IGMP is turned on, and there are no other routers present on that vlan. If you set on an interface option 'use-all-ports' then the optimization is turned off. So please don't use this command unless there is a special reason to do so.

## Parameters

**allowed-groups** <*group-list*>
Restricts the groups to only those specified.

**not-allowed-groups** <*group-list*>
Allows any groups besides those specified.

**use-all-ports**
Disables per-port IGMP control.

## Restrictions

IGMP does not run on multiple IP subnets if created on an interface. Currently, the GSR automatically picks up the first subnet, to run IGMP on it. However any one particular

subnet can be picked up by enabling it. But before doing that, no subnet should already be enabled on that interface.

**Note:** The **dvmrp enable interface** command has a similar restriction.

Regarding the **use-all-ports** option: If the traffic is being supplied by a dvmrp tunnel, which uses CPU-based switching, then for efficiency reasons, port based optimization is not used by this traffic.

# igmp set queryinterval

## Purpose

Configure IGMP Host Membership Query interval.

## Format

**igmp set queryinterval** <*num*>

## Mode

Configure

## Description

Sets the IGMP Host Membership Query time interval. The interval you set applies to all ports on the GSR.

## Parameters

<*num*> A value from 20 – 3600 seconds. The default is 125 seconds.

## Restrictions

None.

## Example

To set the query interval to 30 seconds:

```
gs/r(config)# igmp set queryinterval 30
```

# igmp set responsetime

## Purpose

Configure IGMP Host Membership response wait time.

## Format

**igmp set responsetime** <*num*>

## Mode

Configure

## Description

Sets the wait time for IGMP Host Membership responses. The wait time you set applies to all ports on the GSR.

## Parameters

<*num*> Response wait time in seconds. Specify a number from 10 – 3599. The default is 10.

## Restrictions

None.

## Examples

To set the Host Membership response wait time to 20 seconds:

```
gs/r(config)# igmp set responsetime 20
```

# igmp show interfaces

## Purpose

Shows the interfaces running IGMP.

## Format

**igmp show interfaces [group** <*IPaddr*> | **interface** <*IPaddr*>**]**

## Mode

Enable

## Description

The **igmp show interfaces** command shows interfaces by name or by group. When you use the command is to show interfaces by group, all interfaces containing the group membership are shown.

**Note:** This command is similar to **igmp show memberships**, except whereas the **igmp show interfaces** command shows interface details, the **igmp show memberships** command shows ports.

## Parameters

**group** <*ipAddr*> Address of a multicast group.

**interface** <*ipAddr*> Address of a interface.

## Restrictions

None.

## Example

To show information about the interfaces running IGMP:

```
gs/r# igmp show interfaces

Address: 172.1.1.10 Subnet: 172.1.1/24 Met: 1 Thr: 1
Name : mls15 State: Up Querier Leaf Igmp Dvmrp

Address: 207.135.89.64 Subnet: 207.135.89.0/25 Met: 1 Thr: 1
Name : company State: Up Querier Leaf Igmp Dvmrp
Groups : 224.0.1.12
224.1.127.255
224.0.1.24
224.2.127.253
224.2.127.254

Address: 10.135.89.10 Subnet: 10.135.89.0/25 Met: 1 Thr: 1
Name : test State: Up Querier Igmp Dvmrp

Address: 207.135.122.11 Subnet: 207.135.122.8/29 Met: 1 Thr: 1
Name : mbone State: Up Igmp Dvmrp
Groups : 224.0.1.11
224.0.1.12
224.2.127.254
239.255.255.255
224.2.127.253
```

# igmp show memberships

## Purpose

Display IGMP host memberships.

## Format

**igmp show memberships [group** <*ipAddr*> | *port* <*num*>**]**

## Mode

Enable

## Description

The **igmp show memberships** command displays IGMP host members on a specific interface and/or for a particular multicast group.

## Parameters

| | |
|---|---|
| **group** <*ipAddr*> | Address of the multicast group for which to display host memberships. |
| ***port*** <*num*> | Port numbers on which the members reside. |

## Restrictions

None.

## Examples

To display host members for multicast group 225.0.1.20:

```
gs/r(config)# igmp show memberships group 225.0.1.20
```

To displays host members for multicast group 225.0.1.20 on port et.1.1:

```
gs/r(config)# igmp show memberships group 225.0.1.20 port et.1.1
```

The following is a fuller example.

```
gs/r(config)# igmp show memberships

Group : 224.0.1.11 Ports: et.1.1
Group : 224.0.1.12 Ports: et.1.1
et.5.1
Group : 224.0.1.24 Ports: et.5.1
Group : 224.1.127.255 Ports: et.5.1
Group : 224.2.127.253 Ports: et.1.1
et.5.1
Group : 224.2.127.254 Ports: et.1.1
et.5.1
Group : 239.255.255.255 Ports: et.1.1
```

# igmp show timers

## Purpose

Display IGMP timers.

## Format

**igmp show timers**

## Mode

Enable

## Description

The **igmp show timers** command displays IGMP timers.

## Parameters

None.

## Restrictions

None.

# Chapter 17

# interface Commands

The interface commands let you create IP and IPX interfaces, add network mask and broadcast address information to existing IP interfaces, and display configuration information for IP and IPX interfaces.

## Command Summary

Table 12 lists the interface commands. The sections following the table describe the command syntax.

**Table 12. interface commands**

| |
|---|
| **interface add ip** <*InterfaceName*> **address-netmask** <*ipAddr-mask*> **[broadcast** <*ipaddr*>**]** |
| **interface create ip** <*InterfaceName*> **address-mask** <*ipAddr-mask*> **[broadcast** <*ipAddr*>**]** **vlan** <*name*> **|** **port** <*port*> **mtu** <*num*> **[output-mac-encapsulation** <*MACencap*>**]** **[up | down] [mac-addr** <*MACaddr-spec*>**]** |
| **interface create ipx** <*InterfaceName*> **address** <*ipxAddr*> **vlan** <*name*> **|** **port** <*port*> **[output-mac-encapsulation** <*MACencap*>**] [up | down] [mac-addr** <*MACaddr-spec*>**]** |
| **interface show ip** <*InterfaceName*> **| all** |
| **interface show ipx** <*InterfaceName*> **| all** |

# interface add ip

## Purpose

Configure secondary addresses for an existing interface.

## Format

**interface add ip** <*InterfaceName*> **address-mask** <*ipAddr-mask*> **[broadcast** <*ipAddr*>**]**

## Mode

Configure

## Description

The **interface add ip** command configures secondary addresses for an existing IP interface.

**Note:** The interface must already exist. To create an interface, enter the **interface create ip** command.

## Parameters

<*InterfaceName*> Name of the IP interface; for example, gs/r4.

**address-netmask** IP address and netmask of this interface. You can specify the address and mask information using the traditional format (example: 10.1.2.3/255.255.0.0) or the CIDR format (example: 10.1.2.3/16). If you specify an address without mask information, the GSR uses the natural mask for the address (/8 for Class A, /16 for Class B or /24 for Class C).

**broadcast** <*ipAddr*> Broadcast address of this interface.

## Restrictions

You can use this command only on an interface that has already been created using the **interface create ip** command.

## Example

To configure a secondary address of 10.23.4.36 with a 24-bit netmask (255.255.255.0) on the IP interface gs/r4:

```
gs/r(config)# interface add ip gs/r4 address-mask 10.23.4.36/24
```

# interface create ip

## Purpose

Create an IP interface.

## Format

**interface create ip** <*InterfaceName*> **address-mask <*ipAddr-mask*> [broadcast** <*ipAddr*>**]**
**vlan** <*name*> **| port** <*port*> **mtu** <*num*>
**[output-mac-encapsulation** <*MACencap*>**] [up | down]**
**[mac-addr** <*MACaddr-spec*>**]**

## Mode

Configure

## Description

The **interface create ip** command creates and configures an IP interface. Configuration of an IP interface can include information such as the interface's name, IP address, netmask, broadcast address, and so on. You can also create an interface in a disabled (**down**) state instead of the default enabled (**up**) state.

The GSR is pre-allocated a pool of 64 MAC addresses. By default, each new IP interface is automatically configured with the lowest MAC address in the pool (the "base" MAC address). However, you can assign an interface a different MAC address by using the **mac-addr** option.

Interfaces on the GSR are logical interfaces. Therefore, you can associate an interface with a single port or with multiple ports.

- To associate an interface with a single port, use the **port** option with the **interface create** command.
- To associate an interface with multiple ports, first create an IP VLAN and add ports to it, then use the vlan option with the **interface create** command.

**Note:** You must use either the **port** option or the **vlan** option with the **interface create** command.

## Parameters

*<InterfaceName>*
Name of the IP interface; for example, gs/r4.

***address-netmask***
IP address and netmask of this interface. You can specify the address and mask information using the traditional format (example: 10.1.2.3/255.255.0.0) or the CIDR format (example: 10.1.2.3/16). If you specify an address without mask information, the GSR uses the natural mask for the address ( /8 for Class A, /16 for Class B or /24 for Class C).

**vlan** *<name>*
Name of the VLAN associated with this interface.

**port** *<port>*
Port associated with this interface.

***mtu*** *<num>*
Sets the Maximum Transmission Unit (MTU) for this interface.

**up**
Sets the state of the interface to up. (This is the default state.)

**down**
Sets the state of the interface to down.

**output-mac-encapsulation**
The output MAC encapsulation associated with this interface. You can specify one of the following:

- **ethernet_ii** (the default)
- **ethernet_snap**

**mac-addr** *<MACaddr-spec>*
Sets the MAC address for this interface. You can specify one of the following:

- A specific MAC address – specify the entire MAC address as follows: xx:xx:xx:xx:xx:xx
- An offset from the base MAC address in the pool – specify the offset. For example, to specify an offset of 10 from the base MAC address, enter "10". For example, if the base MAC address is 00:E0:63:02:00:00 and you specify an offset of 10, the GSR assigns MAC address 00:E0:63:02:00:0A to the interface.
- The base MAC address – specify the **basemac** keyword. This is the default.

## Restrictions

None.

## Examples

To create a VLAN called IP3, add ports et.3.1 through et.3.4 to the VLAN, then create an IP interface on the VLAN:

```
gs/r(config)# vlan create IP3 ip

gs/r(config)# vlan add ports et.3.1-4 to IP3

gs/r(config)# interface create ip gs/r3 address-mask 10.20.3.42/24 vlan IP3
```

To create an interface called "gs/r7" with the address 10.50.89.88 and a 16-bit subnet mask, enter the following command. The interface is associated with port et.1.3.

```
gs/r(config)# interface create ip gs/r7 address-mask 10.50.89.88/16 port
et.1.3
```

To create an interface called "gs/r1" with a broadcast address of 10.10.42.255, enter the following command. The interface is associated with the VLAN called "marketing". The interface is created in the down (disabled) state.

```
gs/r(config)# interface create ip gs/r1 address-mask
  10.10.42.17/255.255.255.0 broadcast 10.10.42.255 vlan marketing down
```

# interface create ipx

## Purpose

Create an IPX interface.

## Format

**interface create ipx** <*InterfaceName*> **address** <*ipxAddr*>
**vlan** <*name*> | **port** <*port*>
**[output-mac-encapsulation** <*MACencap*>**] [up**|**down]**
**[mac-addr** <*MACaddr-spec*>**]**

## Mode

Configure

## Description

The **interface create ipx** command creates and configures an IPX interface. Configuration of an IPX interface can include information such as the interface's name, IPX address, VLAN, port, and output MAC encapsulation. You can also create an interface in the disabled (**down**) state instead of the default enabled (**up**) state.

The GSR is pre-allocated a pool of 64 MAC addresses. By default, each new IPX interface is automatically configured with the lowest MAC address in the pool (the "base" MAC address). However, you can assign an interface a different MAC address by using the **mac-addr** option.

## Parameters

<*InterfaceName*>
Name of the IPX interface; for example, gs/r9.

**address** <*ipxAddr*>
IPX address of this interface.

**vlan** <*name*>
Name of the VLAN associated with this interface.

**port** <*port*>
Port associated with this interface.

**up**
Sets the state of the interface to up. (This is the default state.)

**down**
Sets the state of the interface to down.

**output-mac-encapsulation**
The output MAC encapsulation associated with this interface. You can specify one of the following:

- **ethernet_ii** (the default)
- **ethernet_snap**
- **ethernet_802.2_ipx**

**mac-addr** <*MACaddr-spec*>
Sets the MAC address for this interface. You can specify one of the following:

- A specific MAC address – specify the entire MAC address as follows: xx:xx:xx:xx:xx:xx
- An offset from the base MAC address in the pool – specify the offset. For example, to specify an offset of 10 from the base MAC address, enter "10". For example, if the base MAC address is 00:E0:63:02:00:00 and you specify an offset of 10, the GSR assigns MAC address 00:E0:63:02:00:0A to the interface.
- The base MAC address – specify the **basemac** keyword. This is the default.

## Restrictions

None.

## Examples

The following commands create a VLAN called IPX10, add all the ports on the line card in slot 1 to the VLAN, and create an IPX interface called "gs/r10" with the IPX address a98d7c6f, associated with VLAN IPX10.

```
gs/r(config)# vlan create IPX10 ipx

gs/r(config)# vlan add ports et.1.* to IPX10

gs/r(config)# interface create ipx gs/r10 address a98d7c6f vlan IPX10
```

The following command creates an interface called "gs/r5" with the IPX address 82af3d57 for port et.1.3. The interface is added in the down (disabled) state.

```
gs/r(config)# interface create ipx gs/r5 address 82af3d57 port et.1.3
down
```

To create an interface called "gs/r6" with the MAC address 00:01:02:03:04:05 and IPX address 82af3d58 for port et.1.4.

```
gs/r(config)# interface create ipx gs/r6 address 82af3d58 port et.1.4
  mac-addr 00:01:02:03:04:05
```

To create an interface called "gs/r7" for a VLAN called "IPX-VLAN" on port et.1.4 with the MAC address at the base of the GSR's MAC address pool:

```
gs/r(config)# interface create ipx gs/r7 address 82af3d59 vlan IPX-VLAN
  et.1.4 mac-addr basemac
```

The following command creates an interface called "gs/r7" for a VLAN called "IPX-VLAN" on port et.1.4 with a MAC address offset by 10 from the base of the GSR's MAC address pool. If the base MAC address in the GSR's MAC address pool is 00:E0:63:02:00:00, the offset of 10 gives the interface the MAC address 00:E0:63:02:00:0A.

```
gs/r(config)# interface create ipx gs/r7 address 82af3d59 vlan IPX-VLAN
  et.1.4 mac-addr 10
```

# interface show ip

## Purpose

Display configuration of an IP interface.

## Format

**interface show ip** <*InterfaceName*> | **all**

## Mode

Enable

## Description

The **interface show ip** command displays configuration information for an IP interface.

**Note:** You can display exactly the same information from within the ip facility using the **ip show interfaces** command.

## Parameters

<*InterfaceName*> | **all**
Name of the IP interface; for example, gs/r4. Specify **all** to show configuration information about all the IP interfaces on the GSR.

## Restrictions

None.

## Examples

To display configuration information for the IP interface called "gs/r7":

```
gs/r# interface show ip gs/r7
```

To display configuration information for all IP interfaces:

```
gs/r# interface show ip all
```

# interface show ipx

## Purpose

Display configuration of an IPX interface.

## Format

**interface show ipx** <*InterfaceName*> | **all**

## Mode

Enable

## Description

The **interface show ipx** command displays configuration information for an IPX interface.

**Note:** You can display exactly the same information from within the ip facility using the **ipx show interfaces** command.

## Parameters

<*InterfaceName*> | **all**
Name of the IPX interface; for example, gs/r9. Specify **all** to show configuration information about all the IPX interfaces on the GSR.

## Restrictions

None.

## Examples

To display configuration information for the IPX interface called "gs/r8":

```
gs/r# interface show ipx gs/r8
```

To display configuration information for all IPX interfaces:

```
gs/r# interface show ipx all
```

# Chapter 18

# ip Commands

The **ip** commands let you display route table entries and various IP related tables.

## Command Summary

Table 13 lists the **ip** commands. The sections following the table describe the command syntax.

**Table 13. ip commands**

| |
|---|
| **ip add route** <*ipAddr-mask*> \| **default gateway** <*hostname-or-IPaddr*> **[host]** **[interface** <*hostname-or-IPaddr*>**]** **[preference** <*num*>**]** **[retain]** **[reject]** **[no-install]** **[blackhole]** **[gate-list** <*gateway list*>**]** |
| **ip disable icmp-redirect** |
| **ip disable forwarding** |
| **ip disable dns-lookup** |
| **ip disable proxy-arp interface** <*name*> \| **all** |
| **ip enable directed-broadcast** |
| **ip helper-address interface** <*interface-name*> <*helper-address*> <*udp-port#*> |
| **ip show connections [no-lookup]** |
| **ip show helper-address** |
| **ip show interfaces** [<*interface-name*>] |
| **ip show routes [no-lookup] [show-arps] [show-multicast] [verbose]** |

# ip add route

## Purpose

Configure a static route.

## Format

**ip add route** <*ipAddr-mask*> | **default gateway** <*hostname-or-IPaddr*> **[host] [interface** <*hostname-or-IPaddr*>**] [preference** <*num*>**] [retain] [reject] [no-install] [blackhole] [gate-list** <*gateway list*>**]**

## Mode

Configure

## Description

The **ip add route** command creates a static route entry in the route table. The static route can be a default route, a route to a network, or a route to a specific host.

## Parameters

<*ipAddr-mask*> IP address and netmask of the destination. You can specify the address and mask information using the traditional format (example: 10.1.2.3/255.255.0.0) or the CIDR format (example: 10.1.2.3/16). If you specify an address without mask information, the GSR uses the natural mask for the address ( /8 for Class A, /16 for Class B or /24 for Class C).

**gateway** <*hostname-or-IPaddr*>
IP address or hostname of the next hop router for this route.

**host** Specifies that this route is a route to a host.

**interface** The next hop interface associated with this route. When this option is specified, gateways are only considered valid when they are on one of these interfaces

**preference** The preference of this static route. The preference controls how this route competes with routes from other protocols. The parameter takes a value between 0-255. The default preference is 60.

**retain** If specified, this option prevents this static route from being removed from the forwarding table when the routing service (GateD) is gracefully shutdown. Normally gated removes all routes except interface routes during a graceful shutdown. The retain option can be used to insure that some routing is available even when GateD is not running.

**reject** If specified, install this route as a reject route. Instead of forwarding a packet like a normal route, reject routes cause packets to be dropped and unreachable messages to be sent to the originator of the packet.

**no-install** If specified, the route will not be installed in the forwarding table when it is active but will be eligible for exporting to other protocols.

**blackhole** This option is the same as the reject option with the exception that unreachable messages are not sent.

**gate-list** <*gateway list*>
Allows you to specify up to four gateways for a particular destination host or network.

## Restrictions

None

## Examples

To configure the router 10.4.1.1 as the default gateway for this GSR:

```
gs/r(config)# ip add route default gateway 10.4.1.1
```

To configure the gateway 10.4.78.11 as the gateway for any packet destined for the subnet 10.4.14.0/24:

```
gs/r(config)# ip add route 10.4.14.0/24 gateway 10.4.78.11
```

To configure the gateway 10.4.78.11 as the gateway for any packet destined for the subnet 10.4.14.0/24:

```
gs/r(config)# ip add route 10.4.14.0/24 gateway 10.4.78.11
```

To configure the gateway 10.4.16.99 as the gateway to the host 10.4.15.2:

```
gs/r(config)# ip add route 10.4.15.2 host gateway 10.4.16.99
```

To configure a reject route entry for packets destined for the subnet 10.14.3.0/24:

```
gs/r(config)# ip add route 10.14.3.0/24 gateway 10.1.16.99 reject
```

# ip disable icmp-redirect

## Purpose

Configure the router to disable sending ICMP redirect packets for an interface.

## Format

**ip disable icmp-redirect interface** <*interface name*> | **all**

## Mode

Configure

## Description

The **ip disable icmp-redirect** command diables the router's ability to redirect packets on an interface. The GSR sends ICMP Redirect packets by default.

## Parameters

**interface** <*interface name*> | **all**

This is the name of the specified IP interface. For example, you can define an interface name of "gs/r4". If you specify the **all** keyword, ICMP redirection is disabled for all network interfaces.

## Restrictions

None

## Examples

To disable ICMP redirection on the "gs/r4" network interface:

```
gs/r(config)# ip disable icmp-redirect gs/r4
```

To disable ICMP redirection on all network interfaces:

```
gs/r(config)# ip disable icmp-redirect all
```

# ip disable forwarding

## Purpose

Disables IP forwarding on the GSR.

## Format

**ip disable forwarding**

## Mode

Configure

## Description

The **ip disable forwarding** command diables the router's ability to forward IP packets. No IP packets will be forwarded to any IP interface if this command is used.

## Parameters

None

## Restrictions

None

# ip disable dns-lookup

## Purpose

Disable DNS name lookup for all commands.

## Format

**ip disable dns-lookup**

## Mode

Configure

## Description

The **ip disable dns-lookup** command disables DNS name lookup for all commands. Sometimes a DNS server is too slow to respond and this can cause a command that displays information about many hosts to take a long time to finish. Disabling DNS lookup displays all host addresses as IP addresses instead of host names.

## Parameters

None

## Restrictions

None

# ip disable proxy-arp interface

## Purpose

Disables the proxy ARP feature on the GSR.

## Format

**ip disable proxy-arp interface** <*interface name*> | **all**

## Mode

Configure

## Description

By default, the GSR acts as a proxy for ARP requests with destination addresses of hosts to which the GSR can route traffic. The **ip disable proxy-arp interface** command disables the proxy ARP feature. Unless you actually require the use of proxy ARP, it is advisable to disable it on the GSR.

## Parameters

**interface** <*interface name*> | **all**

This is the name of the specified IP interface. For example, you can define an interface name of "gs/r4". If you specify the **all** keyword, the proxy ARP feature is disabled for all network interfaces.

## Restrictions

None

# ip enable directed-broadcast

## Purpose

Configure the router to forward directed broadcast packets received on an interface.

## Format

**ip enable directed-broadcast interface** <*interface name*> | **all**

## Mode

Configure

## Description

Directed broadcast packets are network or subnet broadcast packets which are sent to a router to be forwarded as broadcast packets. They can be misused to created Denial Of Service attacks. The GSR protects against this possibility by NOT forwarding directed broadcasts, by default. To enable the forwarding of directed broadcasts, use the **ip enable directed-broadcast** command.

## Parameters

**interface** <*interface name*> | **all**

This is the name of the specified IP interface. For example, you can define an interface name of "gs/r4". If you specify the **all** keyword, directed broadcast forwarding is enabled for all network interfaces.

## Restrictions

None

## Examples

To enable directed broadcast forwarding on the "gs/r4" network interface:

```
gs/r(config)# ip enable directed-broadcast interface gs/r4
```

To enable directed broadcast forwarding for all network interfaces:

```
gs/r(config)# ip enable directed-broadcast interface all
```

# ip helper-address

## Purpose

Configure the router to forward specific UDP broadcast packets across interfaces.

## Format

**ip helper-address interface** <*interface-name*> <*helper-address*> <*udp-port#*>

## Mode

Configure

## Description

The **ip helper-address** command allows the user to forward specific UDP broadcast from one interface to another. Typically, broadcast packets from one interface are not forwarded (routed) to another interface. However, some applications use UDP broadcast to detect the availability of a service. Other services, for example BOOTP/DHCP require broadcast packets to be routed so that they can provide services to clients on another subnet. An IP helper can be configured on each interface to have UDP broadcast packets forwarded to a specific host for a specific service.

The **ip helper-address** command allows the user to specify a UDP port number for which UDP broadcast packets with that destination port number will be forwarded. By default, if no UDP port number is specified, the GSR will forward UDP broadcast packets for the following six services:

- BOOTP/DHCP (port 67 and 68)
- DNS (port 37)
- NetBIOS Name Server (port 137)
- NetBIOS Datagram Server (port 138)
- TACACS Server (port 49)
- Time Service (port 37)

## Parameters

| | |
|---|---|
| <*interface-name*> | Name of the IP interface where UDP broadcast is to be forwarded to the helper address. |
| <*helper-address*> | Address of the host where UDP broadcast packets should be forwarded. |
| <*udp-port*> | Destination UDP port number of the broadcast packets to forward. If not specified, packets for the six default services will be forwarded to the helper address. |

## Restrictions

If you specify an interface name, the name must belong to an existing IP interface.

## Examples

To forward UDP broadcast packets on interface gs/r*1* to the host 10.1.4.5 for the six default UDP services:

```
gs/r(config)# ip helper-address interface gs/r1 10.1.4.5
```

To forward UDP broadcast packets on interface gs/r2 to the host 10.2.48.8 for packets with the destination port 111 (port mapper):

```
gs/r(config)# ip helper-address interface gs/r2 10.2.48.8 111
```

# ip show connections

## Purpose

Show all TCP/UDP connections and services.

## Format

**ip show connections [no-lookup]**

## Mode

Enable

## Description

The **ip show connections** command displays all existing TCP and UDP connections to the GSR as well as TCP/UDP services available on the GSR.

## Parameters

**no-lookup** By default, when displaying an IP address, this command attempts to do a reverse DNS lookup to look for the hostname associated with the IP address and display the hostname instead. If you do not want the reverse DNS lookup to occur, specify the **no-lookup** option.

## Restrictions

None.

## Example

The following example displays all established connections and services of the GSR.

```
gs/r# ip show connections
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address
(state)
tcp        0      0  *:gated-gii           *:*                    LISTEN
tcp        0      0  *:http                *:*                    LISTEN
tcp        0      0  *:telnet              *:*                    LISTEN
udp        0      0  127.0.0.1:1025        127.0.0.1:162
udp        0      0  *:snmp                 *:*
udp        0      0  *:snmp-trap            *:*
udp        0      0  *:bootp-relay          *:*
udp        0      0  *:route                *:*
udp        0      0  *:*                    *:*
```

# ip show helper-address

## Purpose

Display the configuration of IP helper addresses.

## Format

**ip show helper-address [**<*interface-name*>**]**

## Mode

Enable

## Description

The **ip show helper-address** command displays the configuration of IP helper addresses configured on the system. One can specify the optional parameter, *interface-name*, to show only the IP helper addresses configured for that interface. If the command is executed without specifying an interface name then the IP helper address configuration of all interfaces are shown.

## Parameters

<*interface-name*> Name of the IP interface to display any configured IP helper addresses.

## Restrictions

If you specify an interface name, the name must belong to an existing IP interface.

## Example

The following example shows that interface gs/r4 has one helper address configured while interface gs/r3 has one helper address configured for the port mapper service (port 111).

```
gs/r# ip show helper-address
Interface       IP address      Helper Address
---------       ----------      --------------
gs/r6            10.1.17.1       none
gs/r5            10.1.16.1       none
gs/r4            10.1.15.1       10.4.1.45
gs/r1            10.1.12.1       none
gs/r0            10.1.11.1       none
gs/r3            10.1.14.1       10.5.78.122(111)
```

# ip show interfaces

## Purpose

Display the configuration of IP interfaces.

## Format

**ip show interfaces** [<*interface-name*>]

## Mode

Enable

## Description

The **ip show interfaces** command displays the configuration of an IP interface. If you issue the command without specifying an interface name then the configuration of all IP interfaces is displayed. This command displays the same information as the **interface show ip** command.

## Parameters

<*interface-name*> Name of the IP interface; for example, gs/r4. If you do not specify an interface name, the GSR displays all the IP interfaces.

## Restrictions

If you specify an interface name, the name must belong to an existing IP interface.

## Example

To display the configuration of the IP interface "gs/r1":

```
gs/r# ip show interfaces gs/r1
gs/r1: flags=9862<BROADCAST NOTRAILERS RUNNING SIMPLEX LINK0 MULTICAST>
        VLAN: IP2
        Ports:
        inet 10.1.12.1/24 broadcast 10.1.12.255
```

# ip show routes

## Purpose

Display the IP routing table.

## Format

**ip show routes [no-lookup] [show-arps] [show-multicast] [verbose]**

## Mode

Enable

## Description

The **ip show routes** command displays the IP routing table. Different command options can be used to show different aspects of the routing table.

## Parameters

**no-lookup** By default, when displaying an IP address, this command attempts to do a reverse DNS lookup to look for the hostname associated with the IP address and display the hostname instead. If you do not want the reverse DNS lookup to occur, specify the **no-lookup** option.

**show-arps** By default, ARP entries are not shown. To show ARP entries (if any are present), specify the **show-arps** option.

**show-multicast** By default, routes to multicast destinations are not shown. To show routes to multicast destinations, specify the **show-multicast** option.

**verbose** Show the routing table in verbose mode. The additional information is useful for debugging.

## Restrictions

None.

## Example

The following example displays the contents of the routing table. It shows that some of the route entries are for locally connected interfaces ("directly connected"), while some of the other routes are learned from RIP.

```
gs/r# ip show routes
Destination                Gateway                    Owner     Netif
-----------                -------                    -----     -----
10.1.0.0/16                50.1.1.2                   RIP       to-linux2
10.2.0.0/16                50.1.1.2                   RIP       to-linux2
10.3.0.0/16                50.1.1.2                   RIP       to-linux2
10.4.0.0/16                50.1.1.2                   RIP       to-linux2
14.3.2.1                   61.1.4.32                  Static    gs/r61
21.0.0.0/8                 50.1.1.2                   RIP       to-linux2
30.1.0.0/16                directly connected         -         to-goya
50.1.0.0/16                directly connected         -         to-linux2
61.1.0.0/16                directly connected         -         gs/r61
62.1.0.0/16                50.1.1.2                   RIP       to-linux2
68.1.0.0/16                directly connected         -         gs/r68
69.1.0.0/16                50.1.1.2                   RIP       to-linux2
127.0.0.0/8                127.0.0.1                  Static    lo
127.0.0.1                  127.0.0.1                  -         lo
210.11.99.0/24             directly connected         -         gs/r41
```

# Chapter 19

# ip-router Commands

The **ip-router** commands let you configure and monitor features and functions that work across the various routing protocols.

## Command Summary

Table 14 lists the **ip-router** commands. The sections following the table describe the command syntax.

**Table 14. ip-router commands**

| |
|---|
| **ip-router authentication add key-chain** <*option-list*> |
| **ip-router authentication create key-chain** <*option-list*> |
| **ip-router global add** <*option-list*> |
| **ip-router global set** <*option-list*> |
| **ip-router global set trace-options** <*option-list*> |
| **ip-router global set trace-state on** \| **off** |
| **ip-router global use provided_config** |
| **ip-router kernel trace** <*option-list*> **detail** \| **send** \| **receive** |
| **ip-router policy add filter** <*option-list*> |
| **ip-router policy add optional-attributes-list** <*option-list*> |
| **ip-router policy aggr-gen destination** <*name*> <*option-list*> |
| **ip-router policy create aggregate-export-source** <*option-list*> |
| **ip-router policy create aggr-gen-dest** <*option-list*> |

**Table 14. ip-router commands (Continued)**

| |
|---|
| **ip-router policy create aggr-gen-source** <*option-list*> |
| **ip-router policy create aspath-export-source** <*number-or-string*> <*option-list*> |
| **ip-router policy create bgp-export-destination** <*number-or-string*> <*option-list*> |
| **ip-router policy create bgp-export-source** <*number-or-string*> <*option-list*> |
| **ip-router policy create bgp-import-source** <*number-or-string*> <*option-list*> |
| **ip-router policy create direct-export-source** <*option-list*> |
| **ip-router policy create filter** <*option-list*> |
| **ip-router policy create optional-attributes-list** <*option-list*> |
| **ip-router policy create ospf-export-destination** <*number-or-string*> <*option-list*> |
| **ip-router policy create ospf-export-source** <*number-or-string*> <*option-list*> |
| **ip-router policy create ospf-import-source** <*number-or-string*> <*option-list*> |
| **ip-router policy create rip-export-destination** <*number-or-string*> <*option-list*> |
| **ip-router policy create rip-export-source** <*number-or-string*> <*option-list*> |
| **ip-router policy create rip-import-source** <*number-or-string*> <*option-list*> |
| **ip-router policy create static-export-source** <*option-list*> |
| **ip-router policy create tag-export-source** <*number-or-string*> <*option-list*> |
| **ip-router policy export destination** <*option-list*> |
| **ip-router policy import source** <*option-list*> |
| **ip-router policy redistribute from-proto** <*protocol*> <*option-list*><br>**to-proto rip \| ospf \| bgp** |
| **ip-router show configuration-file active \| permanent** |
| **ip-router show rib [detail]** |
| **ip-router show route [ip-addr-mask \| default] [detail]** |
| **ip-router show state to-file \| to-terminal** |

# ip-router authentication add key-chain

## Purpose

Add a key to an existing key-chain.

## Format

**ip-router authentication add key-chain** <*option-list*>

## Mode

Configure

## Parameters

<*option-list*>
Specifies the options you are adding. Specify one of the following:

**key** <*string*>
Adds a new key to an existing key-chain. The key can be up to 16 characters long.

**type primary | secondary**
Specifies whether the key is a primary key or a secondary key within the key chain.

## Restrictions

None.

# ip-router authentication create key-chain

## Purpose

Create a key-chain and associate an identifier with it.

## Format

**ip-router authentication create key-chain** <*option-list*>

## Mode

Configure.

## Parameters

<*option-list*>
Specifies the options you are adding. Specify one of the following:

**key** <*string*>
Specifies a key to be included in this key chain. The key can be up to 16 characters long.

**type primary** | **secondary**
Specifies whether the key is a primary key or a secondary key within the key chain.

**id**
Specifies an integer between 1 and 255. This option is only necessary for MD5 authentication method.

## Restrictions

None.

# ip-router global add

## Purpose

Add an interface or martian. Martians are invalid addresses that are rejected by the routing software.

## Format

**ip-router global add interface** <*name-or-IPaddr*>

**ip-router global add martian** <*ipAddr/mask*> | **default [host] [allow]**

## Mode

Configure

## Parameters

**interface** <*name-or-IPaddr*>
Makes an interface known to the IP router.

**martian** <*ipAddr/mask*> | **default [host] [allow]**
Adds a martian. Specify the following options:

| | |
|---|---|
| <*ipAddr/mask*> | The IP address and netmask for the martian. |
| **default** | Adds default martian. |
| **host** | Specifies that this martian is a host address. |
| **allow** | Allows a subset of a range that was disallowed. |

## Restrictions

None.

# ip-router global set

## Purpose

Set various global parameters required by various protocols.

## Format

**ip-router global set** <*option-list*>

## Mode

Configure

## Parameters

<*option-list*>
Specify one of the following:

**autonomous-system** <*num1*> **loops** <*num2*>
The autonomous system number. <*num1*> sets the as number for the router. It is only required if the router is going to run BGP. Specify a number from 1 – 65534. <*num2*> controls the number of times the as may appear in the as-path. Default is 1. It is only required if the router is going to run protocols that support as-path, such as BGP.

**router-id** <*hostname-or-IPaddr*>
The router ID for use by BGP and OSPF. The most preferred address is any address other than 127.0.0.1 on the loopback interface. If there are no secondary addresses on the loopback interface, then the default router ID is set to the address of the first interface which is in the up state that the GSR encounters (except the interface en0, which is the Control Module's interface). The address of a non point-to-point interface is preferred over the local address of a point-to-point interface.

**interface** <*interface-name*> | **all preference** <*num*> **down-preference** <*num*> **passive autonomous-system** <*num*>

Specify the following:

<*interface-name*> | **all**
Specify an interface that was added using the *ip-router global add interface* command, or **all** for all interfaces.

**preference** <*num*>
Sets the preference for routes to this interface when it is up and functioning. Specify a number from 0 – 255. Default value is 0.

**down-preference** <*num*>
Sets the preference for routes to this interface when it is down. Specify a number from 0 – 255. Default value is 255.

**passive**
Prevents changing of route preference to this interface if it is down.

**autonomous-system** <*num*>
The AS that will be used to create as-path associated with the route created from the definition of this interface.

## Restrictions

None.

# ip-router global set trace-options

## Purpose

Set various trace options.

## Format

**ip-router global set trace-options** <*option-list*>

## Mode

Configure

## Parameters

<*option-list*>
Specifies the trace options you are setting. Specify one or more of the following:

| | |
|---|---|
| **startup** | Trace startup events. |
| **parse** | Trace lexical analyzer and parser of gate-d config files. |
| **ydebug** | Trace lexical analyzer and parser in detail. |
| **adv** | Trace allocation and freeing of policy blocks. |
| **symbols** | Trace symbols read from kernel at startup. |
| **iflist** | Trace the reading of the kernel interface list. |
| **all** | Tun on all tracing. |
| **general** | Turn on normal and route tracing |
| **state** | Trace state machine transitions in protocols. |
| **normal** | Trace normal protocol occurrences. Abnormal occurrences are always traced. |
| **policy** | Traces the application of policy to routes being exported and imported. |
| **task** | Traces system interfaces and task processing associated with this protocol or peer. |
| **timer** | Traces timer usage by this protocol or peer |
| **route** | Traces routing table changes for routes installed by this protocol or peer. |

## Restrictions

None.

# ip-router global set trace-state

## Purpose

Enable or disable tracing.

## Format

**ip-router global set trace-state on | off**

## Mode

Configure

## Parameters

**on | off** Specifies whether you are enabling or disabling tracing. Specify **on** to enable tracing or specify **off** to disable tracing. The default is **off**.

## Restrictions

None.

# ip-router global use provided_config

## Purpose

Causes the GSR to use the configuration file stored in the Control Module's NVRAM.

## Format

**ip-router global use provided_config**

## Mode

Configure

## Parameters

None.

**Note:** This command requires that you first copy the GateD configuration into the Control Module's NVRAM.

To do this, enter the following command in Enable mode:

```
gs/r# copy tftp-server to gated.conf
TFTP server [10.50.89.88]? 10.50.89.88
Source filename [tmp/gated.conf]?
########
%TFTP-I-XFERRATE  Received 5910 bytes in 0.1 seconds
```

## Restrictions

None.

# ip-router kernel trace

## Purpose

Provides trace capabilities between the Routing Information Base and the Forwarding Information Base.

## Format

**ip-router kernel trace** *<option-list>* **detail | send | receive**

## Mode

Configure

## Parameters

*<option-list>*
Specifies the kernel trace options. Specify one or more of the following:

| | |
|---|---|
| **packets** | Packets exchanged with the kernel. |
| **routes** | Routes exchanged with the kernel. |
| **redirect** | Redirect messages received from the kernel. |
| **interface** | Interface messages received from the kernel. |
| **other** | All other messages received from the kernel. |
| **remnants** | Routes read from the kernel when the GSR routing process starts. |
| **request** | The GSR routing process requests to Add/Delete/Change routes in the kernel forwarding table. |
| **info** | Informational messages received from the routing socket, such as TCP loss, routing lookup failure, and route resolution request. |

## Restrictions

None.

# ip-router policy add filter

## Purpose

Adds a route filter. Routes are specified by a set of filters that will match a certain set of routes by destination, or by destination and mask.

## Format

**ip-router policy add filter** <*number-or-string*> **network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**][host-net]**

## Mode

Configure

## Parameters

**filter** <*number-or-string*>
Specifies the identifier of the route filter.

**network** <*IP-address*>
Specifies networks that are to be filtered. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be filtered are specified, then any destination that falls in the range implied by this network-specification is matched, so the mask of the destination is ignored. If a natural network is specified, the network, any subnets, and any hosts will be matched. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

**exact**
Specifies that the mask of the routes to be filtered must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

**refines**
Specifies that the mask of the routes to be filtered must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

**between** <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as longer) than the lower limit (the first number parameter) and no more specific (i.e. as long as or shorter) than the upper limit (the second parameter).

**host-net**

This option qualifies that the specified network is a host. To match, the address must exactly match the specified and the network mask must be a host mask (i.e. all ones). This is equivalent to a network specification of host/255.255.255.255 along with the exact option.

## Restrictions

None.

# ip-router policy add optional-attributes-list

## Purpose

Expands a previously created optional-attributes-list.

## Format

**ip-router policy add optional-attributes-list** <*option-list*>

## Mode

Configure

## Parameters

<*option-list*>
Specifies the options. Specify one or more of the following:

**optional-attributes-list** <*number-or-string*>
Specifies the identifier for the optional attributes list you are expanding.

**community-id** <*number*>
Specifies a community identifier portion of a community split. This is combined with the autonomous system value entered to create a value for the community attribute.

**autonomous-system** <*number*>
Specifies the autonomous system portion of a community split. This would be combined with the community id value entered to create a value for the community attribute. Specify a number from 1 – 65534.

**no-export**
Specifies that all routes received with this attribute value ***will not*** be advertised outside a BGP confederation boundary.

**well-known-community**
Specifies one of the well-known communities.

**no-advertise**
Specifies that all routes received with this attribute value ***will not*** be advertised to other BGP peers.

**no-export-subconfed**
Specifies that all routes received with this attribute value ***will not*** be advertised to

external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).

**reserved-community** <*number*>
Specifies one of the reserved communities which is not well-known. A reserved community is one which is in one of the following ranges (0x0000000 - 0x0000FFFF) or (0xFFFF0000 - 0xFFFFFFFF).

## Restrictions

None.

# ip-router policy aggr-gen destination

## Purpose

Creates an aggregate or generate route.

## Format

**ip-router policy aggr-gen destination** <*number-or-string*> **[source** <*number-or-string*> **[filter** <*number-or-string*> **| [network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**]** **[preference** <*number*> **| restrict]]]]**

## Mode

Configure

## Parameters

**destination** <*number-or-string*>
Is the identifier of the aggregate-destination that specifies the aggregate/summarized route.

**source** <*number-or-string*>
Is the identifier of the aggregate-source that contributes to an aggregate route.

**filter** <*number-or-string*>
Specifies the filter for an aggregate/generate.

**network** <*ipAddr/mask*>
This option specifies networks which are to be aggregated. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be aggregated are specified, then any destination that falls in the range implied by this network-specification is matched, so the mask of the destination is ignored. If a natural network is specified, the network, any subnets, and any hosts will be matched. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

**exact**
This option specifies that the mask of the routes to be aggregated must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

**refines**
This option specifies that the mask of the routes to be aggregated must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

**between** <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as longer) than the lower limit (the first number parameter) and no more specific (i.e. as long as or shorter) than the upper limit (the second parameter).

**preference** <*number*>
This option specifies the preference to be assigned to the resulting aggregate route.

## Restrictions

None.

# ip-router policy create aggregate-export-source

## Purpose

Creates a source for exporting aggregate routes into other protocols.

## Format

**ip-router policy create aggregate-export-source**
<*number-or-string*> **[metric** <*number*> | **restrict]**

## Mode

Configure

## Parameters

| | |
|---|---|
| <*number-or-string*> | Specifies the identifier of the aggregate export source. |
| **metric** <*number*> | Specifies the metric to be associated with the exported routes. |
| **restrict** | Specifies that nothing is exported from the specified source. |

## Restrictions

None.

# ip-router policy create aggr-gen-dest

## Purpose

Creates an aggregate-generation destination. An aggregate-generation destination is one of the building blocks needed to create an aggregate/generate route.

## Format

**ip-router policy create aggr-gen-dest** <*number-or-string*>
**network** <*ipAddr/mask*> | **default [type aggregate | generation] [preference** <*number*>**][brief]**

## Mode

Configure

## Parameters

<*number-or-string*>
Specifies the identifier of an aggregate-generation destination.

**network** <*ipAddr/mask*> | **default**
Specifies the aggregate or generated route.

**type aggregate**
Specifies that the destination is an aggregate.

**type generation**
Specifies that the destination is a generate.

**preference** <*num*>
Specifies the preference to be assigned to the resulting aggregate route. The default preference is 130.

**brief**
Used to specify that the AS path should be truncated to the longest common AS path. The default is to build an AS patch consisting of SETs and SEQUENCES of all contributing AS paths.

## Restrictions

None.

# ip-router policy create aggr-gen-source

## Purpose

Creates a source for the routes contributing to a aggregate/generate route.

## Format

**ip-router policy create aggr-gen-source** <*number-or-string*>
**protocol all | static | direct | aggregate | rip | ospf | bgp [autonomous-system** <*number*>**][aspath-regular-expression** <*string*>**][tag** <*number*>**][preference** <*number*> **| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Specifies the identifier of an aggregate-generation source.

**protocol** <*string*>
Specifies the protocol of the contributing aggregate source. Specify one of the following:

- **all**
- **static**
- **direct**
- **aggregate**
- **rip**
- **ospf**
- **bgp**

**autonomous-system** <*number*>
Restricts selection of routes to those learned from the specified autonomous system. This selection may also be carried out by using route filters to explicitly list the set of routes to be accepted. Specify a number from 1 – 65534.

**aspath-regular-expression** <*string*>
Restricts selection of routes to those specified by the aspath.

**tag** <*number*>
Restricts selection of routes to those identified by a tag.

**preference** <*number*>
Specifies the preference to assign to the contributing routes.

**restrict**
Indicates that these routes cannot contribute to the aggregate.

## Restrictions

None.

# ip-router policy create aspath-export-source

## Purpose

Create an export source where routes to be exported are identified by the autonomous system path associated with them. This command applies only if you are using BGP.

## Format

**ip-router policy create aspath-export-source** <*number-or-string*> <*option-list*>

## Mode

Configure

## Parameters

<*number-or-string*>
Specifies a name or number for the Autonomous System path export source.

<*option-list*>
Specifies the Autonomous System path source options you are setting. Specify one of the following:

**protocol** <*name*>
Specifies the protocol by which the routes to be exported were learned. Specify one of the following:

- all
- **static**
- **direct**
- **aggregate**
- **rip**
- **ospf**
- **bgp**

**aspath-regular-expression** <*string*>
Specifies an aspath regular expression which should be satisfied for the route to be exported.

**origin** <*string*>
Specifies whether the origin of the routes to be exported was an interior gateway protocol or an exterior gateway protocol. Specify one of the following:

- **any**
- **igp**
- **egp**
- **incomplete**

**metric** <*num*>
Specifies metric associated with the exported routes.

**restrict**
Specifies that nothing is exported from the specified source.

**Note:** You can specify **metric** or **restrict** even if you specified **protocol**, **aspath-regular-expression**, or **origin**.

## Restrictions

None.

# ip-router policy create bgp-export-destination

## Purpose

Create an export destination for BGP routes.

## Format

**ip-router policy create bgp-export-destination**
*<number-or-string> <option-list>*

## Mode

Configure

## Parameters

*<number-or-string>*
Creates a BGP export destination and associates an identifier (tag) with it.

*<option-list>*
Specifies the BGP export destination options you are setting. Specify the following:

**autonomous-system** *<num>*
Specifies the autonomous system of the peer-group to which we would be exporting. Specify a number from 1 – 65535.

**optional-attribute-list** *<num-or-string>*
Specifies the identifier of the optional-attribute-list which contains the optional attributes which are to be sent along with these exported routes. This option may be used to send the BGP community attribute. Any communities specified in the optional-attributes-list are sent in addition to any received with the route or those specified with the 'set peer-group' or 'set peer-host' commands.

**metric** *<num>*
Specifies the metric to be associated with the BGP exported routes.

**restrict**
Restricts the export of BGP routes to the specified destination.

**sequence-number** *<num>*
Specifies the relative position of this export-destination in a list of bgp export-destinations.

## Restrictions

None.

# ip-router policy create bgp-export-source

## Purpose

Create a source for exporting bgp routes into other protocols.

## Format

**ip-router policy create bgp-export-source** <*number-or-string*> <*option-list*>

## Mode

Configure

## Parameters

<*number-or-string*>
Creates a BGP export source and associates an identifier (tag) with it.

<*option-list*>
Specifies the BGP export source options you are setting. Specify the following:

**autonomous-system** <*num*>
Specifies the autonomous system of the peer-group from which we would be exporting. A route filter could alternatively be used to explicitly list a set of routes to be accepted. Specify a number from 1 – 65534.

**metric** <*num*>
Specifies the metric to be associated with the BGP exported routes.

**restrict**
Restricts the export of BGP routes from the specified source.

## Restrictions

None.

# ip-router policy create bgp-import-source

## Purpose

Create a source for importing BGP routes.

## Format

**ip-router policy create bgp-import-source** <*number-or-string*> <*option-list*>

## Mode

Configure

## Parameters

<*number-or-string*>
Creates a BGP import source and associates an identifier (tag) with it.

<*option-list*>
Specifies the BGP import source options you are setting. Specify the following:

**autonomous-system** <*num*>
Specifies the autonomous system of the peer-group from which we would be exporting. A route filter could alternatively be used to explicitly list a set of routes to be accepted. Specify a number from 1 – 65534.

**aspath-regular-expression** <*string*>
Specifies the as path regular expression that must be satisfied for the route to be exported. A route filter could alternatively be used to explicitly list a set of routes to be announced.

**origin** <*value*>
Specifies the origin attribute. Specify one of the following:

**any** Specifies that the origin attribute can be any one of **igp**, **egp** and **incomplete**.

**igp** Specifies that the origin attribute of the imported routes is IGP.

**egp** Specifies that the origin attribute of the imported routes is EGP.

**incomplete** Specifies that the origin attribute of the imported routes is incomplete.

**optional-attribute-list** <*num-or-string*>
Specifies the identifier of the optional-attribute-list. This option allows the

specification of import policy based on the path attributes found in the BGP update. If multiple communities are specified in the aspath-opt option, only updates carrying all of the specified communities will be matched. If none is specified, only updates lacking the community attribute will be matched.

**preference** <*num*>
Specifies the preference to be associated with the BGP imported routes.

**restrict**
Specifies that nothing is exported from the specified source.

**sequence number** <*num*>
Indicates the position this bgp import source will have in a list of BGP import sources.

## Restrictions

None.

# ip-router policy create direct-export-source

## Purpose

Creates an export source for interface routes.

## Format

**ip-router policy create direct-export-source** <*number-or-string*> **[interface** <*name-or-IPaddr*>**][metric** <*num*> **| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Creates a source for exporting **interface** (**direct**) routes and associates an identifier with it.

**interface**
This option qualifies that the direct routes should be associated with the specific interface.

**metric** <*num*>
Specifies the metric to be associated with the exported routes.

**restrict**
Restricts the export of routes from the specified source.

## Restrictions

None.

# ip-router policy create filter

## Purpose

Creates a route filter. Routes are filtered by specifying a set of filters that will match a certain set of routes by destination, or by destination and mask.

## Format

**ip-router policy create filter** <*number-or-string*> **network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**][host-net]**

## Mode

Configure

## Parameters

**filter** <*number-or-string*>
Specifies the identifier of the route filter.

**network** <*IP-address*>
This option specifies networks which are to be filtered. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be filtered are specified, then any destination that falls in the range implied by this network-specification is matched, so the mask of the destination is ignored. If a natural network is specified, the network, any subnets, and any hosts will be matched. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

**exact**
This option specifies that the mask of the routes to be filtered must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

**refines**
This option specifies that the mask of the routes to be filtered must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

**between** <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as longer) than the lower limit (the first number parameter) and no more specific (i.e. as long as or shorter) than the upper limit (the second parameter).

**host-net**
This option qualifies that the specified network is a host. To match, the address must exactly match the specified and the network mask must be a host mask (i.e. all ones). This is equivalent to a network specification of host/255.255.255.255 along with the exact option.

## Restrictions

None.

# ip-router policy create optional-attributes-list

## Purpose

Creates an optional-attributes-list for BGP.

## Format

**ip-router policy create optional-attributes-list** *<option-list>*

## Mode

Configure

## Parameters

*<option-list>*
Specifies the options you are setting. Specify the following:

*<number-or-string>*
Specifies the identifier for the attributes list.

**community-id** *<number>*
Specifies a community identifier portion of a community split. This is combined with the autonomous system value entered to create a value for the community attribute.

**autonomous-system** *<number>*
Specifies the autonomous system portion of a community split. This would be combined with the community id value entered to create a value for the community attribute. Specify a number from 1 – 65534.

**no-export**
Specifies that all routes received with this attribute value ***will not*** be advertised outside a BGP confederation boundary.

**well-known-community**
Specifies one of the well-known communities.

**no-advertise**
Specifies that all routes received with this attribute value ***will not*** be advertised to other BGP peers.

**no-export-subconfed**
Specifies that all routes received with this attribute value ***will not*** be advertised to

external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).

**reserved-community** <*number*>
Specifies one of the reserved communities which is not well-known. A reserved community is one which is in one of the following ranges (0x0000000 - 0x0000FFFF) or (0xFFFF0000 - 0xFFFFFFFF).

## Restrictions

None.

# ip-router policy create ospf-export-destination

## Purpose

Create a destination for exporting routes into OSPF.

## Format

**ip-router policy create ospf-export-destination**
*<number-or-string>* **[tag** *<num>***][type 1 | 2][metric** *<num>* **| restrict]**

## Mode

Configure

## Parameters

*<number-or-string>*
Creates an OSPF export destination and associates an identifier with it.

**tag** *<num>*
Tag to be associated with exported OSPF routes.

**type 1 | 2**
Specifies that OSPF routes to be exported are type 1 or type 2 ASE routes. Specify 1 or 2.

**metric** *<num>*
Specifies the metric to be associated with the exported routes.

**restrict**
Restricts the export of the specified routes.

## Restrictions

It is not possible to create OSPF intra- or inter-area routes by exporting routes from the routing table into OSPF. You can only export from the routing table into OSPF ASE routes.

# ip-router policy create ospf-export-source

## Purpose

Create a source for exporting OSPF routes into other protocols.

## Format

**ip-router policy create ospf-export-source**
<*number-or-string*> **[type ospf | ospf-ase][metric** <*num*>**| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Creates an OSPF export source and associates an identifier with it.

**type ospf**
Exported routes are OSPF routes.

**type ospf-ase**
Exported routes are OSPF ASE routes.

**metric** <*num*>
Specifies the metric to be associated with the exported routes.

**restrict**
Specifies that nothing is to be exported from this source.

## Restrictions

None.

# ip-router policy create ospf-import-source

## Purpose

Create a source for importing OSPF routes.

## Format

**ip-router policy create ospf-import-source** <*number-or-string*> **[tag** <*num*>**][preference** <*num*> **| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Creates an OSPF import source and associates an identifier with it.

**tag** <*num*>
Tag to be associated with the imported routes.

**preference** <*num*>
Preference associated with the imported OSPF routes.

**restrict**
Specifies that matching **ospf-ase** routes are not imported.

## Restrictions

None.

# ip-router policy create rip-export-destination

## Purpose

Create a destination for exporting routes into RIP.

## Format

**ip-router policy create rip-export-destination** <*number-or-string*>
**[interface** <*name-or-IPaddr*> **| gateway** <*name-or-IPaddr*>**] [metric** <*num*> **| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Specifies an identifier for the RIP export destination:

**interface** <*name-or-IPaddr*> **| all**
Specifies router interfaces over which to export routes. Specify **all** to export routes to all interfaces.

**gateway** <*name-or-IPaddr*>
Specifies the gateway that will receive the exported routes.

**metric** <*num*>
Specifies the metric to be associated with the exported routes. Specify a number from 1 – 16.

**restrict**
Restricts the export of routes to the specified destination.

## Restrictions

None.

# ip-router policy create rip-export-source

## Purpose

Create a source for exporting RIP routes into other protocols

## Format

**ip-router policy create rip-export-source**
*<number-or-string>* **[interface** *<name-or-IPaddr>* **| gateway** *<name-or-IPaddr>***][metric** *<num>* **| restrict]**

## Mode

Configure

## Parameters

*<number-or-string>*
Specifies an identifier for the RIP export source:

**interface** *<name-or-IPaddr>*
Indicates that only routes learned over specified interfaces are exported.

**gateway** *<name-or-IPaddr>*
Indicates that only routes learned over specified gateways are exported.

**metric** *<num>*
Specifies the metric to be associated with the exported routes.

**restrict**
Indicates that nothing is exported from the specified source.

## Restrictions

None.

# ip-router policy create rip-import-source

## Purpose

Create a source for importing RIP routes.

## Format

**ip-router policy create rip-import-source** <*number-or-string*>
**[interface** <*name-or-IPaddr*> **| gateway** <*name-or-IPaddr*>**][preference** <*num*> **| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Specifies an identifier for the RIP import source:

**interface** <*name-or-IPaddr*>
Indicates that only routes learned over specified interfaces are imported.

**gateway** <*name-or-IPaddr*>
Indicates that only routes learned over specified gateways are imported.

**preference** <*num*>
Specifies the preference to be associated with the imported routes.

**restrict**
Indicates that nothing is imported from the specified source.

## Restrictions

None.

# ip-router policy create static-export-source

## Purpose

Creates a source for exporting static routes into other protocols.

## Format

**ip-router policy create static-export-source** <*number-or-string*>
**[interface** <*name-or-IPaddr*>**][metric** <*num*>**| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Creates a source for exporting **static** routes and associates an identifier with it.

**interface**
This option qualifies that the **static** routes should be associated with the specific interface.

**metric** <*num*>
Specifies the metric to be associated with the exported routes.

**restrict**
Restricts the export of routes from the specified source.

## Restrictions

None.

# ip-router policy create tag-export-source

## Purpose

Create an export source where routes to be exported are identified by the tag associated with them.

## Format

**ip-router policy create tag-export-source** <*number-or-string*>
**protocol all | static | direct | aggregate | rip | ospf | bgp**
**[tag** <*number*>**][metric** <*number*>**| restrict]**

## Mode

Configure

## Parameters

<*number-or-string*>
Specifies the identifier of an tag-export source.

**protocol** <*string*>
Specifies the protocol of the contributing source. Specify one of the following:

- **all**
- **static**
- **direct**
- **aggregate**
- **rip**
- **ospf**
- **bgp**

**tag** <*number*>
Restricts selection of routes to those identified by a tag.

**metric** <*number*>
Specifies the metric to assign to the exported routes.

**restrict**
Indicates that the matching routes are not exported.

## Restrictions

None.

# ip-router policy export destination

## Purpose

Creates an export policy from the various building blocks.

## Format

**ip-router policy export destination** <*exp-dest-id*>
[**source** <*exp-src-id*> [**filter** <*filter-id*> | [**network** <*ipAddr/mask*> [**exact** | **refines** | **between** <*low-high*>] [**metric** <*number*> | **restrict**]]]]

## Mode

Configure

## Parameters

<*exp-dest-id*>
Is the identifier of the export-destination which determines where the routes are to be exported. If no routes to a particular destination are to be exported, then no additional parameters are required.

<*exp-src-id*>
If specified, is the identifier of the export-source which determines the source of the exported routes. If a export-policy for a given export-destination has more than one export-source, then the *ip-router policy export destination* <*exp-dest-id*> command should be repeated for each <*exp-src-id*>.

<*filter-id*>
If specified, is the identifier of the route-filter associated with this export-policy. If there is more than one route-filter for any export-destination and export-source combination, then the *ip-router policy export destination* <*exp-dest-id*> *source* <*exp-src-id*> command should be repeated for each <*filter-id*>.

**network** <*ipAddr/mask*>
Specifies networks which are to be exported. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be exported are specified, then any destination that falls in the range implied by this network-specification is matched, so the mask of the destination is ignored. If a natural network is specified, the network, any subnets, and any hosts will be matched. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

**exact**
This option specifies that the mask of the routes to be exported must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

**refines**
This option specifies that the mask of the routes to be exported must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

**between** <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as longer) than the lower limit (the first number parameter) and no more specific (i.e. as long as or shorter) than the upper limit (the second parameter).

**restrict**
Specifies that routes matching the filter are not to be exported.

**metric** <*number*>
Specifies the metric to be associated with the routes that match the specified filter.

## Restrictions

None.

# ip-router policy import source

## Purpose

Creates an import policy.

## Format

**ip-router policy import source** <*imp-src-id*> **[filter** <*filter-id*>**| [network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**] [preference** <*number*> **| restrict]]]**

## Mode

Configure

## Parameters

<*imp-src-id*>
Is the identifier of the import-source that determines the source of the imported routes. If no routes from a particular source are to be imported, then no additional parameters are required.

<*filter-id*>
If specified, is the identifier of the route-filter associated with this import-policy. If there is more than one route-filter for any import-source, then the *ip-router policy import source* <*imp-src-id*> command should be repeated for each <*filter-id*>.

**network** <*ipAddr/mask*>
Specifies networks which are to be imported. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be imported are specified, then any destination that falls in the range implied by this network-specification is matched, so the mask of the destination is ignored. If a natural network is specified, the network, any subnets, and any hosts will be matched. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

**exact**
This option specifies that the mask of the routes to be imported must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

**refines**
This option specifies that the mask of the routes to be imported must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

**between** <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as longer) than the lower limit (the first number parameter) and no more specific (i.e. as long as or shorter) than the upper limit (the second parameter).

**restrict**
Specifies that routes matching the filter are not to be imported.

**preference** <*number*>
Specifies the preference with which the imported routes that match the specified filter should be installed.

## Restrictions

None.

# ip-router policy redistribute

## Purpose

Creates a simple route redistribution policy

## Format

**ip-router policy redistribute from-proto** <*protocol*> **to-proto** <*protocol*> **[network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**]] [metric** <*number*> **| restrict] [source-as** <*number*>**] [target-as** <*number*>**] [tag] [ase-type]**

## Mode

Configure

## Parameters

**from-proto** <*protocol*>
Specifies the protocol of the source routes. The values for the from-proto parameter are **rip**, **ospf**, **bgp**, **direct**, **static**, **aggregate**, or **ospf-ase**.

**to-proto** <*protocol*>
Specifies the destination protocol where the routes are to be exported. The values for the to-proto parameter are **rip**, **ospf**, or **bgp**.

**network** <*ipAddr/mask*>
Provides a means to define a filter for the routes to be distributed. The network parameter defines a filter that is made up of an IP address and a mask. Routes that match the filter are considered as eligible for redistribution.

Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be redistributed are specified, then any destination that falls in the range implied by this network-specification is matched, so the mask of the destination is ignored. If a natural network is specified, the network, any subnets, and any hosts will be matched. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

**exact**
This option specifies that the mask of the routes to be redistributed must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network.

**refines**
This option specifies that the mask of the routes to be redistributed must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

**between** <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as longer) than the lower limit (the first number parameter) and no more specific (i.e. as long as or shorter) than the upper limit (the second parameter).

**restrict**
Specifies that routes matching the filter are not to be redistributed.

**metric**
Indicates the metric to be associated with the redistributed routes.

tag
Tag to be associated with the exported OSPF routes.

ase-type
Routes exported from the GateD routing table into OSPF default to becoming type 1 ASEs. This default may be explicitly overridden here. Thus, this option should be used to specify if the routes are to be exported as OSPF Type 1 or Type 2 ASE routes.

**Note:** Each protocol (RIP, OSPF, and BGP) has a configurable parameter that specifies the default-metric associated with routes exported to that protocol. If a metric is not explicitly specified with the redistribute command, then it is inherited from the default-metric associated with the protocol to which the routes are being exported.

## Restrictions

None.

# ip-router show configuration file

## Purpose

Display the active or startup configuration file in GateD format.

## Format

**ip-router show configuration-file active | permanent**

## Mode

Enable

## Parameters

| | |
|---|---|
| **active** | Shows the active GateD configuration file in RAM; this is the default. |
| **permanent** | Shows the permanent GateD configuration file in NVRAM, if available. |

## Restrictions

None.

# ip-router show rib

## Purpose

Display routing information base.

## Format

**ip-router show rib [detail]**

## Mode

Enable

## Description

The **ip-router show rib** command shows the route-manager's routing information base (RIB). For any given network, the routing daemon could have multiple routes. The active route to any network is shown with a plus (+) sign next to it. The last active route is shown with a minus (-) next to it. If a route has been the last active route and is also the current active route, then it is shown with a asterisk (*) sign next to it. The legend is as follows:

- "+" Active Route
- "-" Last Active
- "*" Both

If the detail option is used, then additional information is displayed about these routes. The announcements bits for the active route are shown which shows the protocol into which this route is advertised.

## Parameters

**detail** Allows you to view additional information about the routes in the RIB.

## Restrictions

None.

## Examples:

A sample output of the **ip-router show rib detail** command is shown below:

```
10.12.1          mask 255.255.255.252
entries 2    announce 1
TSI:
RIP 150.1.255.255mc  <> metric 1
RIP 222.1.1.255mc  <> metric 1
BGP_Sync_64805 dest 10.12.1/2 metric 0
BGP group type Routing AS 64805 no metrics
Instability Histories:

*Direct    Preference:   0
*NextHop: 10.12.1.2          Interface: 10.12.1.2(to-c4500)
State: <Int Active Retain>
Age: 5:12:10    Metric: 0    Metric2: 0    Tag: 0
Task: IF
Announcement bits(5):
2-KRT 4-RIP.0.0.0.0+520 5-RIP.0.0.0.0+520
6-BGP_Sync_64805
7-BGP_Group_64805
AS Path: IGP (Id 1)

OSPF    Preference: -10
*NextHop: 10.12.1.1          Interface: 10.12.1.2(to-c4500)
State: <NotInstall NoAdvise Int Hidden Gateway>
Local AS: 64805
Age: 1:20:05   Metric: 1    Metric2: -1    Tag: 0
Task: OSPF
AS Path: (64805) IGP (Id 9551)
Cost: 1    Area: 0.0.0.0    Type: Net    AdvRouter:
172.23.1.14
```

In this case there two routes to network 10.12.1.0/255.255.255.252 One of them is a direct route and other route is learned through OSPF. The direct route has a better preference (lower preference is considered better preference), and is thus the active route. The direct route has been installed since 5 hours, 12 minutes and 10 seconds. This direct route is being announced to the Forwarding Information Base (FIB) which is indicated by KRT, over two RIP interfaces (which is indicated by 4-RIP.0.0.0.0+520, 5-RIP.0.0.0.0+520) and also to the BGP internal peer-group for autonomous system 64805. All the routes in the RIB are shown by this command.

To see a specific route, use the **ip-router show route** command.

# ip-router show route

## Purpose

Displays the state of GateD.

## Format

**ip-router show route [ip-addr-mask | default] [detail]**

## Mode

Enable

## Description

This command shows a specific route in the route-manager's routing information base (RIB). For any given network, the routing daemon could have multiple routes. The active route to any network is shown with a plus (+) sign next to it. The last active route is shown with a minus (-) next to it. If a route has been the last active route and is also the current active route, then it is shown with a asterisk (*) sign next to it. The legend is as follows:

- "+" Active Route
- "-" Last Active
- "*" Both

If the detail option is used, then additional information is displayed about this routes. The announcements bits for the active route are shown which shows the protocol into which this route is advertised.

## Parameters

*<ipAddr/mask>* | **default**
Allows you to specify a particular IP address mask for the RIB route in question, or refer to the default address mask.

**detail**
Allows you to view additional information about the routes in the RIB.

## Restrictions

None.

## Examples

A sample output of the **ip-router show rib detail** command is shown below.

```
10.12.1          mask 255.255.255.252
entries 2    announce 1
TSI:
RIP 150.1.255.255mc  <> metric 1
RIP 222.1.1.255mc  <> metric 1
BGP_Sync_64805 dest 10.12.1/2 metric 0
BGP group type Routing AS 64805 no metrics
Instability Histories:

*Direct    Preference:   0
*NextHop: 10.12.1.2          Interface: 10.12.1.2(to-c4500)
State: <Int Active Retain>
Age: 5:12:10    Metric: 0    Metric2: 0    Tag: 0
Task: IF
Announcement bits(5):
2-KRT 4-RIP.0.0.0.0+520 5-RIP.0.0.0.0+520
6-BGP_Sync_64805
7-BGP_Group_64805
AS Path: IGP (Id 1)

OSPF    Preference: -10
*NextHop: 10.12.1.1          Interface: 10.12.1.2(to-c4500)
State: <NotInstall NoAdvise Int Hidden Gateway>
Local AS: 64805
Age: 1:20:05   Metric: 1    Metric2: -1    Tag: 0
Task: OSPF
AS Path: (64805) IGP (Id 9551)
Cost: 1    Area: 0.0.0.0    Type: Net    AdvRouter:
172.23.1.14
```

In this case there two routes to network 10.12.1.0/255.255.255.252 One of them is a direct route and other route is learned through OSPF. The direct route has a better preference (lower preference is considered better preference), and is thus the active route. The direct route has been installed since 5 hours, 12 minutes and 10 seconds. This direct route is being announced to the Forwarding Information Base (FIB) which is indicated by KRT, over two RIP interfaces (which is indicated by 4-RIP.0.0.0.0+520, 5-RIP.0.0.0.0+520) and also to the BGP internal peer-group for autonomous system 64805.

To see all the routes in the RIB, use the **ip-router show rib** command.

# ip-router show state

## Purpose

Displays the state of GateD.

## Format

**ip-router show state to-file | to-terminal**

## Mode

Enable

## Parameters

| | |
|---|---|
| **to-file** | Saves the routing-process state in the `gated.dmp` file. |
| **to-terminal** | Displays the routing-process state on the console. |

## Restrictions

None.

# Chapter 20

# ip-redundancy Commands

The **ip-redundancy** commands let you display and configure the Virtual Router Redundancy Protocol (VRRP) on the GSR. VRRP is defined in RFC 2338.

## Command Summary

Table 15 lists the **ip-redundancy** commands. The sections following the table describe the command syntax.

**Table 15. ip-redundancy commands**

| |
|---|
| **ip-redundancy associate vrrp** <*vrid*> **interface** <*interface*> **id** <*vrid*> |
| **ip-redundancy clear vrrp-stats interface** <*interface*> **id** <*vrid*> |
| **ip-redundancy create vrrp** <*vrid*> **interface** <*interface*> |
| **ip-redundancy set vrrp** <*vrid*> **interface** <*interface*> <*option*> |
| **ip-redundancy show vrrp interface** <*interface*> **id** <*vrid*> |
| **ip-redundancy start vrrp** <*vrid*> **interface** <*interface*> |
| **ip-redundancy trace vrrp** <*option*> |

# ip-redundancy associate

## Purpose

Associates an IP address with a virtual router.

## Format

**ip-redundancy associate vrrp** <*vrid*> **interface** <*interface*> **address** <*ipaddr/mask*>

## Mode

Configure

## Description

The **ip-redundancy associate** command adds an IP address to the list of IP addresses associated with a virtual router.

## Parameters

| | |
|---|---|
| <*vrid*> | Is the identifier of a virtual router. Specify a number between 1-255 |
| <*interface*> | Is the name of the interface where the virtual router resides. |
| <*ipaddr/mask*> | Is the IP address and subnet mask to be associated with the virtual router. |

## Restrictions

None

## Example

To add IP address/mask 1.2.3.4/16 to the list of IP addresses associated with virtual router 1 on interface test1:

```
gs/r(config)# ip-redundancy associate vrrp 1 interface test1 address
1.2.3.4/16
```

# ip-redundancy clear vrrp-stats

## Purpose

Clears statistics gathered for VRRP.

## Format

**ip-redundancy clear vrrp-stats interface** <*interface*> **[id** <*vrid*>**]**

## Mode

Enable

## Description

The **ip-redundancy clear vrrp-stats** command is used in conjunction with the **ip-redundancy show vrrp** command, which displays information about the virtual routers associated with an interface. When you specify the **verbose** option with the **ip-redundancy show vrrp** command, additional statistics are shown, including the number of times a Backup router became the Master, the number of VRRP advertisements received, and counts of VRRP packets that contain errors. When you run the **ip-redundancy clear vrrp-stats** command, these statistics are reset to zero.

## Parameters

<*interface*> Causes VRRP statistics to be cleared for all virtual routers on the specified interface.

<*vrid*> Causes VRRP statistics to be cleared for the virtual router with the specified VRID. Enter a number between 1-255.

## Restrictions

None.

## Example

To clear statistics for virtual router 1 on interface test1:

```
gs/r# ip-redundancy clear vrrp-stats interface test1 id 1
```

# ip-redundancy create

## Purpose

Creates a virtual router.

## Format

**ip-redundancy create vrrp** <*vrid*> **interface** <*interface*>

## Mode

Configure

## Description

The **ip-redundancy create** command creates a virtual router on a specified interface.

## Parameters

<*vrid*> Is the identifier of the virtual router to create. Specify a number between 1-255.

<*interface*> Is the interface on which to create the virtual router.

## Restrictions

None.

## Example

To create a virtual router with an identifier (VRID) of 1 on interface test1:

```
gs/r(config)# ip-redundancy create vrrp 1 interface test1
```

# ip-redundancy set

## Purpose

Sets parameters for a virtual router.

## Format

**ip-redundancy set vrrp** <*vrid*> **interface** <*interface*> **priority** <*number*> | **adv-interval** <*number*> | **preempt-mode enabled** | **disabled** | **auth-type none** | **text auth-key** <*key*>

## Mode

Configure

## Description

The **ip-redundancy set** command lets you specify parameters for a virtual router, including backup priority, advertisement interval, whether the router can preempt a Master router that has a lower priority, and the type of authentication used.

## Parameters

| | |
|---|---|
| <*vrid*> | Is the identifier of a virtual router. Specify a number between 1-255. |
| <*interface*> | Is the name of the interface where the virtual router resides. |
| **priority** <*number*> | Specifies the backup priority to be used by this virtual router. This number must be between 1-254. The default is 100. The priority number applies only if the virtual router is not the IP address owner. The priority of the IP address owner is always 255 and cannot be changed. |
| **adv-interval** <*number*> | Is the interval between VRRP advertisements in seconds. The default is 1 second. |
| **preempt-mode** | Specifies whether the router can preempt a Master router that has a lower priority. Use one of the following keywords: |

**enabled** Preempt mode is enabled. A backup router can pre-empt a lower-priority Master router.

**disabled** Pre-empt mode is disabled. A backup router cannot pre-empt a lower-priority Master router.

**auth-type** Specifies the type of authentication used for VRRP exchanges between routers. Use one of the following keywords:

**none** VRRP exchanges are not authenticated (the default).

**text** VRRP exchanges are authenticated with a clear-text password.

**auth-key** <*key*> Is the clear-text password used to authenticate VRRP exchanges. If you specify the **text** keyword, you must also specify the **auth-key** parameter.

## Restrictions

None.

## Examples

To specify 200 as the priority used by virtual router 1 on interface test1:

```
gs/r(config)# ip-redundancy set vrrp 1 interface test1 priority 200
```

To set the advertisement interval to 3 seconds:

```
gs/r(config)# ip-redundancy set vrrp 1 interface test1 adv-interval 3
```

To prevent a Backup router from taking over as Master from a Master router that has a lower priority:

```
gs/r(config)# ip-redundancy set vrrp 1 interface test1 preempt-mode
disabled
```

To authenticate VRRP exchanges on virtual router 1 on interface test1 with a password of 'digital':

```
gs/r(config)# ip-redundancy set vrrp 1 interface test1 auth-type text auth-key
digital
```

# ip-redundancy show

## Purpose

Shows information about virtual routers.

## Format

**ip-redundancy show vrrp interface** <*interface*> **[id** <*vrid*>**] [verbose]**

## Mode

Enable

## Description

The **ip-redundancy show vrrp** command displays configuration information about virtual routers on an interface. You can display information for one virtual router or for all the virtual routers on an interface. If you specify the verbose option, additional statistics are shown, including the number of times a Backup router became the Master, the number of VRRP advertisements received, and counts of VRRP packets that contain errors. These statistics are gathered from the time you start the virtual router, or from the time you last ran the **ip-redundancy clear vrrp-stats** command.

## Parameters

| | |
|---|---|
| <*interface*> | Is the name of the interface where the virtual router resides. If you do not specify the <*vrid*> parameter, information about all virtual routers on the interface is displayed. |
| <*vrid*> | Is the identifier of a virtual router. Specify a number between 1-255. |
| **verbose** | Causes VRRP statistics to be displayed for each virtual router |

## Restrictions

None.

## Examples

To display information about all virtual routers on interface test1:

```
gs/r# ip-redundancy show vrrp interface test1

VRRP Virtual Router 100 - Interface test1
-----------------------------------------
Uptime                 0 days  0 hours  0 minutes  17 seconds.
State                  Backup
Priority               100 (default value)
Virtual MAC address    00005E:000164
Advertise Interval     1 sec(s) (default value)
Preempt Mode           Enabled (default value)
Authentication         None (default value)
Primary Address        10.8.0.2
Associated Addresses   10.8.0.1
                       100.0.0.1

VRRP Virtual Router 200 - Interface test1
-----------------------------------------
Uptime                 0 days  0 hours  0 minutes  17 seconds.
State                  Master
Priority               255 (default value)
Virtual MAC address    00005E:0001C8
Advertise Interval     1 sec(s) (default value)
Preempt Mode           Enabled (default value)
Authentication         None (default value)
Primary Address        10.8.0.2
Associated Addresses   10.8.0.2
```

To display VRRP statistics for virtual router 100 on interface test1:

```
gs/r# ip-redundancy show vrrp 1 interface test1 verbose

VRRP Virtual Router 100 - Interface test1
-----------------------------------------
Uptime                  0 days  0 hours  0 minutes  17 seconds.
State                   Backup
Priority                100 (default value)
Virtual MAC address     00005E:000164
Advertise Interval      1 sec(s) (default value)
Preempt Mode            Enabled (default value)
Authentication          None (default value)
Primary Address         10.8.0.2
Associated Addresses    10.8.0.1
                        100.0.0.1

Stats:
  Number of transitions to master state           2
  VRRP advertisements rcvd                        0
  VRRP packets sent with 0 priority               1
  VRRP packets rcvd with 0 priority               0
  VRRP packets rcvd with IP-address list mismatch 0
  VRRP packets rcvd with auth-type mismatch       0
  VRRP packets rcvd with checksum error           0
  VRRP packets rcvd with invalid version          0
  VRRP packets rcvd with invalid VR-Id            0
  VRRP packets rcvd with invalid adv-interval     0
  VRRP packets rcvd with invalid TTL              0
  VRRP packets rcvd with invalid 'type' field     0
  VRRP packets rcvd with invalid auth-type        0
  VRRP packets rcvd with invalid auth-key         0
```

# ip-redundancy start vrrp

## Purpose

Starts a virtual router.

## Format

**ip-redundancy start vrrp** <*vrid*> **interface** <*interface*>

## Mode

Configure

## Description

The **ip-redundancy start vrrp** command starts a virtual router on the specified interface.

## Parameters

| | |
|---|---|
| <*vrid*> | Is the identifier of a virtual router. Specify a number between 1-255. |
| <*interface*> | Is the name of the interface where the virtual router resides. |

## Restrictions

None.

## Example

To start virtual router 1 on interface test1:

```
gs/r# ip-redundancy start vrrp 1 interface test1
```

# ip-redundancy trace

## Purpose

Traces VRRP events.

## Format

**ip-redundancy trace vrrp events | state-transitions | packet-errors**

**ip-redundancy trace vrrp all enabled | disabled**

## Mode

Configure

## Description

The **ip-redundancy trace vrrp** command displays messages when certain VRRP events take place on the GSR. Use this command to display messages when a virtual router changes from one state to another (i.e., from Backup to Master), a VRRP packet error is detected, or when any VRRP event occurs.

## Parameters

| | |
|---|---|
| **events** | Displays a message when VRRP receives any type of event. This option is disabled by default. |
| **state-transitions** | Displays a message when a VRRP router changes from one state to another. This option is enabled by default. |
| **packet-errors** | Displays a message when a VRRP packet error is detected. This option is enabled by default. |
| **all enabled \| disabled** | Enables or disables all VRRP tracing. |

## Restrictions

None.

## Example

To display a message whenever a VRRP packet error is encountered:

```
gs/r(config)# ip-redundancy trace vrrp packet-errors enabled
```

# Chapter 21

# ipx Commands

The **ipx** commands let you add entries to the IPX SAP table for SAP servers and display the IPX forwarding database, RIP table, and SAP table.

## Command Summary

Table 16 lists the **ipx** commands. The sections following the table describe the command syntax.

**Table 16. ipx commands**

| |
|---|
| **ipx add route** *<networkaddr> <nextroutnextnode> <metric> <ticks>* |
| **ipx add sap** *<type> <SrvcName> <node> <socket> <metric> <interface-network>* |
| **ipx find rip** *<address>* |
| **ipx find sap** *<entrytype> <type> <SrvcName> <network>* |
| **ipx show interfaces** *<interface>* |
| **ipx show tables routing \| rip \| sap \| summary** |

# ipx add route

## Purpose

Add an IPX RIP route entry to the routing table.

## Format

**ipx add route** <*networkaddr*> <*nextroutnextnode*> <*metric*> <*ticks*>

## Mode

Configure

## Description

The **ipx add route** command adds a route into the IPX RIP routing table.

## Parameters

| | |
|---|---|
| <*networkaddr*> | Destination network address. |
| <*nextroutnextnode*> | Next router's Network.Node address. |
| <*metric*> | The number of hops to this route. You can specify a number from 0 – 14. |
| <*ticks*> | Ticks associated with this route. |

## Restrictions

Route entries that you add using the **ipx add route** command override dynamically learned entries, regardless of hop count.

## Example

To add an IPX route to IPX network A1B2C3F5 via router A1B2C3D4.00:E0:63:11:11:11 with a metric of 1 and a tick of 100:

```
gs/r(config)# ipx add route A1B2C3F5 A1B2C3D4.00:E0:63:11:11:11 1 100
```

# ipx add sap

## Purpose

Add an IPX SAP entry to the routing table.

## Format

**ipx add sap** <*type*> <*SrvcName*> <*node*> <*socket*> <*metric*> <*interface-network*>

## Mode

Configure

## Description

The **ipx add sap** command adds an entry for an IPX server to the IPX SAP table.

## Parameters

| | |
|---|---|
| <*type*> | The type of service. Specify the service type using its hexadecimal value. |
| <*SrvcName*> | Name of the IPX server. You can use any characters in the name except the following: " * . / : ; < = > ? [ ] \ \| ]<br>**Note:** Lowercase characters are changed to uppercase characters. |
| <*node*> | *The* IPX network and node address. Specify the address in the following format: <*netaddr*>**.**<*macaddr*>. Example: a1b2c3d4.aa:bb:cc:dd:ee:ff. |
| <*socket*> | The socket number for this SAP entry. You can specify a Hexadecimal number from 0x0 – 0xFFFF. |
| <*metric*> | The number of hops to the server. You can specify a number from 1 – 14. |
| <*interface-network*> | The interface network associated with this SAP entry. |

## Restrictions

SAP entries that you add using the **ipx add sap** command override dynamically learned entries, regardless of hop count. Moreover, if a dynamic route entry that is associated with the static SAP entry ages out or deleted, the GSR does not advertise the corresponding static SAP entries for the service until it relearns the route.

# ipx find rip

## Purpose

Find an IPX address in the routing table.

## Format

**ipx find rip** <*address*>

## Mode

Configure

## Description

The **ipx find rip** command searches for an IPX address in the routing table.

## Parameter

<*address*> The IPX network address of this interface. Specify the IPX address using its hexadecimal value.

## Restrictions

None.

## Example

To find an IPX network in the route table:

```
gs/r(config)# ipx find rip A1B2C3F5
```

# ipx find sap

## Purpose

Find a SAP entry in the routing table.

## Format

**ipx find rip** <*entrytype*> <*type*> <*SrvcName*> <*network*>

## Mode

Configure

## Description

The **ipx find sap** command searches for a SAP entry in the routing table.

## Parameters

<*entrytype*> The types of entry you want to find. Specify one of the following:

**all** Finds static and dynamic SAP entries.

**dynamic** Finds only the dynamic SAP entries.

**static** Finds only the static SAP entries.

<*type*> The type of service. Specify the service type using its hexadecimal value.

<*SrvcName*> Name of the IPX service. You can use any characters in the name except the following: " * . / : ; < = > ? [ ] \ |

**Note:** Lowercase characters are changed to uppercase characters.

<*network*> Network on which the service resides. Specify the address in the following format: <*netaddr.*> Example: a1b2c3d4.

## Restrictions

None.

## Example

To find a SAP entry in the route table:

```
gs/r(config)# ipx find sap dynamic 4 FILESERVER a2b2c3d4
```

# ipx show interfaces

## Purpose

Display the configuration of IPX interfaces.

## Format

**ipx show interfaces** <*interface*>

## Mode

Enable

## Description

The **ipx show interfaces** command displays the configuration of an IPX interface. If you issue the command without specifying an interface name then the configuration of all IPX interfaces is displayed.

## Parameters

<*interface*> Name of the IPX interface; for example, gs/r14.

## Restrictions

If you specify an interface name, the name must belong to an existing IPX interface.

## Example

To display the configuration of all IPX interfaces:

```
gs/r# ipx show interfaces
gs/r12:
flags=9863<UP BROADCAST NOTRAILERS RUNNING SIMPLEX LINK0 MULTICAST>
        VLAN: _VLAN-1
        Ports: et.1.7
        IPX: A1B2C3D4.00:E0:63:11:11:11
gs/r14:
flags=9863<UP BROADCAST NOTRAILERS RUNNING SIMPLEX LINK0 MULTICAST>
        VLAN: _VLAN-2
        Ports: et.1.2
        IPX: ABCD1234.00:E0:63:11:11:11
```

# ipx show tables

## Purpose

Show IPX routing information.

## Format

**ipx show tables routing | rip | sap | summary**

## Mode

User

## Description

The **ipx show tables** command displays the IPX forwarding information base, the IPX RIP table, or the IPX SAP table.

## Parameters

| | |
|---|---|
| **routing** | Shows the IPX routing table. |
| **rip** | Shows the IPX RIP table. |
| **sap** | Shows the IPX SAP table. |
| **summary** | Shows a summary of the IPX RIP/SAP table. |

## Restrictions

None.

# Chapter 22

# l2-tables Commands

The **l2-tables** commands let you display various L2 tables related to MAC addresses.

## Command Summary

Table 17 lists the **l2-tables** commands. The sections following the table describe the command syntax.

**Table 17. l2-tables commands**

| |
|---|
| **l2-tables show all-flows [vlan** <*VLAN-num*> **[source-mac** <*MACaddr*>**]] [undecoded]** |
| **l2-tables show all-macs [verbose [undecoded]]**<br>**[vlan** <*VLAN-num*>**] [source] [destination] [multicast]** |
| **l2-tables show bridge-management** |
| **l2-tables show igmp-mcast-registrations [vlan** <*VLAN-num*>**]** |
| **l2-tables show mac** <*MACaddr*> **vlan** <*VLAN-num*> |
| **l2-tables show mac-table-stats** |
| **l2-tables show port-macs** <*port-list*> **\| all-ports**<br>**[[vlan** <*VLAN-num*>**] [source] [destination] [multicast] [undecoded] [no-stats] verbose]** |
| **l2-tables show vlan-igmp-status vlan** <*VLAN-num*> |

# l2-tables show all-flows

## Purpose

Show all L2 flows (for ports in flow-bridging mode).

## Format

**l2-tables show all-flows [vlan** <*VLAN-num*> **[source-mac** <*MACaddr*>**]] [undecoded]**

## Mode

User or Enable

## Description

The **l2-tables show all-flows** command shows all the L2 flows learned by the GSR. The GSR learns flows on ports that are operating in flow-bridging mode.

## Parameters

**vlan** <*VLAN-num*>
The VLAN number associated with the flows. The VLAN number can be from 1 – 4095.

**source-mac** <*MACaddr*>
The source MAC address of the flows. Specify the MAC address in either of the following formats:

xx:xx:xx:xx:xx:xx
xxxxxx:xxxxxx

**undecoded**
Prevents the gs/r from displaying the vendor names with the MAC addresses. Instead, the OUI of each MAC address is displayed "as is," in hexadecimal format. If you do not use this option, the gs/r decodes the OUI and displays the vendor name.

## Restrictions

None.

# l2-tables show all-macs

## Purpose

Show all MAC addresses currently in the L2 tables.

## Format

**l2-tables show all-macs [verbose [undecoded]]**
**[vlan** <*VLAN-num*>**] [source] [destination] [multicast]**

## Mode

User or Enable

## Description

The **l2-tables show all-macs** command shows how many MAC addresses the GSR has in its L2 tables. You can format the displayed information based on VLAN, source MAC address, destination MAC address or multicast. If you enter the verbose option, the command also shows the individual MAC addresses.

## Parameters

| | |
|---|---|
| **vlan** <*VLAN-num*> | Displays only MAC addresses in the specified VLAN. |
| **source** | Displays only source addresses. |
| **destination** | Displays only destination addresses. |
| **multicast** | Displays only multicast and broadcast addresses. |
| **verbose** | Shows detailed information for each MAC address entry. |
| **undecoded** | Prevents the GSR from displaying the vendor names with the MAC addresses. Instead, the OUI of each MAC address is displayed "as is," in hexadecimal format. If you do not use this option, the GSR decodes the OUI and displays the vendor name. |

## Restrictions

None.

# l2-tables show bridge-management

## Purpose

Show information about all MAC addresses registered by the system.

## Format

**l2-tables show bridge-management**

## Mode

User or Enable

## Description

The **l2-tables show bridge-management** command shows MAC addresses that have been inserted into the L2 tables for management purposes. Generally, these entries are configured so that a port forwards a frame to the Control Module if the management MAC matches the frame's destination MAC.

An example of a bridge-management MAC is Spanning Tree's bridge group address (0180C2:000000), which is be registered in the L2 tables of GSR ports on which the Spanning Tree Protocol (STP) is enabled.

## Parameters

None.

## Restrictions

None.

# l2-tables show igmp-mcast-registrations

## Purpose

Show information about multicast MAC addresses registered by IGMP.

## Format

**l2-tables show igmp-mcast-registrations [vlan** <*VLAN-num*>**]**

## Mode

User or Enable

## Description

The **l2-tables show igmp-mcast-registrations** command shows the multicast MAC addresses that IGMP has registered with the L2 tables. The GSR forwards the multicast MAC addresses only to the ports that IGMP specifies.

## Parameters

**vlan** <*VLAN-num*> Displays only the multicast MAC addresses registered for the specified VLAN.

## Restrictions

None.

# l2-tables show mac

## Purpose

Show information about a particular MAC address.

## Format

**l2-tables show mac** <*MACaddr*> **vlan** <*VLAN-num*>

## Mode

User or Enable

## Description

The **l2-tables show mac** command shows the port number on which the specified MAC address resides.

## Parameters

<*MACaddr*> Is a MAC address. You can specify the address in either of the following formats:

xx:xx:xx:xx:xx:xx
xxxxxx:xxxxxx

**vlan** <*VLAN-num*>Displays the MAC address for this VLAN.

## Restrictions

None.

# l2-tables show mac-table-stats

## Purpose

Show statistics for the MAC addresses in the MAC address tables.

## Format

**l2-tables show mac-table-stats**

## Mode

User or Enable

## Description

The **l2-tables show mac-table-stats** command shows statistics for the master MAC address table in the Control Module and the MAC address tables on the individual ports.

## Parameters

None.

## Restrictions

None.

# l2-tables show port-macs

## Purpose

Show information about MACs residing in a port's L2 table.

## Format

**l2-tables show port-macs** <*port-list*> | **all-ports**
**[[vlan** <*VLAN-num*>**] [source] [destination] [multicast] [undecoded] [no-stats] verbose]**

## Mode

User or Enable

## Description

The **l2-tables show port-macs** command shows the information about the learned MAC addresses in individual L2 MAC address tables. Each port has its own MAC address table. The information includes the number of source MAc addresses and the number of destination MAC addresses in the table. If you enter the **verbose** option, the MAC addresses also are displayed.

## Parameters

**port** <*port-list*> | **all-ports**
Specifies the port(s) for which you want to display MAC address information. You can specify a single port or a comma-separated list of ports. If you use the **all-ports** keyword, MAC address information is displayed for all ports.

**vlan** <*VLAN-num*>
Specifies the type of MAC address for which you want to show statistics.

**source**
Displays statistics for only source addresses.

**destination**
Displays statistics for only destination addresses.

**multicast**
Displays statistics for only multicast and broadcast addresses.

**undecoded**
Displays the MAC addresses in hexadecimal format rather than undecoded format. Undecoded format dos not show the vendor name in place of the first three hexadecimal digits (example: DIGITAL:33:44:55). The default is undecoded (example: 00:11:22:33:44:55).

**no-stats**
Lists the MAC addresses without displaying any statistics.

**verbose**
Shows detailed statistics for each MAC address entry.

## Restrictions

None.

# l2-tables show vlan-igmp-status

## Purpose

Show whether IGMP is on or off on a VLAN.

## Format

**l2-tables show vlan-igmp-status vlan** <*VLAN-num*>

## Mode

Enable

## Description

The **l2-tables show vlan-igmp-status** command shows the multicast MAC addresses that IGMP has registered with the L2 tables. This command also shows the ports to which the multicast MAC addresses are forwarded.

**Note:** For IGMP forwarding to occur for a multicast MAC address, IGMP must be enabled on the VLAN with which the MAC address is associated.

## Parameters

**vlan** <*VLAN-num*> The VLAN number. The VLAN number can be from 1 – 4095.

## Restrictions

None.

# Chapter 23

# lfap Commands

The **lfap** commands let you configure the LFAP client on the GSR and manage the Layer-3 IP accounting information that is delivered by TCP to an external server.

## Command Summary

Table 18 lists the **lfap** commands. The sections following the table describe the command syntax.

**Table 18. lfap commands**

| |
|---|
| **lfap set batch-interval** <*number*> |
| **lfap set batch-size** <*number*> |
| **lfap set lost-contact-interval** <*number*> |
| **lfap set poll-interval** <*number*> |
| **lfap set server** <*IP address(es)*> |
| **lfap set server-retry-interval** <*number*> |
| **lfap show all** |
| **lfap show configuration** |
| **lfap show servers** |
| **lfap show statistics** |
| **lfap show status** |
| **lfap start** |

# lfap set batch-interval

## Purpose

Defines the number of seconds between subsequent transmissions of flow creation and deletion information to a FAS.

## Format

**lfap set batch-interval**<*number*>

## Mode

Configure

## Description

The **lfap set batch-interval** command defines the number of seconds between flow creation and deletion transmissions to a FAS.

## Parameter

<*number*> The number of seconds (from 1 to 2,000, inclusive) between transmission of flow creation and deletion information (the interval). The default value is 1.

## Restrictions

None

## Example

To set the interval between flow creation and deletion transmissions to 5 seconds:

```
gs/r(config)# lfap set batch-interval 5
```

# lfap set batch-size

## Purpose

Defines the number of flow creation and deletion records included in batch transmissions to a FAS.

## Format

**lfap set batch-size** <*number*>

## Mode

Configure

## Description

The **lfap set batch-size** command defines the number of flow creation and deletion records included in information transmissions to a FAS.

## Parameter

<*number*> The number of records (from 1 to 2,000, inclusive) contained in a transmission of flow creation and deletion information to a FAS. The default value is 32.

## Restrictions

None

## Example

To set the number of flow creation and deletion records contained in a batch transmission to 256:

```
gs/r(config)# lfap set batch-size 256
```

# lfap set lost-contact-interval

## Purpose

Defines the period of time (in seconds) before the LFAP client realizes it has lost contact with a FAS.

## Format

**lfap set lost-contact-interval** <*number*>

## Mode

Configure

## Description

The lfap set lost-contact-interval command allows you to define the amount of time (in seconds) the LFAP client will wait before realizing it has lost contact with a FAS and declare the connection lost.

## Parameter

<*number*> The number of seconds (from 10 to 2,000, inclusive) the LFAP client waits before realizing that it has lost contact with a FAS. The default value is 60.

## Restrictions

None

## Example

To set the amount of time the LFAP client waits before realizing that it has lost contact with a FAS to 30 seconds:

```
gs/r(config)# lfap set lost-contact-interval 30
```

# lfap set poll-interval

## Purpose

Sets the interval (in minutes) between transmissions of accounting information to the FAS server.

## Format

**lfap set poll-interval** <*number*>

## Mode

Configure

## Description

The **lfap set poll-interval** command allows you to set the time period (in minutes) between subsequent transmissions of accounting data to the FAS server.

## Parameters

<*number*> Defines the number of minutes (from 1 to 1,440, inclusive) between transmissions of accounting data to the FAS server. The default value is 1.

## Restrictions

None

## Example

To set the number of minutes between accounting data transmissions to the FAS server to 15 minutes:

```
gs/r(config)# lfap set poll-interval 15
```

# lfap set server

## Purpose

Sets one or more FAS IP addresses for the LFAP client to contact.

## Format

**lfap set server** *["]<IP address> [<IP address>] [<IP address>]["]*

## Mode

Configure

## Description

The **lfap set server** command allows you to set up to three FAS IP servers for the LFAP client to contact.

## Parameters

*<IP address>* Sets the IP address of the FAS servers to contact. You may specify a maximum of three IP servers in the command line, separating each IP address with a space. However, if you specify more than one IP server, you must surround the IP addresses in the command line with double-quotes. (See "Examples" below.)

## Restrictions

At least one IP server must be configured before the LFAP client can be started. Also, in order to delete an address from the list of IP servers to contact, you must enter a new **lfap set server** command line. (Simply negating the previous **lfap set server** command will not appropriately counter the initial command execution.)

## Examples

To set one IP server to contact:

```
gs/r (config)# lfap set server 5.5.5.5
```

To set three IP servers to contact:

```
gs/r (config)# lfap set server “5.5.5.5 6.6.6.6 7.7.7.7”
```

# lfap set server-retry-interval

## Purpose

Sets the interval (in seconds) between the LFAP client's attempts to restore contact with a lost FAS.

## Format

**lfap set server-retry-interval** <*number*>

## Mode

Configure

## Description

The **lfap set server-retry-interval** command allows you to customize the amount of time (in seconds) the LFAP client should wait before attempting to restore contact with a lost FAS. After the LFAP client has attempted to contact each server, it will then wait the specified number of seconds before attempting to resume contact.

## Parameters

| | |
|---|---|
| <*number*> | The number of seconds (from 1 to 2,000, inclusive) the LFAP client will wait before attempting to re-establish contact with a lost FAS. The default value is 60 seconds. |

## Restrictions

None

## Example

To set the number of seconds between attempts to resume contact with a lost FAS to 45:

```
gs/r(config)# lfap set server-retry-interval 45
```

# lfap show all

## Purpose

Displays all of the pertinent LFAP client data, including status, servers, configuration, and statistics.

## Format

**lfap show all**

## Mode

Enable

## Description

The **lfap show all** command allows you to analyze the current status of the LFAP client and any servers to which it is currently connected. In the output of the command execution, you will find data pertaining to the following aspects of the LFAP client:

- LFAP Client Status (including connection status)
- LFAP Client Flow Accounting Servers (FASs)
- LFAP Client Configuration, including the following:
  - poll interval
  - batch size
  - batch interval
  - lost contact interval
  - server retry interval
- LFAP Client Statistics, including the following:
  - number of servers
  - up time

- connection successes and failures, including the following:
  - messages sent/received
  - lost information
  - flows

## Parameters

None

## Restrictions

None

# lfap show configuration

## Purpose

Displays the current LFAP client configuration information.

## Format

**lfap show configuration**

## Mode

Enable

## Description

The **lfap show configuration** command allows you to view the current configuration of the LFAP client. In the output of the command execution, you will find the following LFAP client configuration data:

- Poll Interval
- Batch Size
- Batch Interval
- Lost Contact Interval
- Server Retry Interval

## Parameters

None

## Restrictions

None

# lfap show servers

## Purpose

Displays a list of server IP addresses to which the LFAP client is connected, or will try to contact.

## Format

**lfap show servers**

## Mode

Enable

## Description

The **lfap show servers** command allows you to view the list of IP servers to which the LFAP client is currently connected, or will attempt to contact. In the output of the command execution, you will find a list of, at most, three IP addresses of associated FASs.

## Parameters

None

## Restrictions

None.

# lfap show statistics

## Purpose

Displays all of the LFAP client statistics on a per-server basis.

## Format

**lfap show statistics**

## Mode

Enable

## Description

The **lfap show statistics** command allows you to view the current statistics of the LFAP client. In the output of the command execution, you will find data pertaining to the following LFAP client statistics:

- number of servers
- up time
- connection successes and failures, including the following:
  - messages sent/received
  - lost information
  - flows

## Parameters

None

## Restrictions

None

# lfap show status

## Purpose

Displays the present status of the LFAP client.

## Format

**lfap show status**

## Mode

Enable

## Description

The **lfap show status** command allows you to view the current status of the LFAP client. In the output of the command execution, you will find the following LFAP client data:

- LFAP Client Status, defined as one of the following:
  - started
  - stopped
  - failed
- Connection Status, defined as one of the following:
  - connection established
  - connection lost
  - trying to connect

## Parameters

None

## Restrictions

None

# lfap start

## Purpose

Starts the LFAP client.

## Format

**lfap start**

## Mode

Configure

## Description

The **lfap start** command issues a command to the LFAP client to attempt to connect to a FAS server in the list.

## Parameters

None

## Restrictions

At least one IP server must be configured before this command can execute successfully.

# Chapter 24

# logout Command

The **logout** command ends the CLI session.

## Format

**logout**

## Mode

All modes

## Description

The **logout** command ends your CLI session. If you have uncommitted changes in the scratchpad, a message warns you that the changes are not saved and gives you an opportunity to cancel the logout and save the changes.

## Parameters

None.

## Restrictions

None.

# Chapter 25

# multicast Commands

The multicast dvmrp commands let you display information about IP multicast interfaces.

## Command Summary

Table 19 lists the multicast commands. The sections following the table describe the command syntax.

**Table 19. multicast commands**

| |
|---|
| **multicast show interface [**<*ipAddr*> | <*hostname*>**]** |
| **multicast show mroutes [child** <*IPaddr*>**] [group** <*ipaddr*>**] [parent** <*IPaddr*>**]** |

# multicast show interface

## Purpose

Display information about IP multicast interfaces.

## Format

**multicast show interface [**<*ipAddr*> | <*hostname*>**]**

## Mode

Enable

## Description

The **multicast show interface** command displays interfaces that are running IGMP or DVMRP.

**Note:** This command is a superset of the **dvmrp show interface** and **igmp show interface** commands.

## Parameters

<*ipAddr*> | <*hostname*> IP address or hostname of the interface.

## Restrictions

None.

## Examples

To display IP multicast information about interface 10.50.89.90:

```
gs/r# multicast show interface 10.50.89.90
```

The following example shows a larger listing.

```
gs/r# multicast show interface

Address: 172.1.1.10 Subnet: 172.1.1/24 Met: 1 Thr: 1
Name : mls15 State: Up Querier Leaf Igmp Dvmrp

Address: 207.135.89.64 Subnet: 207.135.89.0/25 Met: 1 Thr: 1
Name : company State: Up Querier Leaf Igmp Dvmrp
Groups : 224.0.1.12
224.1.127.255
224.0.1.24
224.2.127.253
224.2.127.254

Address: 10.135.89.10 Subnet: 10.135.89.0/25 Met: 1 Thr: 1
Name : test State: Up Querier Igmp Dvmrp
Peer : 10.135.89.67 Flags: 0xe Version: 3.255

Address: 190.1.0.1 Subnet: 190.1/16 Met: 1 Thr: 1
Name : rip State: Dis

Address: 207.135.122.11 Subnet: 207.135.122.8/29 Met: 1 Thr: 1
Name : mbone State: Up Igmp Dvmrp
Peer : 207.135.122.10 Flags: 0xe Version: 3.255
Groups : 224.0.1.11
224.0.1.12
224.2.127.254
239.255.255.255
224.2.127.253

Address: 10.40.1.10 Subnet: 10.40.1/24 Met: 1 Thr: 1
Name : downstream State: Up Dvmrp
Peer : 10.40.1.1 Flags: 0xf Version: 3.255

Address: 10.100.1.1 Subnet: 10.100.1/24 Met: 1 Thr: 1
Name : dan State: Dn Dvmrp
```

# multicast show mroutes

## Purpose

Display the IP multicast routing table.

## Format

**multicast show mroutes [child** <*IPaddr*>**] [group** <*ipaddr*>**] [parent** <*IPaddr*>**]**

## Mode

Enable

## Description

The **multicast show mroutes** command displays the IP multicast routing table entry for the specified multicast group address.

This command lists all the multicast distribution trees, showing the parent interface (from where the traffic is coming), and the children distribution interfaces (to which the traffic is being forwarded). It would also show any cache information available either in hardware forwarding mechanism or in the main processor (for software based forwarding).

**Note:** The cache information can be timed out when not enough traffic is present, but multicast routes can still be present. Cache information is presented in number of flows (Layer 4 sessions). Multicast routes stay at least for 5 minutes, while the hardware forwarding mechanism can time out a flow faster.
Any pruning information if present is also shown.

The search can always be narrowed by looking at a particular group, and/or looking at a particular parent interface, and/or looking at a particular child interface.
Multicast routes are not the same as DVMRP routes.

## Parameters

**child** <*ipaddr*> Address of a child interface.

**group** <*ipaddr*> Address of a multicast group.

**parent** <*ipaddr*> Address of a parent interface.

## Restrictions

None.

## Examples

To display the IP multicast route entry for the group 225.0.0.10:

```
gs/r# multicast show mroutes group 225.0.0.10
```

Here is a fuller example of the output from this command.

```
gs/r# multicast show mroutes
Network: 130.207.8/24 Group: 224.2.1.1 Age: 99s
Parent : mbone Child: test
downstream
Source : 130.207.8.82 Pkts: 383 Flows: 1

Network: 131.120.63/24 Group: 224.2.1.1 Age: 63s
Parent : mbone Pruned Child: test Pruned
downstream Pruned
Source : 131.120.63.33 Pkts: 0 Flows: 0

Network: 147.6.65.0/25 Group: 224.2.2.1 Age: 48s
Parent : mbone Pruned Child: test Pruned
downstream Pruned
Source : 147.6.65.38 Pkts: 0 Flows: 0
```

# Chapter 26

# mtrace Command

## Purpose

Trace multicast path between a source and a receiver

## Format

**mtrace** <*source*> **[destination** <*IPaddr*>**] [group** <*IPaddr*>**] [max-hops** <*number*>**]**

## Mode

User

## Description

The **mtrace** command tracks the multicast path from a source to a receiver. A trace probe is sent in a reverse path from the receiver back to the source. As the probe passes from hop to hop, it collects information such as interface address and packet counts from each router. If the **mtrace** command is executed with only the source parameter then a multicast path is calculated from the *source* to the GSR. One can examine the multicast path between two external hosts by specifying a receiver instead of using the GSR as the default receiver.

## Parameters

| | |
|---|---|
| <*source*> | IP address of the source. |
| **destination** <*IPaddr*> | Destination IP address. |

**group** <*IPaddr*> Multicast destination group address.

**max-hops** <*number*> Maximum number of hops to trace (default: 0, range: 0-32)

## Restrictions

None.

## Examples

To display the multicast path from IP address 2.2.2.2 to the GSR:

```
gs/r# mtrace 2.2.2.2
```

To display the multicast path from 1.1.1.1 to x.y.z.w for the group 239.1.1.1:

```
gs/r# mtrace 1.1.1.1 destination x.y.z.w group 239.1.1.1
```

# Chapter 27

# negate Command

The **negate** command negates a command in the scratchpad or the active configuration.

## Format

**negate** <*cmd-number*> **[scratchpad | active-config]**

## Mode

Configure

## Description

The **negate** command allows you to negate one or more commands by specifying the command number of the commands you want to negate. The command number for each command can be found using the Configure mode **show** command. You can negate commands from the active running system or non-committed commands from the scratchpad. By default, if you do not specify **active-config** or **scratchpad**, the command to negate is assumed to be in the **active-config**.

## Parameters

| | |
|---|---|
| <*cmd-number*> | The number of the command(s) you want to negate. Use the **show** command to display the command numbers. |
| **active-config** | Negate the specified command from the active running system. |
| **scratchpad** | Negate the specified non-committed command from the scratchpad. |

## Restrictions

The specified command number must represent a command that exists.

## Examples

To negate command 23 from the active configuration:

```
gs/r# negate 23
```

To negate commands 3, 5, 6 and 7 from the scratchpad:

```
gs/r# negate 3 5-7 scratchpad
```

# Chapter 28

# no Command

The **no** command removes a configuration command from the active configuration of the running system.

## Format

**no** <*command-to-negate*>

## Mode

Configure

## Description

The **no** command allows you to negate a previously executed command. Following the keyword **no**, one can specify the command to negate in its entirety or use the wildcard character (*****) to negate a group of commands. In addition to the **no** command, one can also use the **negate** command to negate a group of commands using the command number.

## Parameters

<*command*> The CLI command you want to negate. You do not have to enter the entire command. You can use the wildcard character, *, to negate matching commands. For example, if you specify “no acl 100 *” then all commands starting with the words “acl 100” will be negated.

## Restrictions

The command to negate must already be in the active configuration. You cannot negate a command that hasn't been entered.

## Examples

To negate the specified **arp add** command, enter the following. By negating this command, the system removes the ARP entry for *nfs2* from the ARP table.

```
gs/r# no arp add nfs2 macaddr 080020:13a09f exit-port et.3.1
```

To negate all commands starting with the word "acl":

```
gs/r# no acl *
```

# Chapter 29

# ntp Commands

The ntp commands configure and display the characteristics of the NTP (Network Time Protocol) client.

## Command Summary

Table 20 lists the ntp commands. The sections following the table describe the command syntax.

**Table 20. ntp commands**

| |
|---|
| **ntp set server** <*host*> **[interval** <*minutes*>**] [source** <*ipaddr*>**] [version** <*num*>**]** |
| **ntp show all** |
| **ntp synchronize server** <*host*> |

# ntp set server

## Purpose

Specifies the NTP server against which the GSR is to synchronize its clock.

## Format

**ntp set server** <*host*> **[interval** <*minutes*>**] [source** <*ipaddr*>**] [version** <*num*>**]**

## Mode

Configure

## Description

The **ntp set server** command instructs the GSR's NTP client to periodically synchronize its clock. By default, the GSR specifies an NTPv3 client that sends a synchronization packet to the server every 60 minutes. This means the GSR will attempt to set its own clock against the server once every hour. The synchronization interval as well as the NTP version number can be changed.

**Note:** To ensure that NTP has the correct time, you need to specify the time zone, as well. You can set the time zone by using the **system set timezone** command. When specifying daylight saving time, you'll need to use the **system set daylight-saving** command.

## Parameters

| | |
|---|---|
| **server** <*host*> | Specifies the hostname or the IP address of the NTP server. |
| **interval** <*minutes*> | Specifies how often (in minutes) the GSR should synchronize with the server. The default synchronization interval is 60 minutes. Valid interval is between 1 minute to 10080 minutes (7 days). |
| **source** <*ipaddr*> | Specifies the source IP address to be used by the GSR for sending the NTP packet. The IP address must belong to one of the interfaces on the GSR. |
| **version** <*num*> | Specifies the NTP version number of the packet. The default version number is 3 (NTPv3). Valid value is 1-3. |

## Restrictions

None.

## Examples

To send NTP packets to the NTP server 10.13.1.1 with default parameters:

```
gs/r(config)# ntp set server 10.13.1.1
```

To synchronize with a NTP server every 15 minutes with a specific source IP address:

```
gs/r(config)# ntp set server 10.13.1.1 interval 15 source 10.15.3.3
```

# ntp show all

## Purpose

Display NTP information about the GSR.

## Format

**ntp show all**

## Mode

Enable

## Description

The **ntp show all** command displays various NTP information about the GSR, for example, the last time a successful synchronization was made, synchronization interval, NTP version number, etc.

## Parameters

None.

## Restrictions

None.

## Example

```
gs/r# ntp show all
NTP status:
     Synchronization interval: 60 mins
     Version: NTPv3
     Last successful contact: Thu Jul 23 23:08:15 1998
```

# ntp synchronize server

## Purpose

Manually force the GSR to immediately synchronize with a NTP server.

## Format

**ntp synchronize server** <*host*>

## Mode

Enable

## Description

The **ntp synchronize server** command forces the GSR to immediately synchronize its clock with the NTP server. Unlike the Configuration mode **ntp set server** command, this Enable mode command does not send periodic synchronization packets to the server. Instead, each time this command is executed, the GSR synchronizes itself with the server. To have the GSR synchronizes itself periodically, use the **ntp set server** command.

## Parameters

| | |
|---|---|
| <*host*> | Specifies the hostname or the IP address of the NTP server. |

## Restrictions

None.

## Examples

To synchronize the GSR against the NTP server 10.13.1.1:

```
gs/r(config)# ntp synchronize server 10.13.1.1
%NTP-I-TIMESYNC  Time synchronized to Thu Jul 23 23:11:28 1998
```

# Chapter 30

# ospf Commands

The ospf commands let you display and set parameters for the Open Shortest Path First (OSPF) routing protocol.

## Command Summary

Table 21 lists the ospf commands. The sections following the table describe the command syntax.

**Table 21. ospf commands**

| |
|---|
| **ospf add interface** <*interfacename-or-IPaddr*> **[to-area** <*area-addr*> **\| backbone]** **[type broadcast \| non-broadcast]** |
| **ospf add nbma-neighbor** <*hostname-or-IPaddr*> **to-interface** <*hostname-or-IPaddr*> **[eligible]** |
| **ospf add stub-host [to-area** <*area-addr*> **\| backbone] [cost** <*num*>**]** |
| **ospf add network \| summary-range** |
| **ospf add virtual-link** <*number-or-string*> **[neighbor** <*IPaddr*>**]** **[transit-area** <*area-num*>**]** |
| **ospf create area** <*area-num*> **[backbone]** |
| **ospf create-monitor destination** <*hostname-or-IPaddr*> |
| **ospf monitor** <*option-list*> |
| **ospf set area** <*area-num*> **[stub] [stub-cost** <*num*>**]** **[authentication-method none \| simple \| md5]** |
| **ospf set ase-defaults [preference** <*num*>**] [cost** <*num*>**]** **[type** <*num*>**] [inherit-metric]** |

**Table 21. ospf commands (Continued)**

| |
|---|
| **ospf set export-interval** <*num*> |
| **ospf set export-limit** <*num*> |
| **ospf set interface** <*interfacename-or-IPaddr*> **\| all**<br>**[state disable \| enable] [cost** <*num*>**] [no-multicast] [retransmit-interval** <*num*>**]**<br>**[transit-delay** <*num*>**] [priority** <*num*>**] [hello-interval** <*num*>**]**<br>**[router-dead-interval** <*num*>**] [poll-interval** <*num*>**][key-chain** <*num-or-string*>**]** |
| **ospf set monitor-auth-method none \| simple \| md5** |
| **ospf set trace-options [lsa-build] [spf] [lsa-transmit] [lsa-receive] [state] [hello] [dd]**<br>**[request] [lsu] [ack]** |
| **ospf set virtual-link** <*number-or-string*> **[state disable \| enable] [cost** <*num*>**]**<br>**[no-multicast] [retransmit-interval** <*num*>**] [transit-delay** <*num*>**] [priority**<br><*num*>**] [hello-interval** <*num*>**] [router-dead-interval** <*num*>**] [poll-interval**<br><*num*>**]** |
| **ospf show** <*option-list*> |
| **ospf start \| stop** |

# ospf add interface

## Purpose

Associates an interface with an OSPF area.

## Format

**ospf add interface** <*interfacename-or-IPaddr*> **[to-area** <*area-addr*> **| backbone] [type broadcast | non-broadcast]**

## Mode

Configure

## Parameters

<*interfacename-or-IPaddr*>
An interface name or an IP address.

**to-area** <*area-addr*> **| backbone**
OSPF Area with which this interface is to be associated.

**type**
Specifies whether the interface is broadcast or non-broadcast. Specify one of the following:

- **broadcast** (default)
- **non-broadcast**

## Restrictions

None.

# ospf add nbma-neighbor

## Purpose

Specifies an OSPF NBMA Neighbor.

## Format

**ospf add nbma-neighbor** <*hostname-or-IPaddr*> **to-interface** <*interfacename-or-IPaddr*> **[eligible]**

## Mode

Configure

## Parameters

**to-interface** <*interfacename-or-IPaddr*>
Adds the neighbor to the specified OSPF interface.

**eligible**
Specifies whether an OSPF NBMA Neighbor is eligible for becoming a designated router.

## Restrictions

None.

# ospf add network | summary-range

**Note:** Because the **OSPF add network** command is misinterpreted with commands having similar syntax from other vendors, this command will eventually be dropped from the GSR's host of CLI commands. The new command is **ospf add summary-range**. At this time, however, both are acceptable CLI commands, hence both are dealt with in this section.

## Purpose

Configures summary-ranges on Area Border Routers (ABRs). This allows you to reduce the amount of routing information propagated between areas.

On the GSR, summary-ranges are created using the **ospf add summary-range** command – the networks specified using this command describe the scope of an area. Intra-area Link State Advertisements (LSAs) that fall within the specified ranges are not advertised into other areas as inter-area routes. Instead, the specified ranges/networks are advertised as summary network LSAs. If you specify the **restrict** option, the summary network LSAs are not advertised. Each intra-area LSA that does not fall into any range is advertised as an OSPF Type-3 or 4 LSA.

## Format

**ospf add network | summary-range** <*IPaddr/mask*> **[to-area** <*area-addr*>**] [restrict] [host-net]**

## Mode

Configure

## Parameters

<*IPaddr/mask*>
IP Address and network mask value representing the summary-range. Example: 16.122.0.0/255.255.0.0 or 16.122.0.0/16.

**to-area** <*area-addr*>
OSPF Area with which this summary-range is to be associated.

**restrict**
If the restrict option is specified for a network/summary-range, then that network is not advertised in Summary network LSAs.

**host-net**
Specifies that the network is an OSPF Host Network.

## Restrictions

None.

## Example

In the following example, two summary ranges are created:

```
ospf add summary-range 207.135.16.0/24 to-area 207.135.0.0
ospf add summary-range 207.135.17.0/24 to-area 207.135.0.0 restrict
```

Intra-area Link State Advertisements (LSAs) that fall within the range 207.135.16.0/24 are not advertised into other areas as inter-area routes. Instead, the specified range 207.135.16.0/24 is advertised as summary network LSA.

Because the summary range 207.135.17.0/24 has the restrict option associated with it, intra-area link state advertisements (LSAs) that fall within it are not advertised as summary network LSA. Using this mechanism, one can have “hidden networks” within an area, which are not advertised to other areas.

# ospf add stub-host

## Purpose

Adds a stub-host to an OSPF area.

## Format

**ospf add stub-host** <*hostname-or-IPaddr*> **[to-area** <*area-addr*> **| backbone] [cost** <*num*>**]**

## Mode

Configure

## Parameters

**to-area** <*area-addr*> **| backbone**
OSPF Area to which you are adding a stub host.

**cost** <*num*>
The cost that should be advertised for this directly attached stub host. Specify a number from 0 – 65535.

## Restrictions

None.

# ospf add virtual-link

## Purpose

Creates an OSPF Virtual Link.

## Format

**ospf add virtual-link** <*number-or-string*> **[neighbor** <*IPaddr*>**] [transit-area** <*area-num*>**]**

## Mode

Configure

## Parameters

<*number-or-string*>
A number or character string identifying the virtual link.

**neighbor** <*IPaddr*>
The IP address of an OSPF virtual link neighbor.

**transit-area** <*area-num*>
The Area ID of the transit area.

## Restrictions

None.

# ospf create area

## Purpose

Create an OSPF area.

## Format

**ospf create area** <*area-num*> | **backbone**

## Mode

Configure

## Parameters

| | |
|---|---|
| <*area-num*> | The Area ID. Normally, Area IDs are formatted like IP addresses: <*num*>.<*num*>.<*num*>.<*num*>. |
| **backbone** | Specifies that the Area you are adding is the backbone area. |

## Restrictions

None.

# ospf create-monitor

## Purpose

Create an OSPF monitor destination.

## Format

**ospf create-monitor destination** <*hostname-or-IPaddr*>

## Mode

Enable

## Parameters

**destination** <*hostname-or-IPaddr*>
Specifies the destination whose OSPF activity is to be monitored.

## Restrictions

None.

# ospf monitor

## Purpose

Monitor OSPF.

## Format

**ospf monitor statistics | errors | next-hop-list | interfaces | neighbors [destination** <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

**ospf monitor lsdb [display-retransmit-list] [destination** <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

**ospf monitor routes [type all | asbrs-in-area | area-border-routers | asbrs-other-areas | networks-in-area | networks-other-areas | as-routes] [destination** <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

**ospf monitor lsa area-id** <*IPaddr*> **type router-links | network-links | summary-networks | summary-asbr | as-external ls-id** <*IPaddr*> **adv-rtr** <*IPaddr*> **[destination** <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

**ospf monitor as-external-db [display-retransmit-list destination** <*IPaddr*>**] [auth-key** <*string*>**]**

## Mode

Enable

## Parameters

**destination** <*hostname-or-IPaddr*>
Monitors the specified OSPF destination. Default is the router on which the command is executed.

**auth-key** <*string*>
Specifies the authorization key for the OSPF destination. This option is not needed if the OSPF destination does not require a key or if an authorization was specified using the **ospf monitor create-destination** command.

**statistics**
Shows input/output statistics for monitor request, hello, data base description, link-state request, link-state update, and link-state ack packets. Area statistics are

provided, which describe the total number of routing neighbors and number of active OSPF interfaces. Routing table statistics are summarized and reported as the number of intra-area routes, inter-area routes, and AS external data base entries.

**errors**
Shows the various error conditions which can occur between OSPF routing neighbors and the number of occurrences for each.

**next-hop-list**
Shows information about all valid next hops mostly derived from the SPF calculation.

**interfaces**
Shows information about all interfaces configured for OSPF. Information reported includes the area, interface IP address, interface type, interface state, cost, priority, and the IP address of the Designated Router and Backup Designated Router for the network.

**neighbors**
Shows information about all OSPF routing neighbors. Information reported includes the area, local interface address, router ID, neighbor IP address, state, and mode.

**lsdb**
Displays the link-state database (except for ASEs). This table describes the routers and networks making up the AS. If the display-retransmit-list option is specified, the retransmit list of neighbors held by this lsdb structure will also be printed.

**display-retransmit-list** – Displays the retransmit list from the link state database.

**routes**
Displays the OSPF routing table. This table reports the AS border routes, area border routes, summary AS border routes, networks, summary networks and AS external networks currently managed via OSPF.

**type all**
Shows all OSPF routes.

**type asbrs-in-area**
Shows routes to AS boundary routers in this area.

**type area-border-routers**
Shows routes to area border routers for this area.

**type asbrs-other-areas**
Shows summary routes to AS boundary routers in other areas.

**type networks-in-area**
Shows routes to networks in this area.

**type networks-other-areas**
Shows routes to networks in other areas.

**type as-routes**
Shows AS routes to non-OSPF networks.

**lsa**
Displays the link state advertisement. Area_Id is the OSPF area for which the query is directed. Adv_Rtr is the router -id of the router which originated this link state advertisement. Type specifies the type of advertisement to request:

**area-id** <*IPaddr*>
Specifies the OSPF area.

**type router-links**
Requests router link advertisements that describe the collected states of the router interfaces. ls-id is set to the originating router's router-id.

**type network-links**
Requests network link advertisements that describe the set of routers attached to the network. ls-id is set to the IP interface address of the designated router for the network.

**type summary-networks**
Request summary-link advertisements describing routes to networks. ls-id is set to the IP address of the destination network.

**type summary-asbr**
Requests summary-link advertisements describing routes to AS boundary routers. ls-id is set to the AS boundary router's router-id.

**type as-external**
Requests AS external link state advertisements. ls-id is set to the IP address of the destination network.

**ls-id** <*IPaddr*>
Species the ls-id for the type of link-state advertisement requested

**adv-rtr** <*IPaddr*>
Requests the router ID of the originating router.

**as-external-db**
Display the AS external data base entries. This table reports the advertising router, forwarding address, age, length, sequence number, type, and metric for each AS external route. If the display-retransmit-list option is specified, the retransmit list of neighbors held by this lsdb structure will also be printed.

## Restrictions

None.

## Examples

The following are examples of ospf monitor commands.

```
gs/r# ospf monitor statistics

IO stats
Input  Output  Type
8       0  Monitor request
1322    1314  Hello
716     721  DB Description
39     728  Link-State Req
3037    3355  Link-State Update
1317     354  Link-State Ack
ASE: 1903 checksum sum 3BB0F22

LSAs originated: 1915   received: 17
Router: 5  ASE: 1910

Area 0.0.0.0:
Neighbors: 3    Interfaces: 3
Spf: 3  Checksum sum 6CB41
DB: rtr: 5 net: 5 sumasb: 0 sumnet: 2

Routing Table:
Intra Area: 5   Inter Area: 4   ASE: 1
```

```
gs/r# ospf monitor errors

Packets Received:
10: Monitor request                   1342: Hello
716: DB Description                      39: Link-State Req
3212: Link-State Update                1536: Link-State Ack

Packets Sent:
0: Monitor response                   1335: Hello
721: DB Description                      728: Link-State Req
3907: Link-State Update                  359: Link-State Ack

Errors:
0: IP: bad destination                   0: IP: bad protocol
0: IP: received my own packet            0: OSPF: bad packet type
0: OSPF: bad version                     0: OSPF: bad checksum
0: OSPF: bad area id                     0: OSPF: area mismatch
0: OSPF: bad virtual link                0: OSPF: bad authentication type
0: OSPF: bad authentication key          0: OSPF: packet too small
0: OSPF: packet size > ip length         1: OSPF: transmit error
0: OSPF: interface down                  0: OSPF: unknown neighbor
0: HELLO: netmask mismatch               0: HELLO: hello timer mismatch
0: HELLO: dead timer mismatch            0: HELLO: extern option mismatch
0: HELLO: router id confusion            0: HELLO: virtual neighbor
unknown
0: HELLO: NBMA neighbor unknown          0: DD: neighbor state low
0: DD: router id confusion               0: DD: extern option mismatch
```

```
0: DD: unknown LSA type                1: LS ACK: neighbor state low
0: LS ACK: bad ack                  1140: LS ACK: duplicate ack
0: LS ACK: Unknown LSA type            0: LS REQ: neighbor state low
0: LS REQ: empty request               0: LS REQ: bad request
8: LS UPD: neighbor state low          0: LS UPD: newer self-gen LSA
0: LS UPD: LSA checksum bad          131: LS UPD: received less recent
LSA
0: LS UPD: unknown LSA type            2: Interface: Not configed for
OSPF
0: Interface: Invalid type             0: Interface: Mcast disabled.
0: Interface: Invalid state            0: Interface: Address not found
1: No vlinks and src is non local
```

```
gs/r# ospf monitor next-hop-list

Next hops:

Address          Type      Refcount  Interface
-------------------------------------------------------------
10.12.1.1        Neighbor         6  10.12.1.2       to-c4500
10.12.1.2        Direct           1  10.12.1.2       to-c4500
150.1.0.1        Direct           1  150.1.0.1       to-aval-eth5
172.23.1.5       Direct           3  172.23.1.5      to-GSR6
172.23.1.6       Neighbor         5  172.23.1.5      to-GSR6
172.23.1.21      Direct           3  172.23.1.21     to-GSR1
172.23.1.22      Neighbor        19  172.23.1.21     to-GSR1
172.23.1.25      Direct           3  172.23.1.25     lo
222.1.1.1        Direct           1  222.1.1.1       to-linux1
```

```
gs/r# ospf monitor interfaces
>sent to 127.0.0.1

Source <<127.0.0.1       >>

Area: 0.0.0.0
IP Address      Type  State    Cost Pri DR             BDR
-------------------------------------------------------------------------
172.23.1.5      Bcast BackupDR 2    2   172.23.1.6      172.23.1.5
10.12.1.2       Bcast BackupDR 1    2   10.12.1.1       10.12.1.2
172.23.1.21     Bcast BackupDR 1    2   172.23.1.22     172.23.1.21
done
```

```
gs/r# ospf monitor neighbors
> sent to 127.0.0.1

Source <<127.0.0.1       >>
```

```
Interface: 172.23.1.5      Area: 0.0.0.0
Router Id       Nbr IP Addr     State      Mode   Prio
-----------------------------------------------------
0.0.0.6         172.23.1.6      Full       Slave  1

Interface: 10.12.1.2       Area: 0.0.0.0
Router Id       Nbr IP Addr     State      Mode   Prio
-----------------------------------------------------
172.23.1.14     10.12.1.1       Full       Slave  1

Interface: 172.23.1.21     Area: 0.0.0.0
Router Id       Nbr IP Addr     State      Mode   Prio
-----------------------------------------------------
0.0.0.1         172.23.1.22     Full       Master 1
done
```

```
gs/r# ospf monitor routes
> sent to 127.0.0.1

Source <<127.0.0.1       >>
AS Border Routes:
Router          Cost AdvRouter       NextHop(s)
-------------------------------------------------
Area 0.0.0.0:
0.0.0.6            2 0.0.0.6         172.23.1.6
172.23.1.22
0.0.0.4            0 0.0.0.4
0.0.0.1            1 0.0.0.1         172.23.1.22

Total AS Border routes: 3

Area Border Routes:
Router          Cost AdvRouter       NextHop(s)
-------------------------------------------------
Area 0.0.0.0:
0.0.0.3            2 0.0.0.3         172.23.1.22
0.0.0.1            1 0.0.0.1         172.23.1.22

Total Area Border Routes: 2

Summary AS Border Routes:
Router          Cost AdvRouter       NextHop(s)
-------------------------------------------------

Networks:
Destination        Area            Cost Type NextHop         AdvRouter
-------------------------------------------------------------------------

172.23.1.4/30      0.0.0.0            2 Net  172.23.1.5      0.0.0.6
10.12.1.0/30       0.0.0.0            1 Net  10.12.1.1       172.23.1.14
172.23.1.20/30     0.0.0.0            1 Net  172.23.1.21     0.0.0.1
172.23.1.25        0.0.0.0            0 Stub 172.23.1.25     0.0.0.4
172.23.1.8/30      0.0.0.0            2 Net  172.23.1.22     0.0.0.1
10.12.1.4/30       0.0.0.0            2 Net  172.23.1.22     172.23.1.14
172.23.1.14        0.0.0.0            2 Stub 10.12.1.1       172.23.1.14
```

```
172.23.1.26        0.0.0.0            3 Stub 172.23.1.6      0.0.0.6
172.23.1.22
16                 0.0.0.0            2 SNet 172.23.1.22     0.0.0.1
ASEs:
Destination        Cost E      Tag NextHop         AdvRouter
------------------------------------------------------------------------------
15.1                 1   1 c0000000 172.23.1.22     0.0.0.1
Total nets: 9
Intra Area: 5   Inter Area: 4   ASE: 1
done
```

```
gs/r# ospf monitor lsdb

LS Data Base:
Area: 0.0.0.0
Type LinkState ID    AdvRouter        Age Len Sequence Metric Where
-------------------------------------------------------------------
Stub 172.23.1.25     0.0.0.4          341 24  0             0 SpfTree
Stub 172.23.1.14     172.23.1.14      352 24  0             0 SpfTree
Stub 172.23.1.26     0.0.0.6          343 24  0             0 SpfTree
Rtr  0.0.0.1         0.0.0.1          309 72  800009b0      0 SpfTree
Rtr  0.0.0.3         0.0.0.3         1223 36  80000011      0 SpfTree
Rtr  0.0.0.4         0.0.0.4          341 72  80000084      0 SpfTree
Rtr  172.23.1.14     172.23.1.14       74 60  80000bf6      0 Clist
Rtr  0.0.0.6         0.0.0.6          227 60  80000a0d      0 SpfTree
Net  172.23.1.10     0.0.0.1          309 32  80000005      0 SpfTree
Net  172.23.1.22     0.0.0.1          309 32  80000003      0 SpfTree
Net  10.12.1.1       172.23.1.14       74 32  80000002      0 SpfTree
Net  10.12.1.6       172.23.1.14       74 32  8000003d      0 SpfTree
Net  172.23.1.6      0.0.0.6          227 32  80000003      0 SpfTree
SNet 16.255.255.255  0.0.0.3         1129 28  8000000c      1 Uninitialized
SNet 16.255.255.255  0.0.0.1          215 28  80000003      1 Uninitialized
done
```

```
gs/r# ospf monitor as-external-db

AS External Data Base:
Destination        AdvRouter       Forward Addr     Age Len Sequence T Metric
------------------------------------------------------------------------------
130.58.225         0.0.0.4         0.0.0.0          201 36  80000001 21
130.58.174         0.0.0.4         0.0.0.0          201 36  80000001 21
130.56.235         0.0.0.4         0.0.0.0          236 36  80000001 21
130.56.184         0.0.0.4         0.0.0.0          236 36  80000001 21
130.54.245         0.0.0.4         0.0.0.0          238 36  80000001 21
130.54.194         0.0.0.4         0.0.0.0          239 36  80000001 21
130.52.255         0.0.0.4         0.0.0.0          241 36  80000001 21
130.52.204         0.0.0.4         0.0.0.0          241 36  80000001 21
130.51.9           0.0.0.4         0.0.0.0          211 36  80000001 21
130.50.214         0.0.0.4         0.0.0.0          211 36  80000001 21
130.49.19          0.0.0.4         0.0.0.0          213 36  80000001 21
130.48.224         0.0.0.4         0.0.0.0          214 36  80000001 21
```

```
130.47.29       0.0.0.4         0.0.0.0         216 36  80000001 21
130.46.234      0.0.0.4         0.0.0.0         248 36  80000001 21
130.45.39       0.0.0.4         0.0.0.0         251 36  80000001 21
130.44.244      0.0.0.4         0.0.0.0         251 36  80000001 21
130.43.49       0.0.0.4         0.0.0.0         253 36  80000001 21
130.42.254      0.0.0.4         0.0.0.0         221 36  80000001 21
130.41.59       0.0.0.4         0.0.0.0         256 36  80000001 21
130.41.8        0.0.0.4         0.0.0.0         256 36  80000001 21
130.39.69       0.0.0.4         0.0.0.0         258 36  80000001 21
130.39.18       0.0.0.4         0.0.0.0         258 36  80000001 21
130.37.79       0.0.0.4         0.0.0.0         261 36  80000001 21
130.37.28       0.0.0.4         0.0.0.0         261 36  80000001 21
130.35.89       0.0.0.4         0.0.0.0         263 36  80000001 21
130.35.38       0.0.0.4         0.0.0.0         263 36  80000001 21
130.33.99       0.0.0.4         0.0.0.0         267 36  80000001 21
130.33.48       0.0.0.4         0.0.0.0         267 36  80000001 21
130.31.109      0.0.0.4         0.0.0.0         272 36  80000001 21
130.31.58       0.0.0.4         0.0.0.0         272 36  80000001 21
130.29.119      0.0.0.4         0.0.0.0         277 36  80000001 21
130.29.68       0.0.0.4         0.0.0.0         277 36  80000001 21
130.27.129      0.0.0.4         0.0.0.0         282 36  80000001 21
130.27.78       0.0.0.4         0.0.0.0         282 36  80000001 21
130.25.139      0.0.0.4         0.0.0.0         287 36  80000001 21
130.25.88       0.0.0.4         0.0.0.0         287 36  80000001 21
130.23.149      0.0.0.4         0.0.0.0         292 36  80000001 21
130.23.98       0.0.0.4         0.0.0.0         292 36  80000001 21
130.21.159      0.0.0.4         0.0.0.0         297 36  80000001 21
```

# ospf set area

## Purpose

Sets the parameters for an OSPF area.

## Format

**ospf set area** <*area-num*> **[stub] [stub-cost** <*num*>**] [authentication-method none | simple | md5]**

## Mode

Configure

## Parameters

<*area-num*>
The Area ID.

**stub**
Makes this Area a stub area.

**stub-cost** <*num*>
Specifies the cost to be used to inject a default route into the area. Specify a number from 0 – 65535.

**authentication-method none | simple | md5**
Specifies the authentication method used within the area. Specify one of the following:

**none** Does not use authentication.

**simple** Uses a simple string (password) up to 8 characters in length for authentication. If you chose this authentication method, then you should also specify a key-chain identifier using the key-chain option.

**md5** Uses the MD5 algorithm to create a crypto-checksum of an OSPF packet and an authentication key of up to 16 characters.

## Restrictions

None.

# ospf set ase-defaults

## Purpose

Sets the defaults used when importing OSPF ASE routes into the routing table and exporting routes from the routing table into OSPF ASEs.

## Format

**ospf set ase-defaults [preference** <*num*>**] [cost** <*num*>**] [type** <*num*>**] [inherit-metric]**

## Mode

Configure

## Parameters

**preference** <*num*>
Specifies the preference of OSPF ASE routes. Specify a number between 0 and 255.

**cost** <*num*>
Specifies the cost used when exporting non-OSPF route into OSPF as an ASE. Specify a number from 0 – 65535.

**type** <*num*>
Specifies the ASE type. Routes exported from the routing table into OSPF default to becoming type 1 ASEs. You can change the default using the **type** option. You also can override the type in OSPF export policies. Specify either 1 or 2.

**inherit-metric**
Allows an OSPF ASE route to inherit the metric of the external route when no metric is specified on the export. A metric specified with the export command takes precedence. The cost specified in the default is used if you do not specify **inherit-metric**.

## Restrictions

None.

# ospf set export-interval

## Purpose

Specifies the interval at which ASE LSAs will be generated and flooded into OSPF. The default is once per second.

## Format

**ospf set export-interval** <*num*>

## Mode

Configure

## Parameters

<*num*> The interval in seconds. Specify a number equal to or greater than 1. The default is 1 (once per second).

## Restrictions

None.

# ospf set export-limit

## Purpose

Specifies how many ASEs will be generated and flooded in each batch.

## Format

**ospf set export-limit** <*num*>

## Mode

Configure

## Parameters

<*num*> The export limit. Specify a number equal to or greater than 1. The default is 100.

## Restrictions

None.

# ospf set interface

## Purpose

Sets parameters for an OSPF interface.

## Format

**ospf set interface** <*name-or-IPaddr*> | **all**
**[state disable | enable] [cost** <*num*>**] [no-multicast]**
**[retransmit-interval** <*num*>**] [transit-delay** <*num*>**]**
**[priority** <*num*>**] [hello-interval** <*num*>**]**
**[router-dead-interval** <*num*>**] [poll-interval** <*num*>**]**
**[key-chain** <*num-or-string*>**]**

## Mode

Configure

## Parameters

<*name-or-IPaddr*> | **all**
The OSPF interface for which you are setting OSPF parameters.

**state disable | enable**
Enables or disables OSPF on the interface.

**cost** <*num*>
The cost associated with this interface. The cost of all interfaces that a packet must cross to reach a destination are added to get the cost to that destination. The default cost of the OSPF interface is 1, but another non-zero value may be specified. Specify a number from 0 – 65535.

**no-multicast**
Instructs the GSR not to send multicast packets to neighbors on point-to-point interfaces.

**retransmit-interval** <*num*>
The number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface. Specify a number equal to or greater than 1. The default is 5.

**transit-delay** <*num*>
The estimated number of seconds required to transmit a link state update over this interface. Transit delay takes into account transmission and propagation delays and must be greater than 0. Specify a number equal to or greater than 1. The default is 1.

**priority** <*num*>
A number between 0 and 255 specifying the priority for becoming the designated router on this interface. When two routers attached to a network both attempt to become the designated router, the one with the higher priority wins. A router whose router priority is set to 0 is ineligible to become designated router. Specify a number from 0 – 255. The default is 0.

**hello-interval** <*num*>
The length of time, in seconds, between hello packets that the router sends on this interface. Specify a number from 0 – 255. The default is 10 for broadcast interfaces and 30 for point-to-point and other non-broadcast interfaces.

**router-dead-interval** <*num*>
The number of seconds not hearing a router's Hello packets before the router's neighbors will declare it down. Specify a number from 0 – 255. The default is 4 times the value of the hello interval.

**poll-interval** <*num*>
Before adjacency is established with a neighbor, OSPF packets are sent periodically at the specified poll interval. Specify a number equal to or greater than 1. The default value for this option is 120 seconds.

**key-chain** <*num-or-string*>
The identifier of the key-chain containing the authentication keys.

## Restrictions

None.

# ospf set monitor-auth-method

## Purpose

You can query the OSPF state using the OSPF-Monitor utility. This utility sends non-standard OSPF packets that generate a text response from OSPF. By default these requests are not authenticated. If you specify an authentication key, the incoming requests must match the specified authentication key.

## Format

**ospf set monitor-auth-method none | simple | md5**

## Mode

Configure

## Description

This section contains a fuller description of what the command does.

## Parameters

**authentication-method none | simple | md5**

The authentication method used within the area. Specify one of the following:

**none** Does not use authentication.

**simple** Uses a simple string (password) up to 16 characters in length for authentication. If you chose this authentication method, then you should also specify a key-chain identifier using the key-chain option.

**md5** Uses the MD5 algorithm to create a crypto-checksum of an OSPF packet and an authentication key of up to 16 characters.

## Restrictions

None.

# ospf set trace-options

## Purpose

Sets various OSPF trace options.

## Format

**ospf set trace-options lsa-build | spf | lsa-transmit | lsa-receive**

**ospf set trace-options hello | dd | request | lsu | ack [detail] [send] [receive]**

## Mode

Configure

## Parameters

| | |
|---|---|
| **lsa-build** | Traces Link State Advertisement Creation. |
| **spf** | Traces Shortest Path First (SPF) calculations. |
| **lsa-transmit** | Traces Link State Advertisement (LSA) transmission. |
| **lsa-receive** | Traces Link State Advertisement (LSA) reception. |
| **hello** | Traces OSPF hello packets that are used to determine neighbor reachability. |
| **dd** | Traces OSPF Database Description packets that are used in synchronizing OSPF databases. |
| **request** | Traces OSPF Link State Request packets which are used in synchronizing OSPF databases. |
| **lsu** | Traces OSPF Link State Update packets which are used in synchronizing OSPF databases. |
| **ack** | Traces OSPF Link State Ack packets which are used in synchronizing OSPF databases. |
| **detail** | Shows detailed information about OSPF packets. |

| | |
|---|---|
| **send** | Shows OSPF packets sent by the router. |
| **receive** | Shows OSPF packets received by the router. |

## Restrictions

None.

# ospf set virtual-link

## Purpose

Sets the parameters for an OSPF virtual link.

## Format

**ospf set virtual-link** <*number-or-string*>
**[state disable | enable] [cost** <*num*>**] [no-multicast] [retransmit-interval** <*num*>**]**
**[transit-delay** <*num*>**] [priority** <*num*>**] [hello-interval** <*num*>**]**
**[router-dead-interval** <*num*>**] [poll-interval** <*num*>**]**

## Mode

Configure

## Parameters

<*number-or-string*>
The identifier for this virtual link.

**state disable | enable**
Enables or disables the virtual link.

**cost** <*num*>
The cost associated with this virtual link. The cost of all interfaces that a packet must cross to reach a destination are added to get the cost to that destination. The default cost of the OSPF interface is 1, but another non-zero value may be specified. Specify a number from 0 – 65535.

**no-multicast**
Instructs the GSR to not send multicast packets to neighbors on point-to-point virtual links.

**retransmit-interval** <*num*>
The number of seconds between link state advertisement retransmissions for adjacencies belonging to this virtual link. Specify a number equal to or greater than 1.

**transit-delay** <*num*>
The estimated number of seconds required to transmit a link state update over this virtual link. Transit delay takes into account transmission and propagation delays and must be greater than 0. Specify a number equal to or greater than 1.

**priority** <*num*>
A number between 0 and 255 specifying the priority for becoming the designated router on this virtual link. When two routers attached to a network both attempt to become the designated router, the one with the higher priority wins. A router whose router priority is set to 0 is ineligible to become designated router. Specify a number from 0 – 255.

**hello-interval** <*num*>
The length of time, in seconds, between hello packets that the router sends on this virtual link. Specify a number from 0 – 255. The default is 60 seconds.

**router-dead-interval** <*num*>
The number of seconds not hearing a router's Hello packets before the router's neighbors will declare it down. Specify a number from 0 – 255. The default value for this parameter is 4 times the value of the **hello-interval** parameter

**poll-interval** <*num*>
Before adjacency is established with a neighbor, OSPF packets are sent periodically at the specified poll interval. Specify a number from 0 – 255. The default is 120 seconds.

## Restrictions

None.

# ospf show

## Purpose

Show OSPF information.

## Format

**ospf show** <*option-list*>

## Mode

Enable

## Parameters

<*option-list*>
Specifies the OSPF information you want to display. Specify one or more of the following:

| | |
|---|---|
| **all** | Displays all OSPF tables. |
| **globals** | Displays OSPF globals. |
| **timers** | Displays OSPF timers. |
| **areas** | Displays OSPF areas. |
| **interfaces** | Displays OSPF interfaces. |
| **next-hop-list** | Displays valid next hop entries. |
| **import-policies** | Displays OSPF import policies. |
| **export-policies** | Displays OSPF export policies. |
| **statistics** | Displays OSPF statistics. |
| **errors** | Displays OSPF errors. |
| **virtual-links** | Displays OSPF virtual links. |
| **summary-asb** | Displays OSPF border routes. |
| **AS-external-LDSB** | Displays OSPF Autonomous System external link states. |
| **exported-routes** | Displays routes redistributed into OSPF. |

**Note:** The **areas**, **virtual-links**, **summary-asb**, **AS-external-LDSB**, and **exported-routes** options can be used with the following display options:

**to file** Saves output in the file `/gatedtrc/gated.dmp`.

**to terminal** Displays output on the console. This is the default.

# ospf start|stop

## Purpose

Start or stop the OSPF protocol. OSPF is disabled by default on the GSR.

## Format

**ospf start | stop**

## Mode

Configure

## Parameters

**start** Starts OSPF.

**stop** Stops OSPF.

## Restrictions

None.

# Chapter 31

# ping Command

The **ping** command tests connection between the GSR and an IP host.

## Format

**ping** <*hostname-or-IPaddr*> **packets** <*num*> **size** <*num*> **wait** <*num*> **[flood] [dontroute]**

## Mode

User or Enable

## Description

The **ping** command test connection between the GSR and an IP host. The ping command sends ICMP echo packets to the host you specify.

- If the packets reach the host, the host sends a ping response to the GSR and the CLI displays messages stating that the host can be reached.
- If the host does not respond, the GSR assumes the host cannot be reached from the GSR and the CLI display messages stating that the host did not reply.

## Parameters

*<hostname-or-IPaddr>*
The host name or IP address you want to ping.

**packets** <*num*>
The number of ping packets you want to send. The default is 1.

**size** <*num*>
The packet size. For Ethernet, specify a number from 0 – 1364.

**wait** <*num*>
The number of seconds the GSR will wait for a positive response from the host before assuming that the host has not responded. The default is 1.

**flood**
Causes the GSR to send a new ping request as soon as a ping reply is received. If you do not specify the **flood** option, the GSR waits to send a new request. The amount of time the GSR waits is specified by the **wait** option.

**dontroute**
Restricts the ping to locally attached hosts.

## Restrictions

If you enter this command from the User mode, the only parameter you can use is <*hostname-or-IPaddr*>. To use any of the other parameters, you must be in Enable mode.

# Chapter 32

# port Commands

The port commands set and display the following parameters:

- Port state (enabled or disabled)
- Bridging status (flow-based or address-based)
- Port operating mode (half duplex or full duplex)
- Port speed for the 10/100 ports (10-Mbps or 100-Mbps)
- Port mirroring (used for analyzing network traffic)

## Command Summary

Table 22 lists the port commands. The sections following the table describe the command syntax.

**Table 22. port commands**

| |
|---|
| **port disable** <*port-list*> |
| **port flow-bridging** <*port-list*> \| **all-ports** |
| **port mirroring to** <*port*> **cpu-port-traffic** \| **traffic-from [**<*port*> \| **any] traffic-to [**<*slot*> \| **any]** |
| **port set [**<*port-list*> \| **all-ports] [duplex full** \| **half] [speed 10Mbps** \| **100Mbps** \| <*number*>**] [auto-negotiation on** \| **off] [hash-mode m0** \| **m1** \| **m2** \| **m3] [wan encapsulation frame-relay** \| **ppp]** |
| **port show bridging-status** <*port-list*> \| **all-ports** |
| **port show port-status** <*port-list*> \| **all-ports** |

**Table 22. port commands (Continued)**

| |
|---|
| **port show stp-info** <*port-list*> \| **all-ports** |
| **port show vlan-info** <*port-list*> \| **all-ports** |
| **port show mirroring-status** <*slot*> \| **all-slots** |

# port disable

## Purpose

Disable a port.

## Format

**port disable** <*port-list*>

## Mode

Configure

## Description

The **port disable** command disables the specified ports. Disabled ports do not send or receive any traffic. You might want to disable unused ports to prevent network users from inadvertently or unscrupulously connecting to unoccupied but enabled ports on the GSR.

## Parameters

**port** <*port-list*> Specifies the ports you are disabling.

## Restrictions

None.

## Examples

To disable port et.1.3 on the GSR:

```
gs/r(config)# port disable et.1.3
```

To disable ports 1 through 5 on the Ethernet line card in slot 3 of the GSR chassis:

```
gs/r(config)# port disable et.3.1-5
```

# port flow-bridging

## Purpose

Set ports to use flow-based bridging.

## Format

**port flow-bridging** <*port-list*> | **all-ports**

## Mode

Configure

## Description

The **port flow-bridging** command changes the specified ports from using address-based bridging to using flow-based bridging. A port can use only one type of bridging at a time.

Each port has an L2 lookup table where MAC address or flows are stored.

- If the port is configured for address-based bridging (default), each L2 table entry consists of a MAC address and a VLAN ID.
- If the port is configured for flow-based bridging, each L2 table entry consists of a source MAC address, a destination MAC address, and a VLAN ID.

Suppose that a port on the GSR is connected to a hub that is connected to three workstations, A, B, and C. If each workstation is talking to one another and sending broadcast traffic, the L2 table on the GSR's port would contain the following entries for the workstations. Assume that the VLAN ID is "1" for all entries.

If the ports are configured for address-based bridging:

- MAC address A
- MAC address B
- MAC address C
- MAC broadcast address

If the ports are configured for flow-based bridging:

- MAC addresses A->B

- MAC addresses B->A
- MAC addresses B->C
- MAC addresses A->C
- MAC addresses C->A
- MAC addresses C->B
- MAC addresses A->broadcast
- MAC addresses B->broadcast
- MAC addresses C->broadcast

## Parameters

<*port-list*> | **all-ports** Specifies the ports you are changing to flow-based bridging. The keyword **all-ports** changes all the ports on the GSR to flow-based bridging.

## Restrictions

None.

## Examples

To configure Ethernet port et.3.7 for flow-based bridging:

```
gs/r(config)# port flow-bridging et.3.7
```

# port mirroring

## Purpose

Mirror traffic to a port for external analysis.

## Format

**port mirroring to** <*port*> **cpu-port-traffic | traffic-from [**<*port*>**| any]**
**traffic-to [**<*slot*>**| any]**

## Mode

Configure

## Description

The **port mirroring** command mirrors the type of traffic you specify to a port. By attaching a protocol analyzer to the port, you can observe and analyze the mirrored traffic.

## Parameters

<*port*>
Specifies the port to which you want to send the mirrored traffic. Attach your protocol analyzer to this port.

**cpu-port-traffic**
Mirrors traffic forwarded out by the Control Module. If you specify this option, you cannot specify the **traffic-from** or **traffic-to** options.

**traffic-from [**<*port*>**| any]**
Mirrors all traffic coming from the specified port. If you specify this option, you must also specify the **traffic-to** option.

**traffic-to [**<*port*>**| any]**
Mirrors traffic sent to the specified slot. The keyword **any** mirrors traffic sent to any of the GSR slots that contain line cards. If you specify this option, you must also specify the **traffic-to** option. To mirror traffic from the Control Module, use the **cpu-port-traffic** option.

## Restrictions

Note the following restrictions:

- Unless you are mirroring the traffic form the Control Module, you must specify either an input port or an output slot.
- You cannot specify the **any** keyword with both the **traffic-from** and **traffic-to** options at the same time.
- None of the ports on the slot containing the protocol analyzer port can send or receive traffic while port mirroring is taking place. When a port is selected to receive mirrored traffic, none of the other ports on the line card can be used for normal traffic. For this reason, the protocol analyzer port cannot be on the same slot (line card) as the mirrored port(s).
- Do not configure an interface on the protocol analyzer port.
- Port Mirroring is not currently supported for WAN ports.

## Examples

To copy traffic coming from port et.3.1 and going to any slot, enter the following command. The copied traffic is sent to port et.1.1, to which the protocol analyzer is attached.

```
gs/r(config)# port mirroring to et.1.1 traffic-from et.3.1 traffic-to
any
```

To copy traffic coming from any port and going to slot 4, enter the following command. The copied traffic is sent to port et.1.1, to which the protocol analyzer is attached.

```
gs/r(config)# port mirroring to et.1.1 traffic-from any traffic-to 4
```

To capture all traffic going to and from the Control Module, enter the following command. The copied traffic is sent to port et.1.1, to which the protocol analyzer is attached.

```
gs/r(config)# port mirroring to et.1.1 cpu-port-traffic
```

# port set

## Purpose

Set port operating mode and port speed.

## Format

**port set** [<*port-list*> | **all-ports**] **[duplex full | half]**
**[speed 10Mbps | 100Mbps |** <*number*>**] [auto-negotiation on | off]**
**[hash-mode m0 | m1 | m2 | m3] [wan encapsulation frame-relay | ppp]**

## Mode

Configure

## Description

Depending on the media type of a port, the **port set** command lets you set various parameters of each port.

For 10/100-Mbps Ethernet, you can set the following:

- Operating mode (half-duplex or full-duplex).
- Port speed (10-Mbps or 100-Mbps). This parameter applies only to ports on the 10/100 line cards.
- Hash mode

**Note:** By default, all ports use autosensing to detect the operating mode and speed of the network segment to which they are connected. If you use this command to set a port parameter, the setting disables autosensing for that parameter on the port. For example, if you set the speed of a segment to 10-Mbps, that segment no longer uses autosensing for the port speed and will always attempt to operate at 10-Mbps.

For Gigabit Ethernet, you can set the following:

- Auto-negotiation
- Hash mode

For WAN ports, you can set the following:

- Wan-encapsulation (either frame-relay or ppp)
- Speed (in Megabits per second)

**Note:** "Duplex", "autonegotiation", and "hash mode" are not applicable parameters for WAN interfaces.

## Parameters

*<port-list>* **| all-ports**
Specifies the ports. The **all-ports** keyword applies the settings you select to all the GSR ports.

**duplex full | half**
Sets the operating mode to half duplex or full duplex. This option is valid for 10/100 Mbps Ethernet only.

**speed 10Mbps | 100Mbps**
Sets the port speed to 10-Mbps or 100-Mbps. This option is valid for 10/100 Mbps Ethernet only.

**auto-negotiation on | off**
Turn on or off auto-negotiation for Gigabit Ethernet.

**hash-mode m0 | m1 | m2 | m3**
Set the Layer 2 hash mode for this port. Assuming a MAC address of the value 0011:2233:4455, the following describes the various hash modes:

- **m0** – 0011:2233:4455
- **m1** – 0011:2233:5544
- **m2** – 0011:3322:4455 (default hash mode)
- **m3** – 1100:2233:4455

## Restrictions

For 10/100 Mbps Ethernet, you must set both the operating mode and the speed. You cannot set one without setting the other. For Gigabit Ethernet, you can only turn on or off auto-negotiation. You cannot set the speed or duplex for Gigabit modules.

## Examples

To configure port et.1.5 to be 10 Mbps and half duplex:

```
gs/r(config)# port set et.1.5 speed 10mbps duplex half
```

To turn off auto-negotiation for the Gigabit port gi.4.2:

```
gs/r(config)# port set gi.4.2 auto-negotiation off
```

To set the Layer 2 hash mode for all ports to m0:

```
gs/r(config)# port set all-ports hash-mode m0
```

To set the speed for a HSSI ppp WAN port located on port 1 of slot 3:

```
gs/r(config)# port set hs.3.1 wan-encapsulation ppp speed 45000000
```

To set the speed for a serial frame relay WAN port located at port 4 of slot 2, VC 100:

```
gs/r(config)# port set se.2.4.100 wan-encapsulation frame-relay speed
    1500000
```

# port show bridging-status

## Purpose

Display the bridging status of GSR ports.

## Format

**port show bridging-status** <*port-list*> | **all-ports**

## Mode

Enable

## Description

The **port show bridging-status** command lets you display bridging-status information for GSR ports.

## Parameters

| | |
|---|---|
| <*port-list*> | **all-ports** | Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the GSR ports. |

## Restrictions

None.

## Example

To display the bridging status for all available ports:

```
gs/r# port show bridging-status all-ports
Port         Mgmt Status  phy-state     link-state Bridging Mode
----         -----------  ---------     ---------- -------------
et.4.1       No Action    Disabled      Link Down  Address
et.4.2       No Action    Disabled      Link Down  Address
et.4.3       No Action    Forwarding    Link Up    Address
et.4.4       No Action    Disabled      Link Down  Address
et.4.5       No Action    Disabled      Link Down  Address
et.4.6       No Action    Forwarding    Link Up    Address
et.4.7       No Action    Disabled      Link Down  Address
et.4.8       No Action    Disabled      Link Down  Address
```

# port show port-status

## Purpose

Display various information about specified ports.

## Format

**port show port-status** <*port-list*> | **all-ports**

## Mode

Enable

## Description

The **port show port-status command** lets you display port-status information for GSR ports.

## Parameters

| | |
|---|---|
| <*port-list*> | **all-ports** | Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the GSR ports. |

## Restrictions

None.

## Example

To display the port status for all ports on Ethernet module 5 (et.5):

```
gs/r# port show port-status et.5.*
Port         Port Type              Link    Duplex  Speed      Negotiation
----         ---------              ----    ------  -----      -----------
et.5.1        10/100-Mbit Ethernet   Up      Half    10 Mbits   Auto
et.5.2        10/100-Mbit Ethernet   Down    UNKNOWN UNKNOWN    Auto
et.5.3        10/100-Mbit Ethernet   Down    UNKNOWN UNKNOWN    Auto
et.5.4        10/100-Mbit Ethernet   Up      Full    100 Mbits  Auto
et.5.5        10/100-Mbit Ethernet   Down    UNKNOWN UNKNOWN    Auto
et.5.6        10/100-Mbit Ethernet   Down    UNKNOWN UNKNOWN    Auto
et.5.7        10/100-Mbit Ethernet   Down    UNKNOWN UNKNOWN    Auto
et.5.8        10/100-Mbit Ethernet   Up      Full    100 Mbits  Auto
```

# port show stp-info

## Purpose

Display Spanning Tree (STP) information for GSR ports.

## Format

**port show stp-info** <*port-list*> | **all-ports**

## Mode

Enable

## Description

The **port show stp-info** command lets you display Spanning-Tree information for GSR ports.

## Parameters

| | |
|---|---|
| <*port-list*> | **all-ports** Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the GSR ports. |

## Restrictions

None.

## Example

To display the spanning tree information for all available ports:

```
gs/r# port show stp-info all-ports
Designated
Port      Priority  Cost   STP       State      Designated-Bridge     Port
----      --------  ----   ---       -----      -----------------  ----------
et.1.1       128    00100  Enabled   Listening  8000:00e063111111     80 01
et.1.2       128    00100  Enabled   Listening  8000:00e063111111     80 02
et.1.3       128    00100  Enabled   Listening  8000:00e063111111     80 03
et.1.4       128    00100  Enabled   Listening  8000:00e063111111     80 04
et.1.5       128    00100  Enabled   Listening  8000:00e063111111     80 05
et.1.6       128    00100  Enabled   Listening  8000:00e063111111     80 06
et.1.7       128    00100  Enabled   Listening  8000:00e063111111     80 07
et.1.8       128    00100  Enabled   Listening  8000:00e063111111     80 08
```

# port show vlan-info

## Purpose

Display VLAN information for GSR ports.

## Format

**port show vlan-info** <*port-list*> | **all-ports**

## Mode

Enable

## Description

The **port show vlan-info** command lets you display VLAN information about GSR ports.

## Parameters

| | |
|---|---|
| <*port-list*> | **all-ports** | Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the GSR ports. |

## Restrictions

None

## Example

To display the VLAN information for all available ports:

```
gs/r# port show vlan-info all-ports
Port         Access Type     IP VLANs     IPX VLANs    Bridging VLANs
------       ---------       ---------    --------     --------------
et.4.1       access          DEFAULT      DEFAULT      DEFAULT
et.4.2       access          DEFAULT      DEFAULT      DEFAULT
et.4.3       access          DEFAULT      DEFAULT      DEFAULT
et.4.4       access          DEFAULT      DEFAULT      DEFAULT
et.4.5       access          DEFAULT      DEFAULT      DEFAULT
et.4.6       access          DEFAULT      DEFAULT      DEFAULT
et.4.7       access          DEFAULT      DEFAULT      DEFAULT
et.4.8       access          DEFAULT      DEFAULT      DEFAULT
```

# port show mirroring-status

## Purpose

Show the port mirroring status for slots in the GSR chassis.

## Format

**port show mirroring-status** <*slot*> | **all-slots**

## Mode

Enable

## Description

The **port show mirroring-status** command shows the following port mirroring status information for the specified chassis slots:

- Whether port mirroring is enabled
- The ports or slots that are being mirrored
- The mirroring mode (input port, output slot, or both)

## Parameters

| | |
|---|---|
| <*slot*> \| **all-slots** | Specifies the chassis slots for which you want to display port mirroring status. The **all-slots** keyword displays port mirroring status for all the slots in the chassis. |

## Restrictions

None.

## Examples

To display the port mirroring status for slot 5:

```
gs/r(config)# port show mirroring-status 5
```

# Chapter 33

# port mirroring Command

## Purpose

Apply port mirroring to one or more target ports on a GSR to monitor their activity.

## Format

**port mirroring monitor-port** <*port number*> **target-port** <*port list*>

## Mode

Configure

## Description

The **port mirroring** command allows you to monitor the activity of one or more ports on a GSR via a single port.

## Parameters

**monitor-port** <*port number*>
The port you will use to monitor activity on the port(s) designated in the target port portion of the command line.

**target-port** *<port list>*

The port(s) for which you want to monitor activity. You can specify a single port or a comma-separated list of ports.

## Restrictions

Even though multiple target ports may be defined for a given GSR, only one monitor port may be defined. Also, DIGITAL recommends that you monitor Gigabit ports through other Gigabit ports—you would almost certainly experience speed-inconsistency-related problems monitoring a Gigabit port through a 10Base-T or 100Base-TX port.

### Known Problems

- Packets that are lost due to CRC and BUFFER_OVERFLOW errors are not mirrored to the monitor-port.
- In the example below, routed packets from source A to destination B on link 2 are seen as leaving src mac of GSR when port 1.2 is being monitored.


## Examples

To mirror traffic on ethernet ports et.2.2-4 to port et1.2:

```
gs/r(config)# port mirroring monitor-port et.1.2 target-port
    et.2.2 et.2.3 et.2.4
```

After configuring et.1.2 as a monitor-port, et.1.2 is unusable for any other function in the system. This is indicated by a A LINK_DOWN message. However, et.1.2 is capable of transmitting TX packets and its LED will be lit while in operation.

# Chapter 34

# ppp Commands

The following commands allow you to define Point-to-Point Protocol (PPP) service profiles, and specify and monitor PPP High-Speed Serial Interface (HSSI) and standard serial ports.

## Command Summary

Table 23 lists the PPP commands. The sections following the table describe the command syntax.

**Table 23. ppp commands**

| |
|---|
| **ppp apply service** <*service name*> **ports** <*port list*> |
| **ppp define service** <*service name*> **[bridging enable \| disable] [high-priority-queue-depth** <*number*>**] [ip enable \| disable] [ipx enable \| disable] [lcp-magic on \| off] [low-priority-queue-depth** <*number*>**] [max-configure** <*number*>**] [max-failure** <*number*>**] [max-terminate** <*number*>**] [med-priority-queue-depth** <*number*>**] [red on \| off] [red-maxTh-high-prio-traffic** <*number*>**] [red-maxTh-low-prio-traffic** <*number*>**] [red-maxTh-med-prio-traffic** <*number*>**] [red-minTh-high-prio-traffic** <*number*>**] [red-minTh-low-prio-traffic** <*number*>**] [red-minTh-med-prio-traffic** <*number*>**] [retry-interval** <*number*>**] [rmon on \| off]** |
| **ppp restart lcp-ncp ports** <*port list*> |
| **ppp set peer-addr** <*IP address*> **ports** <*port*> |
| **ppp set ppp-encaps-bgd ports** <*port list*> |
| **ppp show service** <*service name*> **\| all** |
| **ppp show stats port** <*port*> **[bridge-ncp] [ip-ncp] [link-status]** |

# ppp apply service

## Purpose

Apply a pre-defined service profile to an interface.

## Format

**ppp apply service** <*service name*> **ports** <*port list*>

## Mode

Configure

## Description

Issuing the **ppp apply service ports** command allows you to apply a previously defined service profile to a given PPP WAN port.

## Parameters

| | |
|---|---|
| <*service name*> | The name of the previously defined service you wish to apply to the given port(s) or interfaces. |
| <*port list*> | The port(s) to which you wish to apply the pre-defined service profile. You can specify a single port or a comma-separated list of ports. |

## Restrictions

Usage is restricted to PPP WAN ports only.

## Example

To apply the service "s1" to slot 2, serial ports 1 and 2:

```
gs/r(config)# ppp apply service s1 ports se.2.1 se.2.2
```

# ppp define service

## Purpose

Define a service profile for WAN ports.

## Format

**ppp define service** <*service name*> **[bridging enable | disable] [high-priority-queue-depth** <*number*>**] [ip enable | disable] [ipx enable | disable] [lcp-magic on | off] [low-priority-queue-depth** <*number*>**] [max-configure** <*number*>**] [max-failure** <*number*>**] [max-terminate** <*number*>**] [med-priority-queue-depth** <*number*>**] [red on | off] [red-maxTh-high-prio-traffic** <*number*>**] [red-maxTh-low-prio-traffic** <*number*>**] [red-maxTh-med-prio-traffic** <*number*>**] [red-minTh-high-prio-traffic** <*number*>**] [red-minTh-low-prio-traffic** <*number*>**] [red-minTh-med-prio-traffic** <*number*>**] [retry-interval** <*number*>**] [rmon on | off]**

## Mode

Configure

## Description

The **ppp define service** command allows you to specify the following attributes for a newly created service profile:

- Activate and deactivate bridging, IP, and/or IPX for PPP WAN ports. If you do not specify any bridging, IP, or IPX protocols for PPP WAN ports, they are all activated by default. If you specify a bridging, IP, or IPX protocol, you *must* also explicitly define the behavior of the other two (i.e., **enabled** or **disabled**).
- The allowable PPP queue depth for high-, low-, and medium-priority items.
- Enable and disable the use of LCP magic numbers. Magic numbers are used to help detect loopback conditions.
- The maximum allowable number of unanswered/improperly answered configuration requests before determining that the connection to the peer is lost.
- The maximum allowable number of negative-acknowledgment responses for a given interface before declaring an inability to converge.
- The maximum allowable unacknowledged terminate requests before determining that the peer is unable to respond.
- Activate or deactivate Random Early Discard (RED) for PPP ports.

- The maximum and minimum threshold values for RED high-, low-, and medium-priority traffic.

  In general, DIGITAL recommends that the maximum threshold values be less than or equal to the respective high-, low-, or medium-priority queue depth. The minimum threshold values should be one-third of the respective maximum threshold.

- The number of seconds that will pass before a subsequent "resending" of the configuration request will be transmitted.

- Activate and deactivate RMON for PPP WAN ports.

## Parameters

*<service name>*
The name you wish to assign to the newly created service profile.

**bridging enable | disable**
Specifying the **enable** keyword activates bridging for PPP WAN ports. Specifying the **disable** keyword deactivates bridging for PPP WAN ports.

**high-priority-queue-depth** *<number>*
The number of items allowed in the PPP queue. You can specify a number between 1 and 65,535. DIGITAL recommends a value within the 5 - 100 item range. The default value is 20.

**ip enable | disable**
Specifying the **enable** keyword activates IP for PPP WAN ports. Specifying the **disable** keyword deactivates IP for PPP WAN ports.

**ipx enable | disable**
Specifying the **enable** keyword activates IPX for PPP WAN ports. Specifying the **disable** keyword deactivates IPX for PPP WAN ports.

**lcp-magic on | off**
Specifying the **on** keyword enables the use of LCP magic numbers. Specifying the **off** keyword disables the use of LCP magic numbers. The use of LCP magic numbers is enabled by default.

**low-priority-queue-depth** *<number>*
The number of items allowed in the PPP queue. You can specify a number between 1 and 65,535. DIGITAL recommends a value within the 5 - 100 item range. The default value is 20.

**max-configure** *<number>*
The maximum allowable number of unanswered requests. You can specify any number greater than or equal to 1. The default value is 10.

**max-failure** <*number*>
The maximum allowable number of negative-acknowledgment transmissions. You can specify any number greater than or equal to 1. The default value is 5.

**max-terminate** <*number*>
The maximum allowable number of unanswered/improperly answered connection-termination requests before declaring the link to a peer lost. You can specify any number greater than or equal to 1. The default value is 2.

**med-priority-queue-depth** <*number*>
The number of items allowed in the PPP queue. You can specify a number between 1 and 65,535. DIGITAL recommends a value within the 5 - 100 item range. The default value is 20.

**red on | off**
Specifying the **on** keyword enables RED for PPP WAN ports. Specifying the **off** keyword disables RED for PPP WAN ports.

**red-maxTh-high-prio-traffic** <*number*>
The maximum allowable threshold for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

**red-maxTh-low-prio-traffic** <*number*>
The maximum allowable threshold for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

**red-maxTh-med-prio-traffic** <*number*>
The maximum allowable threshold for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

**red-minTh-high-prio-traffic** <*number*>
The minimum allowable threshold for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

**red-minTh-low-prio-traffic** <*number*>
The minimum allowable threshold for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

**red-minTh-med-prio-traffic** <*number*>
The minimum allowable threshold for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

**retry-interval** <*number*>
The number of seconds between subsequent configuration request transmissions (the interval). You can specify any number greater than or equal to 1. The default value is 30.

**rmon on | off**
Specifying the **on** keyword enables RMON for PPP WAN ports. Specifying the **off** keyword disables RMON for PPP WAN ports.

## Restrictions

Usage is restricted to PPP WAN ports only.

## Example

To create a service profile named "pppserv4" with the following attributes:

- Bridging enabled
- IP and IPX enabled
- LCP magic numbers disabled
- RED disabled
- A retry interval of 20 seconds
- rmon enabled

then you would enter the following command line in Configure mode:

```
gs/r(config)# ppp define service pppserv4 bridging enable ip enable ipx
    enable lcp-magic off red off retry-interval 20 rmon on
```

# ppp restart lcp-ncp

## Purpose

Restart PPP LCP/NCP negotiation.

## Format

**ppp restart lcp-ncp ports** <*port list*>

## Mode

Enable

## Description

The **ppp restart lcp-ncp** command allows you to reset and restart the LCP/NCP negotiation process for PPP WAN ports.

## Parameters

<*port list*> The ports for which you would like to re-establish LCP/NCP negotiation.

## Restrictions

This command line is available only for PPP WAN ports.

## Example

To restart LCP/NCP negotiation on serial ports 1 and 2 of slot 4:

```
gs/r# ppp restart lcp-ncp ports se.4.1 se.4.2
```

# ppp set peer-addr

## Purpose

Set the peer address in case that IPCP/IPXCP can't resolve the address.

## Format

**ppp set peer-addr** <*IP address*> **ports** <*port*>

## Mode

Configure

## Description

Issuing the **ppp set peer-addr** command allows you to set the peer address if it can't be resolved by IPCP or IPXCP.

## Parameters

<*IP address*> The IP or IPX address you wish to use.

<*port*> The port to which you wish to assign the address.

## Restrictions

Usage is restricted to PPP port only.

## Example

To assign an ip address 10.1.1.1/16 to slot 2, serial port 1:

```
gs/r(config)# ppp set peer-addr ip-addr 10.1.1.1/16 ports se.2.1
```

# ppp set ppp-encaps-bgd

## Purpose

Force the ingress packets to be encapsulated in bridged format.

## Format

**ppp set ppp-encaps-bgd ports** *<port list>*

## Mode

Configure

## Description

Issuing the **ppp set ppp-encaps-bgd** command allows you to use bridged format encapsulation on a given ppp port.

## Parameters

*<port list>* The port(s) to which you wish to use bridged encapsulation. You can specify a single port or a comma-separated list of ports.

## Restrictions

Usage is restricted to ppp port only.

## Example

To force the bridged encapsulation to slot 2, serial ports 1 and 2:

```
gs/r(config)# ppp ppp-encaps-bgd ports se.2.1 se.2.2
```

# ppp show service

## Purpose

Displays PPP service profiles.

## Format

**ppp show service** <*service name*> | **all**

## Mode

Enable

## Description

The **ppp show service** command allows you to display one or all of the available PPP service profiles.

## Parameters

| | |
|---|---|
| <*service name*> | The service profile you wish to display. |
| **all** | Displays all of the available PPP service profiles. |

## Restrictions

None.

## Example

To display the available PPP service profiles named profile_4:

```
gs/r# ppp show service profile_4
```

# ppp show stats

## Purpose

Displays bridge NCP, IP NCP, and link-status parameters.

## Format

**ppp show stats port** <*port*> **[bridge-ncp] [ip-ncp] [link-status]**

## Mode

Enable

## Description

The **ppp show stats** command allows you to display parameters for bridge NCP, IP NCP, and link-status on PPP WAN ports. You can specify one, two, or three of the available parameter types.

## Parameters

| | |
|---|---|
| <*port*> | The PPP WAN port for which you wish to view bridge NCP, IP NCP, and/or link-status parameters. |
| **bridge-ncp** | Specifies that you wish to view bridging NCP parameters for the given port. |
| **ip-ncp** | Specifies that you wish to view IP NCP parameters for the given port. |
| **link-status** | Specifies that you wish to view link-status parameters for the given port. |

## Restrictions

None.

## Example

To display the available link-status and IP NCP parameters for the PPP WAN interface located at slot 4, port 1:

```
gs/r# ppp show stats port se.4.1 ip-ncp link-status
```

# Chapter 35

# qos Commands

The qos commands define and display Quality of Service (QoS) parameters. Use the command to classify Layer 2, Layer 3, and Layer 4 traffic into the following priorities:

- control
- high
- medium
- low

By assigning priorities to network traffic, you can ensure that critical traffic will reach its destination even if the exit ports for the traffic are experiencing greater than maximum utilization. Use the **qos set l2**, **qos set ip**, and **qos set ipx** commands to assign priorities for Layer-2, IP, and IPX traffic respectively.

## Flows

For Layer 3 (IP and IPX) traffic, you can define "flows", blueprints or templates of IP and IPX packet headers.

- The IP fields are source IP address, destination IP address, UDP/TCP source port, UDP/TCP destination port, TOS (Type of Service), transport protocol (TCP or UDP) and a list of incoming interfaces.
- The IPX fields are source network, source node, destination network, destination node, source port, destination port, and a list of incoming interfaces.

The flows specify the contents of these fields. If you do not enter a value for a field, a wildcard value (all values acceptable) is assumed for the field.

## Precedence

A precedence from 1 – 7 is associated with each field in a flow. The GSR uses the precedence value associated with the fields to break ties if packets match more than one flow. The highest precedence is 1 and the lowest is 7. Here are the default precedences of the fields:

- **IP** – destination port (1), destination address (2), source port (3), source IP address (4), TOS (5), interface (6), protocol (7).
- **IPX** – destination network (1), source network (2), destination node (3), source node (4), destination port (5), source port (6), interface (7).

Use the **qos precedence ip** and **qos precedence ipx** commands to change the default precedences.

## Queuing Policies

You can use one of two queuing policies on the GSR:

- **strict priority** – assures the higher priorities of throughput but at the expense of lower priorities. For example, during heavy loads, low-priority traffic can be dropped to preserve throughout of control-priority traffic, and so on.
- **weighted fair queuing** – distributes priority throughput among the four priorities (control, high, medium, and low) based on percentages.

The GSR can use only one queuing policy at a time. The policy is used on the entire GSR. The default queuing policy is strict priority.

# Command Summary

Table 24 lists the **qos** commands. The sections following the table describe the command syntax.

**Table 24. qos commands**

| |
|---|
| **qos precedence [sip** <*num*>**] [dip** <*num*>**] [srcport** <*num*>**] [destport** <*num*>**]**<br>**[tos** <*num*>**] [protocol** <*num*>**] [intf** <*num*>**]** |
| qos precedence ipx [srcnet <*num*>] **[srcnode** <*num*>**]**<br>**[srcport** <*num*>**] [dstnet** <*num*>**] [dstnode** <*num*>**] [dstport** <*num*>**] [intf** <*num*>**]** |
| **qos set ip** <*name*> <*priority*> <*srcaddr/mask*> \| **any** <*dstaddr/mask*> \| **any** <*srcport*> \| **any**<br><*dstport*> \| **any** <*tos*> \| **any** <*interface-list*> \| **any** <*protocol*> |
| **qos set ipx** <*name*> <*priority*> <*srcnet*> \| **any** <*srcmask*> \| **any** <*srcport*> \| **any**<br><*dstnet*> \| **any** <*dstmask*> \| **any** <*dstport*> \| **any** <*interface-list*> \| **any** |

**Table 24. qos commands (Continued)**

| |
|---|
| **qos set l2 name** <*name*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*> **vlan** <*vlanID*> **in-port-list** <*port-list*> **priority control \| high \| medium \| low \|** <*trunk-priority*> |
| **qos set queuing-policy weighted-fair** |
| **qos set weighted-fair control** <*percentage*> **high** <*percentage*> **medium** <*percentage*> **low** <*percentage*> |
| **qos show ip** |
| **qos show ipx** |
| **qos show l2 all-destination all-flow ports** <*port-list*> **vlan** <*vlanID*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*> |

# qos precedence ip

## Purpose

Set the precedence of the IP flow fields.

## Format

**qos precedence ip [sip** <*num*>**] [dip** <*num*>**] [srcport** <*num*>**] [destport** <*num*>**] [tos** <*num*>**] [protocol** <*num*>**] [intf** <*num*>**]**

## Mode

Configure

## Description

The **qos precedence ip** command lets you set the QoS precedence for various flow fields in IP traffic. You can set a precedence from 1 – 7 for the following IP fields:

- IP source address
- IP destination address
- Source TCP or UDP port
- Destination TCP or UDP port
- Type of Service (TOS) for the packet
- Protocol (TCP or UDP)
- Incoming interface

The precedence 1 is the highest priority. IP interfaces or flow fields within IP packets that have a precedence of 1 are given first priority. The default priorities are as follows:

- destination port (1)
- destination address (2)
- source port (3)
- source IP address (4)
- TOS (5)

- interface (6)
- protocol (7).

## Parameters

**sip** <*num*>
Specifies the precedence of the source address field in IP flows. Specify a precedence from 1 – 7.

**dip** <*num*>
Specifies the precedence of the destination address field in IP flows. Specify a precedence from 1 – 7.

**srcport** <*num*>
Specifies the precedence of the source port field in IP flows. Specify a precedence from 1 – 7.

**dstport** <*num*>
Specifies the precedence of the destination port field in IP flows. Specify a precedence from 1 – 7.

**tos** <*num*>
Specifies the precedence of the TOS field in IP flows. Specify a precedence from 1 – 7.

**protocol** <*num*>
Specifies the precedence of the transport layer protocol name field in IP flows. Specify a precedence from 1 – 7.

**intf** <*num*>
Specifies the precedence of the IP interface based on the interface's name. Specify a precedence from 1 – 7.

## Restrictions

None.

## Examples

To change the precedence for fields within IP flows from the default precedences listed above:

```
gs/r(config)# qos precedence ip sip 3 dip 1 srcport 2 destport 4 tos 5
    protocol 6 intf 7
```

# qos precedence ipx

## Purpose

Set the precedence of the IPX flow fields.

## Format

qos precedence ipx [srcnet <*num*>] **[srcnode** <*num*>**]** **[srcport** <*num*>**]** **[dstnet** <*num*>**]** **[dstnode** <*num*>**]** **[dstport** <*num*>**]** **[intf** <*num*>**]**

## Mode

Configure

## Description

The **qos precedence ipx** command lets you set the precedence of the following fields in IPX flows.

- Source network
- Source port
- Source node
- Destination network
- Destination node
- Destination port
- Incoming interface

You can set the precedence of the following fields from 1 – 7. The precedence 1 has the highest priority and 7 has the lowest. The default priorities are as follows:

- destination network (1)
- source network (2)
- destination node (3)
- source node (4)
- destination port (5)

- source port (6)
- interface (7).

## Parameters

**srcnet** <*num*>
Specifies the precedence of the source network field in IPX flows. Specify a precedence from 1 – 7.

**srcport** <*num*>
Specifies the precedence of the source port field in IPX flows. Specify a precedence from 1 – 7.

**srcnode** <*num*>
Specifies the precedence of the source node field in IPX flows. Specify a precedence from 1 – 7.

**dstnet** <*num*>
Specifies the precedence of the destination network field in IPX flows. Specify a precedence from 1 – 7.

**dstnode** <*num*>
Specifies the precedence of the destination node field in IPX flows. Specify a precedence from 1 – 7.

**dstport** <*num*>
Specifies the precedence of the destination port field in IPX flows. Specify a precedence from 1 – 7.

**intf** <*num*>
Specifies the precedence of the IPX interface based on the interface's name. Specify a precedence from 1 – 7.

## Restrictions

None.

## Examples

To change the precedence for fields within IPX flows from the default precedences listed above:

```
gs/r(config)# qos precedence ipx srcnet 1 srcnode 2 srcport
    dstnet 3 srcport 4 dstnode 5 dstport 6 intf 7
```

# qos set ip

## Purpose

Set a priority for an IP flow.

## Format

**qos set ip** <*name*> <*priority*> <*srcaddr/mask*> | **any**
<*dstaddr/mask*> | **any** <*srcport*> | **any** <*dstport*> | **any** <*tos*> | **any** <*interface-list*> | **any**
<*protocol*>

## Mode

Configure

## Description

The **qos set ip** command sets the priority for an IP flow based on the following fields in the flow:

- Flow name
- Source IP address and network mask
- Destination IP address and network mask
- Source port
- Destination port
- TOS
- Interface list
- Transport layer protocol (TCP or UDP)

You can set the priority of each field to control, low, medium, or high. The default is low.

## Parameters

<*name*>
Specifies the IP flow name.

*<priority>*
Specifies the priority you are assigning to the flow parameters you specified from the list above. You can specify one of the following priorities:

**control** Assigns control priority to the IP flow parameters you have specified. This is the highest priority.

**high** Assigns high priority to the IP flow parameters you have specified.

**medium** Assigns medium priority to the IP flow parameters you have specified.

**low** Assigns low priority to the IP flow parameters you have specified. This is the default.

*<srcaddr/mask>* | **any**
Specifies the source IP address and network mask for which you are assigning a priority. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").

If you specify **any** instead of a network mask, the GSR assumes a wildcard "don't care" condition. If you do not specify a mask, then the GSR assumes a mask of 255.255.255.255. You cannot substitute the mask with the **any** keyword. The keyword **any** is for the entire *<srcaddr/mask>* pair.

*<dstaddr/mask>* | **any**
Specifies the destination IP address and network mask for which you are assigning a priority. The same requirements and restrictions for *<srcaddr/mask>* **apply to** *<dstaddr/mask>*.

If you specify **any** instead of a network mask, the GSR assumes a wildcard "don't care" condition. If you do not specify a mask, then the GSR assumes a mask of 255.255.255.255. You cannot substitute the mask with the **any** keyword. The keyword **any** is for the entire *<dstaddr/mask>* pair.

*<srcport>* | **any**
Specifies the source TCP or UDP port for which you are assigning a priority. Specify a port number from 1 – 65535 or **any** to allow any value.

*<dstport>* | **any**
Specifies the destination TCP or UDP port for which you are assigning a priority. Specify a port number from 1 – 65535 or **any** to allow any value.

*<tos>* | **any**
Specifies the TOS for which you are assigning a priority. Specify a number from 0– 15 or **any** to allow any value.

*<interface-list>* | **any**
Specifies one or more IP interface names for which you are assigning priority. If you specify a list, delimit the interface names with commas.

*<protocol>*
Specifies the transport layer protocol for which you are assigning priority. You can specify one of the following values:

**tcp** Assigns the priority parameters to the TCP protocol.

**udp** Assigns the priority parameters to the UDP protocol.

**any** Assigns the priority parameters to both the TCP and UDP protocols.

## Restrictions

None.

## Examples

The following command creates a flow called "flow1". This flow provides a template for an IP packet with the IP address 1.1.1.1, network mask 255.255.0.0, destination address 2.2.2.2 (and implied destination mask 255.255.255.255). The flow includes source TCP/UDP port 3010, destination port 3000, a TOS of 15, the interfaces mls1 and mls2, and the TCP protocol as transport layer. This very explicit flow has the highest priority—control.

```
gs/r(config)# qos set ip flow1 control 1.1.1.1/255.255.0.0 2.2.2.2 3010
    3000 15 mls1 mls2 tcp
```

# qos set ipx

## Purpose

Set a priority for an IPX flow.

## Format

**qos set ipx** <*name*> <*priority*> <*srcnet*> | **any** <*srcmask*> | **any** <*srcport*> | **any** <*dstnet*> | **any** <*dstmask*> | **any** <*dstport*> | **any** <*interface-list*> | **any**

## Mode

Configure

## Description

The **qos set ipx** command lets you set the priority for an IPX flow based on the following fields in the flow:

- Flow name
- Source network
- Source network mask
- Source port
- Destination network
- Destination network mask
- Destination port
- Interface list

You can set the priority of each field to control, low, medium, or high. The default is low.

## Parameters

<*name*>
Specifies the IPX flow name.

*<priority>*
Specifies the priority you are assigning to the flow parameters you specified from the list above. You can specify one of the following priorities:

**control** Assigns control priority to the IP flow parameters you have specified. This is the highest priority.

**high** Assigns high priority to the IP flow parameters you have specified.

**medium** Assigns medium priority to the IP flow parameters you have specified.

**low** Assigns low priority to the IP flow parameters you have specified. This is the default.

*<srcnet>* | **any**
Specifies the IPX source network and node address. Specify them in the following format: *<netaddr>***.***<macaddr>*; for example: `a1b2c3d4.aa:bb:cc:dd:ee:ff`.

If you specify **any** instead of a **.***<macaddr>*, the GSR assumes a wildcard value. All MAC addresses are then valid.

*<srcmask>* | **any**
Specifies the IPX source network mask. Specify the mask in hexadecimal digits. If you do not specify a mask value and instead use the value **any**, the GSR internally sets the mask to FFFFFFFF.

*<srcport>* | **any**
Specifies a port number from 1 – 65535 or any to allow any value.

*<dstnet>* | **any**
Specifies the IPX destination network and node address. The same requirements and restrictions for *<dstaddr> apply to <srcaddr>*.

*<dstmask>* | **any**
Specifies the IPX destination network mask. Specify the mask in hexadecimal digits or **any** to allow any value.

*<dstport>* | **any**
Specifies a port number from 1 – 65535 or any to allow any value.

*<interface-list>* | **any**
Specifies one or more IPX interface names for which you are assigning priority. If you specify a list, delimit the interface names with commas.

## Restrictions

None.

## Examples

The following command creates an IPX flow called “abc”. This flow gives a high priority to IPX traffic on interface mls1 from network 12345678.00:01:00:00:00:00, mask 0000ff00, port 55 to network 22222222.02:00:00:00:00:00, mask 0000ff00, port 65.

```
gs/r(config)# qos set ipx abc high 12345678.00:01:00:00:00:00 0000ff00
    55 22222222.02:00:00:00:00:00 0000ff00 65 mls1
```

# qos set l2

## Purpose

Configure priority for a Layer 2 flow.

## Format

**qos set l2 name** <*name*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*> **vlan** <*vlanID*> **in-port-list** <*port-list*> **priority control | high | medium | low |** <*trunk-priority*>

## Mode

Configure

## Description

The **qos set l2** command lets you set QoS priority on a Layer 2 flow. You can set priorities on the following fields in the flow:

- L2 flow name
- Source MAC address
- Destination MAC address
- VLAN ID
- Incoming port(s)

You can set the priority of each field in one of the following ways:

- The flow is assigned a priority within the switch. In this case you specify a priority of control, low, medium, or high. The default is low.
- The flow is assigned a priority within the switch, but in addition, if the exit ports are VLAN trunk ports, the flow is assigned an 802.1Q priority. In this case you specify a number from 1 – 7. The GSR maps the number to the four internal priorities as follows: 0 = low; 1, 2, or 3 = medium; 4, 5,or 6 = high; 7 = control.

## Parameters

**name** <*name*>
Specifies the L2 flow name.

**source-mac** <*MACaddr*>
Specifies the L2 source MAC address. *Specify the MAC address in either of the following formats:*

```
xx:xx:xx:xx:xx:xx
xxxxxx:xxxxxx
```

**dest-mac** <*MACaddr*>
Specifies the L2 destination MAC address.

**vlan** <*vlanID*>
Specifies the name of a VLAN.

**in-port-list** <*port-list*>
Specifies the GSR ports for which you are setting priority for this flow. The priority applies when the L2 packet enters the GSR on one of the specified ports. The priority does not apply to exit ports.

**priority control | high | medium | low |** <*trunk-priority*>
Specifies the priority you are assigning to the flow parameters you specified from the list above. You can specify one of the following priorities:

| | |
|---|---|
| **control** | Assigns control priority to the IPX flow parameters you have specified. This is the highest priority. |
| **high** | Assigns high priority to the IPX flow parameters you have specified. |
| **medium** | Assigns medium priority to the IPX flow parameters you have specified. |
| **low** | Assigns low priority to the IPX flow parameters you have specified. This is the default. |
| <*trunk-priority*> | Assigns n 802.1Q VLAN trunk priority when the exit port is a VLAN trunk port. The GSR maps the number to the four internal priorities as follows: 0 = low; 1, 2, or 3 = medium; 4, 5,or 6 = high; 7 = control. |

## Restrictions

None.

# qos set queuing-policy

## Purpose

Change the queueing policy from strict priority to weighted fair.

## Format

**qos set queuing-policy weighted-fair**

## Mode

Configure

## Description

The **qos set queuing-policy** command lets you override the default queuing policy (strict priority) in favor of weighted fair queuing. The queuing policy applies to all the QoS settings in GSR. Only one type of queuing policy can be active at a time.

To set the queuing policy back to strict priority, enter the following command:

```
gs/r(config)# no qos set queuing-policy weighted-fair
```

## Parameters

**weighted-fair** Sets the queuing policy to weighted fair.

## Restrictions

None.

# qos set weighted-fair

## Purpose

Set percentages for weighted-fair queuing.

## Format

**qos set weighted-fair control** <*percentage*> **high** <*percentage*> **medium** <*percentage*> **low** <*percentage*>

## Mode

Configure

## Description

The **qos set weighted-fair** command lets you set the percentage of GSR bandwidth allocated to the control, high, medium, and low priorities. The percentages apply to all ports. Make sure the total percentages for all four priorities equals 100. you cannot set a priority to 0%.

## Parameters

**control** <*percentage*>
Specifies the percentage of GSR bandwidth allocated to the control priority. Specify a number from 1 – 100. The default is 25.

**high** <*percentage*>
Specifies the percentage of GSR bandwidth allocated to the high priority. Specify a number from 1 – 100. The default is 25.

**medium** <*percentage*>
Specifies the percentage of GSR bandwidth allocated to the medium priority. Specify a number from 1 – 100. The default is 25.

**low** <*percentage*>
Specifies the percentage of GSR bandwidth allocated to the low priority. Specify a number from 1 – 100. The default is 25.

## Restrictions

The total percentages for all four QoS levels must equal 100%.

# qos show ip

## Purpose

Show QoS information for IP flows.

## Format

**qos show ip**

## Mode

Enable

## Description

The **qos show ip** command lets you display QoS information for IP flows.

## Parameters

None.

## Restrictions

None.

# qos show ipx

## Purpose

Show QoS information for IPX flows.

## Format

**qos show ipx**

## Mode

Enable

## Description

The **qos show ipx** command lets you display QoS information for IPX flows.

## Parameters

None.

## Restrictions

None.

# qos show l2

## Purpose

Show QoS information for L2 flows.

## Format

**qos show l2 all-destination all-flow ports** <*port-list*> **vlan** <*vlanID*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>

## Mode

Enable

## Description

The **qos show l2** command lets you display QoS information for L2 flows. You can filter the display according to the following:

- Destinations
- Flows
- Ports
- VLANs
- Source MAC addresses
- Destination MAC addresses
- Priority

## Parameters

**all-destination**
Filters the display to show all the L2 destination priorities.

**all-flow**
Filters the display to show all the L2 flow priorities.

**ports** <*port-list*>
Filters the display to show L2 priority information for specific ports.

**vlan** <*vlanID*>
Filters the display to show L2 priority information for specific VLANs.

**source-mac** <*MACaddr*>Filters the display to show L2 priority information for specific source MAC addresses.

**dest-mac** <*MACaddr*>
Filters the display to show L2 priority information for specific destination MAC addresses.

## Restrictions

None.

# qos show

## Purpose

Show QoS information for L2, IP, and IPX flows.

## Format

**qos show ip | ipx | l2 all-destination all-flow ports** <*port-list*> **vlan** <*vlanID*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>

## Mode

User or Enable

## Description

The **qos show** command lets you display QoS information for IP, IPX, and L2 flows. The command shows information for all IP and IPX flows. For L2 flows, you can filter the display according to the following:

- Destinations
- Flows
- Ports
- VLANs
- Source MAC addresses
- Destination MAC addresses
- Priority

## Parameters

**all-destination**
Filters the display to show all the L2 destination priorities.

**all-flow**
Filters the display to show all the L2 flow priorities.

**ports** <*port-list*>
Filters the display to show L2 priority information for specific ports.

**vlan** <*vlanID*>
Filters the display to show L2 priority information for specific VLANs.

**source-mac** <*MACaddr*>
Filters the display to show L2 priority information for specific source MAC addresses.

**dest-mac** <*MACaddr*>
Filters the display to show L2 priority information for specific destination MAC addresses.

## Restrictions

None.

# Chapter 36

# radius Commands

The **radius** commands let you secure access to the GSR using the Remote Authentication Dial-In User Service (RADIUS) protocol. When a user logs in to the GSR or tries to access Enable mode, he or she is prompted for a password. If RADIUS authentication is enabled on the GSR, it will contact a RADIUS server to verify the user. If the user is verified, he or she is granted access to the GSR.

**Note:** The GSR currently supports the Password Authentication Protocol (PAP) method of authentication but not the Challenge Handshake Authentication Protocol (CHAP) method.

## Command Summary

Table 25 lists the **radius** commands. The sections following the table describe the command syntax.

**Table 25. radius commands**

| |
|---|
| **radius accounting shell start \| stop \| all** |
| **radius authentication login \| enable** |
| **radius enable** |
| **radius set host** <*IPaddr*><br><br>**radius set [timeout** <*number*>**] [key** <*string*>**] [last-resort password \| succeed]** |
| **radius show stats \| all** |

# radius accounting shell

## Purpose

Causes an entry to be logged on the RADIUS server when a shell is stopped or started on the GSR.

## Format

**radius accounting shell start | stop | all**

## Mode

Configure

## Description

The **radius accounting shell** command allows you to track shell usage on the GSR. It causes an entry to be logged on the RADIUS server when a shell is started or stopped. You can specify that an entry be logged when a shell is started, when a shell is stopped, or when a shell is either started or stopped.

## Parameters

**start** Logs an entry when a shell is started.

**stop** Logs an entry when a shell is stopped

**all** Logs an entry when a shell is either started or stopped

## Restrictions

None.

## Example

To cause an entry to be logged on the RADIUS server when a shell is either started or stopped on the GSR:

```
radius accounting shell all
```

# radius authentication

## Purpose

Causes RADIUS authentication to be performed at either the GSR login prompt or when the user tries to access Enable mode.

## Format

**radius authentication login | enable**

## Mode

Configure

## Description

The **radius authentication** command allows you to specify when RADIUS authentication is performed: either when a user logs in to the GSR, or tries to access Enable mode.

## Parameters

**login** Authenticates users at the GSR login prompt.

**enable** Authenticates users when they try to access Enable mode.

## Restrictions

None.

## Example

To perform RADIUS authentication at the GSR login prompt:

```
radius authentication login
```

# radius enable

## Purpose

Enables RADIUS authentication on the GSR. RADIUS authentication is disabled by default on the GSR.

## Format

**radius enable**

## Mode

Configure

## Description

The **radius enable** command causes RADIUS authentication to be activated on the GSR. You set RADIUS-related parameters with the **radius set**, **radius accounting shell**, and **radius authorization** commands, then use the **radius enable** command to activate RADIUS authentication.

## Parameters

None.

## Restrictions

None.

## Example

The following commands set RADIUS-related parameters on the GSR. The commands are then activated with the **radius enable** command:

```
radius set host 207.135.89.15
radius set timeout 30
radius authentication login
radius accounting shell all
radius enable
```

# radius set

## Purpose

Sets parameters for authenticating the GSR through a RADIUS server.

## Format

**radius set host** <*IPaddr*>

**radius set [timeout** <*number*>**] [key** <*string*>**] [last-resort password | succeed]**

## Mode

Configure

## Description

The **radius set** command allows you to set RADIUS-related parameters on the GSR, including the IP address of the RADIUS server, how long to wait for the RADIUS server to authenticate the user, an encryption key, and what to do if the RADIUS server does not reply by a given time.

## Parameters

**host** <*IPaddr*> Is the IP address of a RADIUS server. You can enter up to five RADIUS servers. Enter one server per **radius set host** command.

**timeout** <*number*> Is the maximum time (in seconds) to wait for a RADIUS server to reply. The default is 3 seconds.

**key** <*string*> Is an encryption key to be shared with the RADIUS server.

**last-resort** Is the action to take if a RADIUS server does not reply within the time specified by the **timeout** parameter. Specify one of the following:

- **password** The user is prompted for the password set with **system set password** command (if one has been set).
- **succeed** Access to the GSR is granted.

## Restrictions

None.

## Example

The following commands specify that hosts 137.72.5.9 and 137.72.5.41 are RADIUS servers, and the GSR should wait no more than 30 seconds for a response from one of these servers. If a response from a RADIUS server doesn't arrive in 30 seconds, the user is prompted for the password that was set with the GSR **system set password** command.

```
radius set host 137.72.5.9
radius set host 137.72.5.41
radius set timeout 30
radius set last-resort password
```

# radius show

## Purpose

Displays information about RADIUS configuration on the GSR.

## Format

**radius show stats | all**

## Mode

Enable

## Description

The **radius show** command displays statistics and configuration parameters related to RADIUS configuration on the GSR. The statistics displayed include:

accepts Number of times each server responded and validated the user successfully.

rejects Number of times each server responded and denied the user access, either because the user wasn't known, or the wrong password was supplied.

timeouts Number of times each server did not respond.

## Parameters

**stats** Displays the accepts, rejects, and timeouts for each RADIUS server.

**all** Displays the configuration parameters set with the **radius set** command, in addition to the accepts, rejects, and timeouts for each RADIUS server.

## Restrictions

None.

## Example

To display configuration parameters and RADIUS server statistics:

```
radius show all
```

# Chapter 37

# reboot Command

The **reboot** command reboots the GSR.

## Format

**reboot**

## Mode

Enable.

## Parameters

None.

## Restrictions

None.

# Chapter 38

# rip Commands

The Routing Information Protocol, Version 1 and Version 2, (RIPv1 and RIPv2) is the most commonly used interior gateway protocol. RIP selects the route with the lowest metric as the best route. The metric is a hop count representing the number of gateways through which data must pass in order to reach its destination. The longest path that RIP accepts is 15 hops. If the metric is greater than 15, a destination is considered unreachable and the GSR discards the route. RIP assumes that the best route is the one that uses the fewest gateways, that is, the shortest path. RIPv1 is described in RFC 1058 and RIPv2 is described in RFC 1723.

## Command Summary

Table 26 lists the **rip** commands. The sections following the table describe the command syntax.

**Table 26. rip commands**

| |
|---|
| **rip add interface \| source-gateways \| trusted-gateways** <*hostname-or-IPaddr*> |
| **rip set auto-summary disable \| enable** |
| **rip set broadcast-state always \| choose \| never** |
| **rip set check-zero disable \| enable** |
| **rip set default-metric** <*num*> |
| **rip set interface** <*interfacename-or-IPaddr*> **\| all [receive-rip enable \| disable] [send-rip enable \| disable] [metric-in** <*num*>**] [metric-out** <*num*>**] [version 1 \| version 2 [type broadcast \| multicast]] authentication-method [none \| (simple \| md5 key-chain** <*num-or-string*>**)]** |
| **rip set poison-reverse disable \| enable** |

**Table 26. rip commands (Continued)**

| |
|---|
| **rip set preference** <*num*> |
| **rip show** <*option-list*> |
| **rip start** |
| **rip stop** |
| **rip trace [packets \| request \| response \| local-options] [detail] [send \| receive]** |

# rip add

## Purpose

Adds RIP entities.

**Note:** By default, RIP is disabled on all GSR interfaces. To enable RIP on an interface, you must use the **rip add interface** command.

## Format

**rip add interface** <*interfacename-or-IPaddr*>

**rip add source-gateways | trusted-gateways** <*hostname-or-IPaddr*>

## Mode

Configure

## Description

The **rip add** command lets you add the following RIP entities:

- Interfaces that will run RIP
- Routers that send RIP updates directly, rather than through broadcast or multicast
- Trusted gateways, from which the GSR will accept RIP updates. when you add trusted gateways, the GSR does not accept RIP updates from sources other than those trusted gateways.

## Parameters

**interface**
Informs the RIP process about the specified interfaces. You can specify a list of interface names or IP addresses or use the **all** keyword to specify all interfaces.

**source-gateways**
Adds a router that sends RIP updates directly, rather than using broadcasts or multicasts. You can specify a single interface name or IP address.

**Note:** Updates to source gateways are not affected by the RIP packet transmission state of the interface.

**trusted-gateway**
Adds a trusted source for RIP updates. When you add trusted gateways, the GSR will not accept RIP updates from any sources except the trusted gateways. You can specify a single interface name or IP address.

*<interfacename-or-IPaddr>*
The interface name or IP address of the interface, router, or gateway. You can specify a list or use the keyword **all** to specify all GSR interfaces.

*<hostname-or-IPaddr>*
The hostname or IP address of the source or trusted gateway.

## Restrictions

None.

# rip set auto-summary

## Purpose

Enables automatic summarization and redistribution of RIP routes.

## Format

**rip set auto-summary disable | enable**

## Mode

Configure

## Description

The **rip set auto-summary** command specifies that routes to subnets should be automatically summarized by the classful network boundary and redistributed into RIP.

## Parameters

**disable | enable**
Enables or disables automatic summarization and redistribution of RIP routes.

## Restrictions

None.

# rip set broadcast-state

## Purpose

Determines if RIP packets will be broadcast regardless of the number of interfaces present. This is useful when propagating static routes or routes learned from another protocol into RIP. In some cases, the use of broadcast when only one network interface is present can cause data packets to traverse a single network twice.

## Format

**rip set broadcast-state always | choose | never**

## Mode

Configure

## Description

The **rip set broadcast-state** command specifies whether the GSR broadcasts RIP packets regardless of the number of interfaces present.

## Parameters

**always | choose | never**
Specifies whether the GSR broadcasts RIP packets regardless of the number of interfaces present. Specify one of the following:

**always** Always sends RIP broadcasts regardless of the number of interfaces present.

**choose** Sends RIP broadcasts only if more than one interface is configured on the GSR. This is the default state.

**never** Never sends RIP broadcasts on attached interfaces.

## Restrictions

None.

# rip set check-zero

## Purpose

Specifies whether RIP should make sure that reserved fields in incoming RIP V1 packets are zero. Normally RIP will reject packets where the reserved fields are non-zero.

## Format

**rip set check-zero disable | enable**

## Mode

Configure

## Description

The **rip set check-zero** command specifies whether RIP should make sure that reserved fields in incoming RIP V1 packets are zero. RIP will reject packets where the reserved fields are non-zero.

- If you use the **disable** keyword, RIP does not check the reserved field.
- If you use the **enable** keyword, RIP on the GSR checks to ensure that the reserved fields in incoming RIP packets are zero. If the reserved field in a RIP packet is not zero, the GSR discards the packet. This is the default state.

## Parameters

**disable | enable**
Enables or disables checking of the reserved field.

## Restrictions

None.

# rip set default-metric

## Purpose

Defines the metric used when advertising routes via RIP that were learned from other protocols. If not specified, the default value is 16 (unreachable). This choice of values requires you to explicitly specify a metric in order to export routes from other protocols into RIP. This metric may be overridden by a metric specified in the export command.

**Note:** The metric 16 is equivalent in RIP to "infinite" and makes a route unreachable. You must set the default metric to a value other than 16 in order to allow the GSR to export routes from other protocols such as OSPF and BGP-4 into RIP.

## Format

**rip set default-metric** <*num*>

## Mode

Configure

## Description

The **rip set default metric** command defines the metric used when advertising routes via RIP that were learned from other protocols.

## Parameters

<*num*> Specifies the metric. Specify a number from 1 – 16. The default is 16.

## Restrictions

None.

# rip set interface

## Purpose

Set the RIP state, version, type of update messages, metric and authentication scheme used for each interface running RIP.

## Format

**rip set interface** <*interfacename-or-IPaddr*> | **all**

**[advertise-classfull enable | disable ]**

**[receive-rip enable | disable]**

**[send-rip enable | disable]**

**[metric-in** <*num*>**]**

**[metric-out** <*num*>**]**

**[version 1 | version 2 [type broadcast | multicast]]**

**[authentication-method none | (simple | md5**
**key-chain** <*num-or-string*>**)]**

## Mode

Configure

## Description

The **rip set interface** command lets you set the following parameters for RIP interfaces:

- Whether the interface will accept RIP updates
- Whether the interface will send RIP updates
- The RIP version (RIP V1 or RIP V2)
- The packet type used for RIP V2 updates (broadcast or multicast)
- The metric added to incoming RIP updates
- The metric added to outgoing RIP updates

- The key-chain for RIP update authentication
- The authentication method used for RIP updates (none, simple, or MD5)

## Parameters

<*interfacename-or-IPaddr*> | **all**
The interface names or IP addresses of the interfaces for which you are setting RIP parameters. Specify the **all** keyword if you want to set RIP parameters for all IP interfaces on the GSR.

**advertise-classfull enable | disable**
This command is used to announce a classfull network onto a subnetted RIP Version 1 interface having the same classfull network.

**receive-rip enable | disable**
Specifies whether the interface(s) can receive RIP updates. Specify **enable** if you want to receive RIP updates on the interface. Otherwise, select **disable**.

The default is **enable**.

**Note:** This option affects RIP updates sent from trusted gateways. If you specify **disable**, the GSR will not receive any RIP updates, including those sent from trusted gateways. If you specify **enable** *and* you have set up trusted gateways, the GSR will accept updates only from those trusted gateways.

**send-rip enable | disable**
Specifies whether the interface(s) can send RIP updates. Specify **enable** if you want to send RIP updates from this interface. Otherwise, specify **disable**.

The default is **enable**.

**Note:**This option does not affect the sending of updates to source gateways.

**metric-in** <*num*>
Specifies a metric that the interface adds to incoming RIP routes before adding them to the interface table. Specify a metric from 1 – 16. Use this option to make the GSR prefer RIP routes learned from the specified interfaces less than RIP routes from other interfaces. The default is 1.

**metric-out** <*num*>
Specifies a metric that the interface adds to outgoing RIP routes sent through the specified interfaces. The default is 0. Use this option to make other routers prefer other sources of RIP routes over this router.

**version 1 | version 2 [type broadcast | multicast]**
Specifies the RIP version used on the interface(s).

**broadcast**
Causes RIP V2 packets that are RIP V1-compatible to be broadcast on this interface.

**multicast**
Causes RIP V2 packets to be multicasted on this interface; this is the default.

**authentication-method none | (simple | md5 key-chain** *<num-or-string>***)**
The authentication method the interface uses to authenticate RIP updates. Specify one of the following:

**none**
The interface does not use any authentication.

**simple**
The interface uses a simple password in which an authentication key of up to 8 characters is included in the packet.

**md5**
The interface uses MD5 authentication. This method uses the MD5 algorithm to create a crypto-checksum of a RIP packet and an authentication key of up to 16 characters.

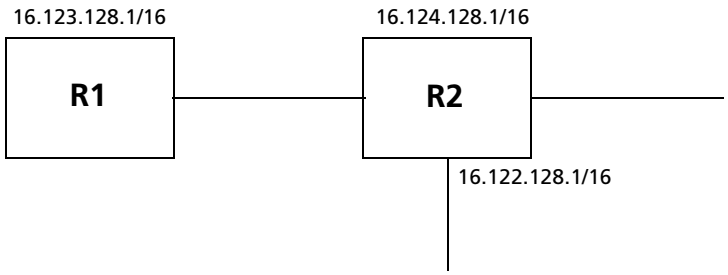
**Note:** If you choose the simple or md5 authentication method, you must also specify a key-chain identifier using the key-chain option.

**key-chain** *<num-or-string>*
The identifier of the key-chain containing the authentication keys. This parameter applies only if you specified simple or md5 for the authentication type.

## Restrictions

None.

## Example




In this example, router R1 has the following three interfaces:

1. It is connected to router R2 over interface 16.123.128.1/16. It is running RIP version 1 on this interface.
2. It has two other interfaces with the following addresses (16.124.128.1/16, 16.122.128.1/16).

3. Router R1 the entire class A network (16.0.0.0/8) behind it.

By default, router R1 would not announce a classful network (16.0.0.0/8) over a subnet (16.123.128.1/16). If that is something which is desired, then the below given command should be entered.

```
rip set interface 16.123.128.1 advertise-classfull enable | disable
```

Typically, a user would enable automatic summarization for RIP. This would create an implicit aggregate 16.0.0.0/8. If it is desired, that this classfull network is announced over a subnetted RIP Version 1 interface, then the above command should be entered.

# rip set poison-reverse

## Purpose

Enables poison reverse on all GSR interfaces.

## Format

**rip set poison-reverse disable | enable**

## Mode

Configure

## Description

The **rip set poison-reverse** command allows you to enable or disable poison reverse on all GSR interfaces. The GSR supports poison reverse as specified by RFC 1058.

**Note:** Turning on poison reverse will approximately double the amount of RIP updates.

## Parameters

**disable | enable**
Enables or disables poison reverse on the GSR.

## Restrictions

None.

# rip set preference

## Purpose

Sets the preference of routes learned from RIP. The default preference is 100. This preference may be overridden by a preference specified in the import command.

## Format

**rip set preference** <*num*>

## Mode

Configure

## Description

The **rip set preference** command sets the preference for destinations learned through RIP. The preference you specify applies to all IP interfaces for which RIP is enabled on the GSR. The default preference is 100. You can override this preference by specifying a different preference in an import policy.

## Parameters

<*num*> Specifies the preference. Specify a number from 0 – 255. The default is 100. Lower numbers have higher preference.

## Restrictions

None.

# rip show

## Purpose

Display RIP information.

## Format

**rip show** <*option-list*>

## Mode

Enable

## Description

The **rip show** command displays RIP information.

## Parameters

<*option-list*>
Specifies the RIP dump information you want to display. Specify one or more of the following:

**all**
Displays all RIP tables.

**globals**
Displays RIP globals.

**timers**
Displays RIP timers.

**interface**
Displays RIP interfaces.

**active-gateways**
Displays active gateways running RIP.

**interface-policies**
Displays RIP interface policies.

**import-policies**
Displays RIP import policies.

**export-policies**
Displays RIP export policies.

## Restrictions

None.

# rip start

## Purpose

Start RIP on the GSR.

**Note:** RIP is disabled by default.

## Format

**rip start**

## Mode

Configure

## Description

The **rip start** command starts RIP on all IP interfaces on the GSR for which RIP is enabled.

## Parameters

None.

## Restrictions

None.

# rip stop

## Purpose

Stop RIP on the GSR.

## Format

**rip stop**

## Mode

Configure

## Description

The **rip stop** command stops RIP on all IP interfaces on the GSR for which RIP is enabled.

## Parameters

None.

## Restrictions

None.

# rip trace

## Purpose

Trace RIP packets.

## Format

**rip trace [packets | request | response | local-options] [detail | send | receive]**

## Mode

Configure

## Description

The **rip trace** command traces the following sets of RIP packets:

- RIP request packets sent or received by the GSR
- RIP response packets sent or received by the GSR

Depending on the options you specify, you can trace all packets, request packets only, or receive packets only. In addition, you can select to trace the request packets, receive packets, or both that are sent by the GSR, received by the GSR, or all packets (both sent packets and received packets).

## Parameters

**packets** Traces all RIP packets, both request packets and response packets. This is the default.

**request** Traces only request packets, such as REQUEST, POLL and POLLENTRY packets.

**response** Traces only response packets.

For the **packets**, **request**, and **response** parameters, you can optionally specify one of the following:

**detail** Shows detailed information about the traced packets.

**receive** Shows information about traced RIP packets received by the GSR.

**send** Shows information about traced RIP packets sent by the GSR.

**Note: The default is to show both send and receive packets.**

**local-options** Sets trace options for this protocol only. These trace options are inherited from those set by the **ip-router global set trace options** command, or you can override them here. Specify one or more of the following:

**all** Turns on all tracing.

**general** Turns on normal and route tracing.

**state** Traces state machine transitions in the protocols.

**normal** Traces normal protocol occurrences.

**Note:**Abnormal protocol occurrences are always traced.

**policy** Traces application of protocol and user-specified policies to routes being imported and exported.

**task** Traces system processing associated with this protocol or peer.

**timer** Traces timer usage by this protocol or peer.

**route** Traces routing table changes for routes installed by this protocol or peer.

## Restrictions

None.

# Chapter 39

# rmon Commands

The **rmon** commands let you display and set parameters for RMON statistics on a per-port basis. RMON information corresponds to RFC 1757.

## Command Summary

Table 27 lists the **rmon** commands. The sections following the table describe the command syntax.

**Table 27. rmon commands**

| |
|---|
| **rmon alarm index** <*index-number*> **interval** <*seconds*>[**falling-event-index** <*num*>][**falling-threshold** <*num*>][**owner** <*string*>][**rising-event-index** <*num*>][**rising-threshold** <*num*>][**startup rising | falling | both**][**type absolute-value | delta-value**][**variable** <*string*>] |
| **rmon event index** <*index-number*> **type none | log | trap | both** [**community** <*string*>][**description** <*string*>][**owner** <*string*>] |
| **rmon history index** <*index-number*> **port** <*port*> [**interval** <*seconds*>][**owner** <*string*>][**samples** <*num*>] |
| **rmon show** [**alarm-monitor**] [**ether-stats** <*port-list*> | **all-ports**] [**event-log**] [**history-log**] [**memory-usage**] |

# rmon alarm

## Purpose

Configures the RMON I Alarm Control group.

## Format

**rmon alarm index** <*index-number*> **interval** <*seconds*>[**falling-event-index** <*num*>][**falling-threshold** <*num*>][**owner** <*string*>][**rising-event-index** <*num*>][**rising-threshold** <*num*>][**startup rising | falling | both**][**type absolute-value | delta-value**][**variable** <*string*>]

## Mode

Configure

## Description

The **rmon alarm** command sets various parameters of the RMON I Alarm Control Group.

## Parameters

<*index-number*>
Is a number that uniquely identifies an entry in the alarm table.

**interval** <*seconds*>
Specifies the sampling interval in seconds when statistical samples of variables are collected and compared to the rising and falling thresholds.

**falling-event-index** <*num*>
Is the action to be taken as defined by the row with this index in the event table when a falling threshold is crossed.

**falling-threshold** <*num*>
Specifies that the sample's value must be less than or equal to the threshold to trigger an alarm. When the sample's value is less than or equal to this threshold, and the value at the last sampling interval was greater than this threshold, a single event is generated.

**owner**<*string*>
Specifies the owner of the alarm resource; for example, an IP address, machine name or person's name.

**rising-event-index** <*num*>
Is the action to be taken as defined by the row with this index in the event table when a rising threshold is crossed.

**rising-threshold** <*num*>
Specifies that the sample's value must be greater than or equal to the threshold to trigger an alarm. When the sample's value is greater than or equal to this threshold, and the value at the last sampling interval was less than this threshold, a single event is generated.

**startup** <*keyword*>
Specifies the condition for which the alarm is to be generated. The condition can be one of the following:

| | |
|---|---|
| **rising** | Causes an alarm to be generated if the sampled variable is greater than or equal to the rising threshold. |
| **falling** | Causes an alarm to be generated if the sampled variable is less than or equal to the falling threshold. |
| **both** | Causes an alarm to be generated if the sampled variable is greater than or equal to the rising threshold or less than or equal to the falling threshold. |

**type** <*keyword*>
Specifies the method of sampling the selected variable and calculating the value to be compared against the thresholds. The sampling method can be one of the following:

| | |
|---|---|
| **absolute-value** | Monitor the absolute value over the sample interval of the variable against the threshold value. |
| **delta-value** | Monitor the change in value over the sample interval of the variable against the threshold value. |

**variable**<*string*>
Specifies the object identifier of the variable to be sampled. Only variables that resolve to an ASN.1 primitive type of INTEGER may be sampled.

## Restrictions

None.

## Examples

To cause an alarm event if the variable defined in alarm 10 crosses the rising threshold:

```
gs/r(config)# rmon alarm index 10 startup rising interval 30 variable
    1.3.6.1.2.1.5.14.0 rising-threshold 40 rising-event-index 1 type
    delta-value
```

To monitor the absolute value of the variable against a threshold value:

```
gs/r(config)# rmon alarm index 10 type absolute-value startup rising
    interval 30 variable 1.3.6.1.2.1.5.14.0 rising-threshold 40 rising-
    event-index 1 type delta-value
```

To specify Mike as the owner of alarm 10:

```
gs/r(config)# rmon alarm index 10 owner Mike type absolute-value startup
    rising interval 30 variable 1.3.6.1.2.1.5.14.0 rising-threshold 40
    rising-event-index 1
```

To specify a 5-second interval on alarm 10:

```
gs/r(config)# rmon alarm index 10 interval 5 type absolute-value startup
    rising interval 30 variable 1.3.6.1.2.1.5.14.0 rising-threshold 40
    rising-event-index 1
```

To specify the rising threshold at 10 on alarm 10:

```
gs/r(config)# rmon alarm index 10 rising-threshold 10 type delta-value
    startup rising interval 30 variable 1.3.6.1.2.1.5.14.0 rising-
    event-index 1
```

# rmon event

## Purpose

Configures the RMON I Event Control group.

## Format

**rmon event index** <*index-number*> **type none | log | trap | both** [**community** <*string*>][**description** <*string*>][**owner** <*string*>]

## Mode

Configure

## Description

The **rmon event** command sets various parameters of the RMON I Event Control Group.

## Parameters

<*index-number*>
Is a number that uniquely identifies an entry in the alarm table.

**community** <*string*>
Specifies the SNMP community string to be sent with the trap. If an SNMP trap is to be sent, it will go to the SNMP community specified in this string.

**description** <*string*>
Specifies a comment describing this event.

**owner** <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

**type** <*keyword*>
Specifies what action to be taken when the event occurs. The action can be one of the following:

**none** Causes no notification to be sent for the event.

**log** Causes an entry for the event to be made in the log table for each event.

**trap** Causes an SNMP trap to be sent to one or more management stations for the event.

**both** Causes both an entry to be made in the log table and an SNMP trap to be sent to one or more management stations.

## Restrictions

None.

## Examples

To set the event community string to public:

```
gs/r(config)# rmon event 10 community public type log
```

To add the description "num-pkts" to event 10:

```
gs/r(config)# rmon event 10 description num-pkts type trap
```

To specify Ed as the owner of event 10:

```
gs/r(config)# rmon event 10 owner Ed type trap
```

To send an SNMP trap when event 10 is triggered:

```
gs/r(config)# rmon event 10 type trap
```

# rmon history

## Purpose

Specifies the port, samples, interval and owner for RMON history.

## Format

**rmon history index** <*index-number*> **port** <*port*> [**interval** <*seconds*>][**owner** <*string*>][**samples** <*num*>]

## Mode

Configure

## Description

The RMON history group periodically records samples of variables and stores them for later retrieval. You use the r**mon history** command to specify the GSR port to collect data from, the number of samples, the sampling interval, and the owner.

## Parameters

<*index-number*>
Is a number that uniquely identifies an entry in the history table.

**interval** <*seconds*>
Specifies the sampling interval in seconds.

**owner** <*string*>
Specifies the owner of the history resource; for example, an IP address, machine name or person's name.

**port** <*port*>
Specifies the port from which to collect data.

**samples** <*num*>
Specifies the number of samples to be collected before wrapping counters.

## Restrictions

None.

## Example

To specify that port et.3.1 collect 60 samples at an interval of 30 seconds:

```
gs/r(config)# rmon history index 10 port et.3.1 samples 60 interval 30
```

# rmon show

## Purpose

Shows RMON information.

## Format

**rmon show** [**alarm-monitor**] [**ether-stats** <*port-list*> | **all-ports**] [**event-log**] [**history-log**] [**memory-usage**]

## Mode

Enable

## Description

The **rmon show** command shows various RMON statistics.

## Parameters

**alarm-monitor** Shows current alarm monitor status.

**ether-stats** <*port-list*> | **all-ports**
Show Ethernet statistics on one or more ports. Use the keyword **all-ports** to show Ethernet statistics on all the ports.

**event-log** Shows threshold events log.

**history-log** Shows port level historical statistics

**memory-usage** Shows RMON memory consumption.

## Restrictions

None.

## Examples

To show RMON statistics on port et.3.1:

```
gs/r(config)# rmon show ether-stats et.3.1
```

To show RMON memory consumption:

```
gs/r(config)# rmon show memory-usage
```

# Chapter 40

# save Command

The **save** command saves the configuration changes you have entered during the current CLI session. You can save the configuration commands in the scratchpad to the active configuration, thus activating changes. You then can save the active changes to the Startup configuration.

## Format

**save active | startup**

## Mode

Configure

**Note:** If you are in Enable mode, you still can save the active configuration changes to the Startup configuration file by entering the **copy active to startup** command.

## Description

Saves configuration changes.

- If you use the **active** keyword, uncommitted changes in the scratchpad are activated. The GSR accumulates configuration commands in the scratchpad until you activate them or clear them (or reboot). When you activate the changes, the GSR runs the commands.
- If you use the **startup** keyword, the configuration of the running system is saved in the Startup configuration file and re-instated by the server the next time you reboot.

## Parameters

**active | startup** Specifies the destination for the configuration commands you are saving.

## Restrictions

None.

# Chapter 41

# sfs Commands

The sfs commands set and display the following parameters:

- Cabletron Discovery Protocol (CDP) parameters

## Command Summary

Table 28 lists the port commands. The sections following the table describe the command syntax.

**Table 28. sfs commands**

| |
|---|
| **sfs enable cdp-hello** <*port-list*> \| **all-ports** |
| **sfs set cdp-hello transmit-frequency** |
| **sfs show cdp-hello port-status** <*port-list*> \| **all-ports** |
| **sfs show cdp-hello transmit-frequency** |

# sfs enable cdp-hello

## Purpose

Enabled the sending of CDP Hello packets.

## Format

**sfs enable cdp-hello** <*port-list*> | **all-ports**

## Mode

Configure

## Description

The **sfs enable cdp-hello** command enables the sending of CDP Hello packets. These are special packets sent out periodically by the router to announce itself to other devices or applications. CDP Hello packets can be enabled to be sent out to all available ports or selected ports only.

## Parameters

| | |
|---|---|
| <*port-list*> | **all-ports** Specifies the ports you want to enable CDP Hello packets. The **all-ports** keyword enables CDP Hello packets for all the GSR ports. |

## Restrictions

None.

## Examples

To enable the sending of CDP Hello packets on port 3 of slot 1:

```
gs/r(config)# sfs enable cdp-hello et.1.3
```

To send CDP Hello packets on all ports:

```
gs/r(config)# sfs enable cdp-hello all-ports
```

# sfs set cdp-hello transmit-frequency

## Purpose

Specify how often CDP Hello packets should be sent.

## Format

**sfs set cdp-hello transmit-frequency** <*secs*>

## Mode

Configure

## Description

The **sfs set cdp-hello transmit-frequency** command specifies how often CDP Hello packets should be sent. The interval is specified in seconds. The default transmit frequency is one packet every 5 seconds.

## Parameters

| | |
|---|---|
| <*secs*> | Specifies the interval in seconds between the transmission of CDP Hello packets. Acceptable value is 1-300. Default is 5 seconds. |

## Restrictions

None.

## Examples

To set the transmit frequency to 10 seconds:

```
gs/r(config)# sfs set cdp-hello transmit-frequency 10
```

# sfs show cdp-hello port-status

## Purpose

Display CDP Hello status of a port.

## Format

**sfs show cdp-hello port-status** <*port-list*> | **all-ports**

## Mode

Enable

## Description

The **sfs show cdp-hello port-status** command displays CDP Hello information of GSR ports.

## Parameters

| | |
|---|---|
| <*port-list*> | **all-ports** Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the GSR ports. |

## Restrictions

None.

## Examples

To display CDP Hello status on all GSR ports:

```
gs/r# sfs show cdp-hello port-status all-ports
```

# sfs show cdp-hello transmit-frequency

## Purpose

Display the transmit frequency of CDP Hello packets.

## Format

**sfs show cdp-hello transmit-frequency**

## Mode

Enable

## Description

The **sfs show cdp-hello transmit-frequency** command display the transmit frequency of CDP Hello packets on the GSR.

## Parameters

None.

## Restrictions

None.

## Examples

To display the transmit frequency of CDP Hello packets:

```
gs/r# sfs show cdp-hello transmit-frequency
```

# Chapter 42

# show Command

## Purpose

The **show** command displays the configuration of your running system.

## Format

**show**

## Mode

Configure

## Description

The **show** command displays the configuration of your running system as well as any non-committed changes in the scratchpad. Each CLI command is preceded with a number. This number can be used with the **negate** command to negate one or more commands. If you see the character **E** (for Error) immediately following the command number, it means the command did not execute successfully due of an earlier error condition. To get rid of the command in error, you can either negate it or fix the original error condition.

When viewing the active configuration file, the CLI displays the configuration file command lines with the following possible annotations:

- Commands without errors are displayed without any annotation.
- Commands with errors are annotated with an "E".

- If a particular command has been applied such that it can be expanded on additional interfaces/modules, then it is annotated with a "P". For example, if you enabled stp on all ports in the current system, however, the GSR contains only 1 module, then that particular command could be expanded at a later date when more modules have been added to the GSR.

  A command like **stp enable et.*.*** would be displayed as follows:

```
P: stp enable et.*.*
```

  indicating that it is only partially applied. If you add more modules to the GSR at a later date and then update the configuration file to encompass all of the available modules in the GSR, then the "P:" portion of the above command line would disappear when displaying this configuration file.

  If a potentially partial command, which was originally configured to encompass all of the available modules on the GSR, becomes only partially activated (after a hotswap or some such chassis reconfiguration), then the status of that command line will automatically change to indicate a partial completion status, complete with "P:".

**Note:** Commands with no annotation or annotated with a "P:" are not in error.

## Parameters

None.

## Restrictions

None.

## Examples

The following command shows when the running system was last modified (Jan 15) and from where (Console). It also shows that there are seven commands currently used to configure the system. In addition, command #7 is shown as having an error condition (**E**) possibly because the VLAN name *abc* is not defined. The actual cause of the error should

have been displayed earlier when the command was first committed to the running system. This is the time when the error was first detected.

```
gs/r(config)# show
!
! Last modified from Console on Thu Jan 15 10:33:30 1998
!
1 : vlan create IP1 ip
2 : vlan create IP2 ip
3 : vlan create IP3 ip
!
4 : interface create ip ssr0 address-netmask 10.1.13.1/24 vlan IP1
5 : interface create ip ssr1 address-netmask 10.1.11.1/24 vlan IP2
6 : interface create ip ssr2 address-netmask 10.1.12.1/24 vlan IP3
7E: interface create ip ssr3 address-netmask 10.1.63.12/24 vlan abc
```

To correct the error condition for command #7, a new command is entered to create a VLAN called IP4. The **show** command now displays not only the active configuration but also non-committed commands in the scratchpad.

```
gs/r(config)# show
!
! Last modified from Console on Thu Jan 15 10:33:30 1998
!
1 : vlan create IP1 ip
2 : vlan create IP2 ip
3 : vlan create IP3 ip
!
4 : interface create ip ssr0 address-netmask 10.1.13.1/24 vlan IP1
5 : interface create ip ssr1 address-netmask 10.1.11.1/24 vlan IP2
6 : interface create ip ssr2 address-netmask 10.1.12.1/24 vlan IP3
7E: interface create ip ssr3 address-netmask 10.1.63.12/24 vlan IP4

******** Non-committed changes in Scratchpad ********
1*: vlan create IP4 ip
```

The following series of short command line examples shows the use of the “partial” flag/annotation when viewing configuration file command line(s).

Suppose you have created VLAN “x” and added ports et.1.1 and et.2.1 to that VLAN. The display in the configuration file would look like this:

```
vlan add ports et1.1 et2.1 to x
```

Now, you decide to hotswap module 2 out of the system. The command line display then looks like the following:

```
P: vlan add ports et.1.1 et.2.1 to x
```

Suppose you now hotswap module 1 out of the system meaning that neither of the ports you configured for this command line exist in the GSR. You will see an “error” indicator/annotation in the command line display as follows:

```
E: vlan add ports et.1.1 et.2.1 to x
```

One more valuable piece of information: certain commands are always shown with a “partial” annotation in their configuration file command lines, as they are always able to be expanded. The following command line gives an example of this:

```
P: ip disable proxy-arp interface all
```

Since this particular command applies to all interfaces, it encompasses all existing interfaces as well as any that might be configured in the future.

# Chapter 43

# smarttrunk Commands

The **smarttrunk** commands let you display and set parameters for SmartTRUNK ports. SmartTRUNK ports are groups of ports that have been logically combined to increase throughput and provide link redundancy.

## Command Summary

Table 29 lists the **smarttrunk** commands. The sections following the table describe the command syntax.

**Table 29. smarttrunk commands**

| |
|---|
| **smarttrunk add ports** <*port list*> **to** <*smartTRUNK*> |
| **smarttrunk clear load-distribution** <*smartTRUNK*> |
| **smarttrunk create** <*smartTRUNK*> **protocol** <*protocol*> |
| **smarttrunk set load-policy on** <*smartTRUNK*> <*load-policy*> |
| **smarttrunk show** <*option*> |

# smarttrunk add ports

## Purpose

Adds physical ports to a SmartTRUNK.

## Format

**smarttrunk add ports** <*port list*> **to** <*smartTRUNK*>

## Mode

Configure

## Description

The **smarttrunk add ports** command allows you to add the ports specified in <*port list*> to a SmartTRUNK. The SmartTRUNK must already have been created with the **smarttrunk create** command. The ports in the SmartTRUNK must be set to full duplex.

## Parameters

| | |
|---|---|
| <*port list*> | Is one or more ports to be added to an existing SmartTRUNK. All the ports in the SmarTRUNK must be connected to the same destination. |
| <*smartTRUNK*> | Is the name of an existing SmartTRUNK. |

## Restrictions

Ports added to a SmartTRUNK must:

- Be set to full duplex
- Be in the same VLAN
- Have the same properties (L2 aging, STP state, and so on)

## Example

To add ports et.1.1, et.1.2, and et.1.3 to SmartTRUNK st.1:

```
gs/r(config)# smarttrunk add ports et.1.(1-3) to st.1
```

# smarttrunk clear load-distribution

## Purpose

Clears load distribution statistics for ports in a SmartTRUNK.

## Format

**smarttrunk clear load-distribution** <*smartTRUNK list*> | **all-smarttrunks**

## Mode

Enable

## Description

The **smarttrunk clear load-distribution** command is used in conjunction with the **smarttrunk show distribution** command, which gathers statistics for the transmitted bytes per second flowing through the SmartTRUNK and each port in it. The **smarttrunk clear load-distribution** command lets you reset load distribution statistics to zero.

## Parameters

<*smartTRUNK list* > Is the name of one or more existing SmartTRUNKs.

**all-smarttrunks** Causes load distribution information to be cleared for all SmartTRUNKs.

## Restrictions

None.

## Example

To clear load distribution information from SmartTRUNK st.1:

```
gs/r# smarttrunk clear load-distribution st.1
```

# smarttrunk create

## Purpose

Creates a SmartTRUNK and specifies a control protocol for it.

## Format

**smarttrunk create** <*smartTRUNK list*> **protocol no-protocol | huntgroup**

## Mode

Configure

## Description

The **smarttrunk create** command allows you to create a SmartTRUNK logical port. Once you have created a SmartTRUNK port, you add physical ports to it with the s**marttrunk add ports** command.

SmartTRUNKs on the GSR are compatible with the DEC Hunt Groups control protocol. If you are connecting the SmartTRUNK to another GSR, you can specify that the SmartTRUNK use this control protocol. SmartTRUNKing and Hunt Groups are comprised of two protocols:

- Logical Link Aging Protocol (LLAP) – Assists in learning and aging
- Physical Link Affinity Protocol (PLAP) – Monitors and maintains the trunking states

SmartTRUNKs are also compatible with devices that do not support the Hunt Groups control protocol, such as those that support Cisco's EtherChannel technology. If you are connecting a SmartTRUNK to devices that do not support Hunt Groups, no control protocol is used. You must specify the **no-protocol** keyword in the **smarttrunk create** command.

## Parameters

| | |
|---|---|
| <*smartTRUNK*> | Is the name of the SmartTRUNK to create. The name of the SmartTRUNK must be in the form st.*x*; for example, st.1. |
| **no-protocol** | Specifies that no control protocol be used. Use this keyword if the SmartTRUNK is connected to a device that does not support the DEC |

Hunt Group control protocol (that is, a device from a vendor other than Digital).

**huntgroup** Specifies that the DEC Hunt Group control protocol be used. Use this keyword if you are connecting the SmartTRUNK to another GSR.

## Restrictions

None.

## Example

The following command creates a SmartTRUNK named st.1, using the DEC Hunt Group control protocol.

```
gs/r(config)# smarttrunk create st.1 protocol huntgroup
```

# smarttrunk set load-policy

## Purpose

Specifies how traffic is distributed across the ports in a SmartTRUNK.

## Format

**smarttrunk set load-policy on** *<smartTRUNK list>* **| all-smarttrunks round-robin | link-utilization**

## Mode

Configure

## Description

The **smarttrunk set load-policy** command lets you specify how a SmartTRUNK distributes traffic among its ports. There are two options: **round-robin** (the default) and **link-utilization**.

Round-robin means that flows are assigned to ports on a sequential basis. The first flow goes to the first port in the SmartTRUNK, the second flow to the second port, and so on. Link-utilization means that a flow is assigned to the least-used port in the SmartTRUNK.

## Parameters

*<smartTRUNK list >* Is the name of one or more SmartTRUNKs.

| | |
|---|---|
| **all-smarttrunks** | Specifies that the command be applied to all SmartTRUNKs. |
| **round-robin** | Specifies that traffic be distributed evenly across all ports. |
| **link-utilization** | Specifies that packets should be sent to the least-used port in the SmartTRUNK. |

## Restrictions

None.

## Example

To specify that SmartTRUNK st.1 distribute flows sequentially among its component ports:

```
gs/r(config)# smarttrunk set load-policy on st.1 round-robin
```

# smarttrunk show

## Purpose

Displays information about SmartTRUNKs on the GSR

## Format

**smarttrunk show trunks**

**smarttrunk show distribution | protocol-state | connections** *<smartTRUNK list>* **| all-smarttrunks**

## Mode

Enable

## Description

The **smarttrunk show** command shows statistics about SmartTRUNKs on the GSR.

## Parameters

| | |
|---|---|
| **trunks** | Shows information about all SmartTRUNKs, including active and inactive ports, and the control protocol used. |
| **distribution** | Provides statistics on how traffic is distributed across the ports in a SmartTRUNK. |
| **protocol-state** | Shows information about the control protocol on a SmartTRUNK. |
| **connections** | Shows information about the SmartTRUNK connection, including the MAC address of the remote switch, and the module number and port number of each remote port. Connection information is reported only if the Hunt Group protocol is enabled for the SmartTRUNK. |

*<smartTRUNK list >* Is the name of one or more SmartTRUNKs.

| | |
|---|---|
| **all-smarttrunks** | Specifies that the command be applied to all SmartTRUNKs. |

## Restrictions

None.

## Examples

To display information about all SmartTRUNKs on the GSR:

```
gs/r# smarttrunk show trunks

Flags: D - Disabled  I - Inactive

SmartTRUNK Active Ports   Inactive Ports   Primary Port  Protocol  Load-Policy    Flags
---------- ------------   --------------   ------------  --------  -----------    -----
st.1                      et.3.(7-8)       None          None      RR
```

To show how traffic is distributed across the ports on SmartTRUNK st.1:

```
gs/r# smarttrunk show distribution st.1

SmartTRUNK  Member Port  Total (bytes/sec)    Port (bytes/sec)     % Load
----------  -----------  -----------------    ----------------     ------
st.1        et.2.4       7660268              2872592              37
st.1        et.2.5       7660268              1915084              25
st.1        et.2.6       7660268              2872592              37
```

To show information about the control protocol for SmartTRUNK st.1:

```
gs/r# smarttrunk show protocol-state st.1

SmartTRUNK  Protocol        State     Port    Port State
----------  --------        -----     ----    ----------
st.1        HuntGroup       Down      et.3.1  Negotiate
                                      et.3.2  Negotiate
```

To show connection information for all SmartTRUNKs:

```
gs/r# smarttrunk show connections all-smarttrunks
SmartTRUNK Local Port  Remote Switch        Remote Module   Remote Port  State
---------- ----------  -------------        -------------   -----------  -----
st.1       et.2.1      Digital A9:6E:57     3               1            Up
st.1       et.2.2      Digital A9:6E:57     3               2            Up
st.1       et.2.3      Digital A9:6E:57     3               3            Up
st.1       gi.3.1      Digital A9:6E:57     4               5            Up
st.2       et.2.4      --                   --              --           Up
st.2       et.2.5      --                   --              --           Up
st.2       et.2.6      --                   --              --           Up
```

**Note:** In the example above, SmartTRUNK st.2 has no control protocol enabled, so no connection information is reported.

# Chapter 44

# snmp Commands

The SNMP commands let you set and show SNMP parameters including SNMP community names and IP host targets for SNMP traps.

## Command Summary

Table 30 lists the **snmp** commands. The sections following the table describe the command syntax.

**Table 30. snmp Commands**

| |
|---|
| **snmp disable trap authentication \| link-up-down** |
| **snmp set chassis-id** <*chassis-name*> |
| **snmp set community** <*community-name*> privilege **read \| read-write** |
| **snmp set target** <*IP-addr*> **community** <*community-name*> **[status enable \| disable]** |
| **snmp show access \| all \| chassis-id \| community \| statistics \| trap** |
| **snmp stop** |

# snmp disable trap

## Purpose

Disable specific SNMP trap types.

## Format

**snmp disable trap authentication | link-up-down**

## Mode

Configure

## Description

The **snmp disable trap** command controls the types of traps the GSR emits based trap type. You can disable the following trap types:

- Authentication – use the **authentication** keyword to prevent the GSR from sending a trap each time it receives an invalid community string or invalid Telnet password.
- Link-state change – use the **link-up-down** keyword to prevent the GSR from sending a trap each time a port changes operational state.

## Parameters

**authentication** Disables authentication traps, which the GSR sends when it receives an invalid SNMP community string or Telnet password.

**link-up-down** Disables link-state change traps, which the GSR sends when a port's operational state changes.

## Restrictions

None.

# snmp set chassis-id

## Purpose

Set the GSR's chassis ID using SNMP.

## Format

**snmp set chassis-id** <*chassis-name*>

## Mode

Configure

## Description

The **snmp set chassis-id** command lets you set a string to give the GSR an SNMP identity.

## Parameters

<*chassis-name*> Is a string describing the GSR.

## Restrictions

None.

# snmp set community

## Purpose

Set an SNMP community string and specify the access privileges for that string.

## Format

**snmp set community** <*community-name*> **privilege read | read-write**

## Mode

Configure

## Description

The **snmp set community** command sets a community string for SNMP access to the GSR. SNMP management stations that want to access the GSR must supply a community string that is set on the switch. This command also sets the level of access to the GSR to read-only or read-write. Communities that are read-only allow SNMP GETs but not SNMP SETs. Communities that have read-write access allow both SNMP GETs and SNMP SETs.

## Parameters

**community** <*community-name*>
Character string for the community string.

**privilege read | read-write**
Access level. Specify one of the following:

| | |
|---|---|
| **read** | Allows SNMP GETs but not SNMP SETs. |
| **read-write** | Allows SNMP GETs and not SNMP SETs. |

## Restrictions

None.

## Example

To set the SNMP community string to "public," which has read-only access:

```
gs/r(config)# snmp set community public privilege read
```

# snmp set target

## Purpose

Sets the target IP address and community string for SNMP traps.

## Format

**snmp set target** <*IP-addr*> **community** <*community-name*> **[status enable | disable]**

## Mode

Configure

## Description

The **snmp set target** command specifies the IP address of the target server to which you want the GSR to send SNMP traps. Trap targets ar enabled by default but you can use the status argument to disable or re-enable a target.

**Note:** In general, community strings sent with traps should not have read-write privileges.

## Parameters

*<IP-addr>*
Is the IP address of the management station from which you want to be able to access the traps.

**Note:** The target IP address should be locally attached to the GSR. Cold start traps might not reach their destination if the target requires dynamic route table entries to be forwarded correctly. The GSR will retry every minute up to four minutes on the cold-start trap.

*<community-name>*
Is the name of the SNMP community for which you are setting the trap target.

**status enable | disable**
Re-enables or disables the target.

## Restrictions

None.

# snmp show

## Purpose

Shows SNMP information.

## Format

**snmp show access | all | chassis-id | community | statistics | trap**

## Mode

Enable

## Description

The **snmp show** command shows the following SNMP information:

- Community strings set on the GSR
- SNMP Statistics
- IP address of SNMP trap target server

## Parameters

| | |
|---|---|
| **access** | Displays the last five SNMP clients to access the GSR. |
| **all** | Displays all SNMP information (equivalent to specifying all the other keywords). |
| **chassis-id** | Displays the GSR's SNMP name. |
| **community** | Displays the GSR's community string. |
| **statistics** | Displays SNMP statistics. |
| **trap** | Displays the IP address of the trap target server. |

## Restrictions

None.

## Examples

The following command displays a log of SNMP access to the GSR. The host that accessed the GSR and the GSR system time when the access occurred are listed.

```
gs/r(confg)# snmp show access
SNMP Last 5 Clients:
     10.15.1.2      Tue Feb 10 18:42:59 1998
     10.15.1.2      Tue Feb 10 18:42:55 1998
     10.15.1.2      Tue Feb 10 18:42:56 1998
     10.15.1.2      Tue Feb 10 18:42:57 1998
     10.15.1.2      Tue Feb 10 18:42:58 1998
```

To display the SNMP identity of the GSR:

```
gs/r(confg)# snmp show chassis-id

SNMP Chassis Identity:
s/n 123456
```

To display the IP address of the trap target server:

```
gs/r(confg)# snmp show trap

Trap Table:
Index  Trap   Target Addr   Community String  Status
1.            10.15.1.2     public            enabled
2.            1.2.3.4       public123         disabled
3.            5.6.7.8       public20          disabled
```

# snmp stop

## Purpose

Stop SNMP access to the device.

## Format

**snmp stop**

## Mode

Configure

## Description

The **snmp stop** command stops SNMP access to the GSR. The GSR will still finish all active requests but will then disregard future requests. When you issue this command, UDP port 161 is closed.

## Parameters

None.

## Restrictions

None.

# Chapter 45

# statistics Commands

The **statistics** commands let you display statistics for various GSR features. You also can clear some statistics.

## Command Summary

Table 31 lists the statistics commands. The sections following the table describe the command syntax.

**Table 31. statistics commands**

| |
|---|
| **statistics clear port-errors** \| **port-stats** \| **rmon** <*port-list*> |
| **statistics show** <*statistic-type*> **[**<*port-list*>**]**<br>**Note:** Not all statistic types accept a port list. |

# statistics clear

## Purpose

Clear statistics.

## Format

**statistics clear** <*statistic-type*> <*port-list*>

## Mode

Enable

## Description

The **statistics clear** command clears port statistics, error statistics, or RMON statistics. When you clear statistics, the GSR sets the counters for the cleared statistics to 0, then begins accumulating the statistics again.

## Parameters

<*statistic-type*>
Type of statistics you want to clear. Specify one of the following:

**port-errors** Clears all error statistics for the specified port.

**port-stats** Clears all normal (non-error) statistics for the specified port.

**rmon** Clears all RMON statistics for the specified port.

<*port-list*>
The ports for which you are clearing statistics. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8). Specify **all-ports** to clear statistics for all the GSR ports.

## Restrictions

None.

# statistics show

## Purpose

Display statistics.

## Format

**statistics show** <*statistic-type*> <*port-list*>

## Mode

User or Enable

## Parameters

<*statistic-type*>
The type of statistics you want to display. Specify one of the following. Some statistics options apply system-wide while others apply only to the Control Module.

System-wide statistics:

**port-errors** Shows error statistics for ports.

**port-stats** Shows normal (non-error) port statistics.

**rmon** Shows RMON statistics.

**ip-interface** <*options*> Shows IP interface statistics.

**ipx-interface** <*options*>Shows IPX interface statistics.

For **ip-interface** and **ipx-interface**, the interface name, input and output frames, and input and output errors are displayed. However, you can use one or more of the following <*options*> to control the type of information displayed:

**packets** Displays packet statistics.

**bytes** Displays byte statistics.

**errors** Displays error statistics.

**input** If specified following one of the three options listed above, displays only input statistics for that option. Both input and output statistics are displayed by default.

**output** If specified following one of the three options listed above, displays only output statistics for that option.

**verbose** Displays all statistics.

Control-Module statistics:

**icmp** Shows ICMP statistics.

**ip** Shows IP statistics.

**ip-routing** Shows IP unicast routing statistics.

**ipx** Shows IPX statistics.

**ipx-routing** Shows IPX unicast routing statistics.

**multicast** Shows IP multicast statistics.

**tcp** Shows TCP statistics.

**udp** Shows UDP statistics.

***<port-list>***
For system-wide statistics options, the ports for which you are showing statistics. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8). Specify **all-ports** to show statistics for all the GSR ports.

## Restrictions

None.

# Chapter 46

# stp Commands

The stp commands let you display and change settings for the Spanning Tree Protocol STP).

## Command Summary

Table 32 lists the stp commands. The sections following the table describe the command syntax.

**Table 32. stp commands**

| |
|---|
| **stp enable port** <*port-list*> |
| stp set bridging [**forward-delay <*num*>] [hello-time <*num*>] [max-age** <*num*>] **[priority** <*num*>**]** |
| stp set port <*port-list*> **priority <*num*> port-cost <*num*>** |
| **stp show bridging-info** |

# stp enable port

## Purpose

Enable STP on one or more ports.

## Format

**stp enable port <*port-list*>**

## Mode

Configure

## Description

The **stp enable port** command enables STP on the specified ports.

## Parameters

| | |
|---|---|
| <*port-list*> | The ports on which you are enabling STP. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8). |

## Restrictions

None

# stp set bridging

## Purpose

Set STP bridging parameters.

## Format

stp set bridging [**forward-delay <*num*>**] [**hello-time <*num*>**] [**max-age** <*num*>] [**priority** <*num*>]

## Mode

Configure

## Description

The **stp set bridging** command lets you configure the following STP parameters:

- Bridging priority
- Hello time
- Maximum age
- Forward delay

## Parameters

**forward-delay** <*num*>
Sets the STP forward delay for the GSR. The forward delay is measured in seconds. Specify a number from 4– 30. The default is 15.

**hello-time** <*num*>
Sets the STP hello time for the GSR. The hello time is measured in seconds. Specify a number from 1– 10. The default is 2.

**max-age** <*num*>
Sets the STP maximum age for the GSR. Specify a number from 6–40. The default is 20.

**priority** <*num*>
Sets the STP bridging priority for the GSR. Specify a number from 0 – 65535. The default is 32768

## Restrictions

None.

## Examples

To set the bridging priority of Spanning Tree for the entire GSR to 1:

```
gs/r(config)# stp set bridging priority 1
```

# stp set port

## Purpose

Set STP port priority and port cost for ports.

## Format

**stp set port** <*port-list*> **priority <*num*> port-cost <*num*>**

## Mode

Configure

## Description

The **stp set port** command sets the STP priority and port cost for individual ports.

## Parameters

**port** <*port-list*>
The port(s) for which you are setting STP parameters. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

**priority <*num*>**
The priority you are assigning to the port(s). Specify a number from 0– 255. The default is 128.

**port-cost <*num*>**
The STP cost you are assigning to the port(s). Specify a number from 1– 65535. The default depends on the port speed: 1 for Gigabit (100-Mbps) ports, 10 for 100-Mbps ports, and 100 for 10-Mbps ports.

## Restrictions

None.

# stp show bridging-info

## Purpose

Display STP bridging information.

## Format

**stp show bridging-info**

## Mode

Enable

## Description

The **stp show bridging-info** command displays STP bridging information for the GSR.

## Parameters

None.

## Restrictions

None.

# Chapter 47

# system Commands

The **system** commands let you display and change system parameters.

## Command Summary

Table 33 lists the system commands. The sections following the table describe the command syntax.

**Table 33. system commands**

| |
|---|
| **system hotswap out** \| **in channel** <*number*> |
| **system image add** <*IPaddr-or-hostname*> <*file-name*> |
| **system image choose** <*file-name*> |
| **system image list** |
| **system image delete** <*file-name*> |
| **system promimage upgrade** <*hostname-or-IPaddr*> <*file-name*> |
| **system set bootprom netaddr** <*IPaddr*> **netmask** <*IPnetmask*> **tftp-server** <*IPaddr*> **[tftp-gateway** <*IPaddr*>**]** |
| **system set contact** <*system-contact*> |
| **system set date year** <*year*> **month** <*month*> **day** <*day*> **hour** <*hour*> **min** <*min*> **second** <*sec*> |
| **system set daylight-savings** |
| **system set dns** server <*IPaddr*>**[,**<*IPaddr*>**[,**<*IPaddr*>**]]** domain <*name*> |

**Table 33. system commands (Continued)**

| |
|---|
| **system set location** <*location*> |
| **system set login-banner** <*string*> \| **none** |
| **system set name** <*system-name*> |
| **system set password** <*mode*> <*string*> \| none |
| **system set poweron-selftest** [on \| quick] |
| **system set syslog** [**server** <*hostname-or-IPaddr*>] [**level** <*level-type*>] [**facility** <*facility-type*>] [**buffer-size** <*size*>] |
| **system set terminal baud** <*baud-rate*> \| **columns** <*num*> \| **rows** <*num*> |
| **system set timezone** <*timezone*> \| <*minutes*> |
| **system show** <*system-parm*> |

# system hotswap

## Purpose

Activates or deactivates a line card.

## Format

**system hotswap out | in slot** <*number*>

## Mode

Enable

## Description

The **system hotswap out** command deactivates a line card in a specified slot on the GSR, causing it to go offline. The command performs the same function as if you had pressed the Hot Swap button on the line card.

The **system hotwap in** command causes a line card that was deactivated with the **system hotwap out** command to go online again. The command performs the same function as if you had removed the card from its slot and inserted it again.

See the *DIGITAL GIGAswitch/Router User Reference Manual* for more information on hot swapping line cards.

## Parameters

**out**
Causes the line card in the specified slot to be deactivated.

**in**
Causes an inactive line card in the specified slot to be reactivated.

**Note:** The **system hotwap in** command works only on a line card that was deactivated with the **system hotwap out** command.

**slot** <*number*>
Is the slot where the line card resides. Specify 1-7 for the GSR-8 or 1-15 for the GSR-16.

## Restrictions

None.

## Example

To deactivate the line card in slot 7 on the GSR:

```
gs/r# system hotswap out slot 7
```

# system image add

## Purpose

Copy a system software image to the GSR.

## Format

**system image add** <*IPaddr-or-hostname*> <*file-name*>

## Mode

Enable

## Description

The **system image add** command copies a system software image from a TFTP server into the PCMCIA flash card on the Control Module. By default, if the GSR has two Control Modules, the system software image is copied to both Control Modules.

## Parameters

<*IPaddr-or-hostname*>
Is the IP address or host name of the TFTP server or a TFTP URL.

<*file-name*>
Is the file name of the system software image file.

**primary-cm**
Copies the system software image only to the primary Control Module.

**backup-cm**
Copies the system software image only to the secondary Control Module.

## Restrictions

None.

## Example

To download the software image file named `gsr8.tar.gz` from the TFTP server 10.1.2.3:

```
gs/r# system image add tftp://10.1.2.3/images/gsr8.tar.gz
```

# system image choose

## Purpose

Select a system software image file.

## Format

**system image choose** <*file-name*>

## Mode

Enable

## Description

The **system image choose** command specifies the system software image file on the PCMCIA flash card that you want the GSR to use the next time you reboot the system.

## Parameters

<*file-name*> The file name of the system software image file.

## Restrictions

None.

# system image delete

## Purpose

Deletes a system software image file from the PCMCIA flash card.

## Format

**system image delete** <*file-name*>

## Mode

Enable

## Description

The **system image delete** command deletes a system software image file from the PCMCIA flash card on the Control Module.

## Parameters

<*file-name*> The file name of the system software image file you want to delete.

## Restrictions

None.

# system image list

## Purpose

Lists the system software image files on the PCMCIA flash card.

## Format

**system image list**

## Mode

Enable

## Description

The **system image list** command lists the system software image files contained on the PCMCIA flash card on the Control Module.

## Parameters

None.

## Restrictions

None.

# system promimage upgrade

## Purpose

Upgrades the boot PROM software on the Control Module.

## Format

**system promimage upgrade** <*IPaddr-or-hostname*> <*file-name*>

## Mode

Enable

## Description

The **system promimage upgrade** command copies and installs a boot PROM software image from a TFTP server onto the internal memory on the Control Module. The boot PROM software image is loaded when you power on the GSR and in turn loads the system software image file.

## Parameters

<*IPaddr-or-hostname*>
The IP address or host name of the TFTP server or a TFTP URL.

<*file-name*>
The file name of the boot PROM software image file.

## Restrictions

None.

## Example

The command in the following example downloads a boot PROM image file from the TFTP server 10.50.89.88.

```
gs/r# system promimage upgrade tftp://10.50.89.88/qa/prom-upgrade
Downloading image ‘qa/prom-upgrade’ from host ‘10.50.89.88’
tftp complete
checksum valid. Ready to program.
flash found at 0xbfc00000
erasing...
programming...
verifying...
programming successful.
Programming complete.
```

# system set bootprom

## Purpose

Sets parameters for the boot PROM.

## Format

system set bootprom **netaddr** <*IPaddr*> **netmask** <*IPnetmask*>
**tftp-server** <*IPaddr*> **[tftp-gateway** <*Ipaddr*>**]**

## Mode

Configure

## Description

The **system set bootprom** command sets parameters to aid in booting the GSR's system software image remotely over the network. You can use this command to set the GSR's IP address, subnet mask, TFTP boot server address, and gateway address.

**Note:** These parameters apply only to the Control Module's en0 Ethernet interface.

## Parameters

**netaddr** <*IPaddr*>
The IP address the GSR uses during the boot exchange with the TFTP boot server.

**netmask** <*IPnetmask*>
The subnet mask the GSR uses during the boot exchange.

**tftp-server** <*IPaddr*>
The TFTP boot server's IP address.

**tftp-gateway** <*Ipaddr*>
The gateway that connects the GSR to the TFTP boot server.

## Restrictions

None.

## Example

The command in the following example configures the GSR to use IP address 10.50.88.2 to boot over the network from TFTP boot server 10.50.89.88.

```
gs/r(config)# system set bootprom netaddr 10.50.88.2 netmask
    255.255.0.0 tftp-server 10.50.89.88
```

# system set contact

## Purpose

Set the contact name and information for this GSR.

## Format

system set contact <*system-contact*>

## Mode

Configure

## Description

The **system set contact** command sets the name and contact information for the network administrator responsible for this GSR.

## Parameters

<*system-contact*>
A string listing the name and contact information for the network administrator responsible for this GSR. If the string contains blanks or commas, you must use the quotation marks around the string. (Example: **"Jane Doe, janed@corp.com, 408-555-5555 ext. 555"**.)

## Restrictions

None.

# system set date

## Purpose

Set the system time and date.

## Format

system set date year <*year*> month <*month*> day <*day*>
hour <*hour*> min <*min*> second <*sec*>

## Mode

Enable

## Description

The **system set date** command sets the system time and date for the GSR. The GSR keeps the time in a battery-backed realtime clock. To display the time and date, enter the **system show date** command.

## Parameters

**year** <*number*>
Four-digit number for the year. (Example: **1998**)

**month** <*month-name*>
Name of the month. You must spell out the month name. (Example: **March**)

**day** <*day*>
Number from 1 – 31 for the day.

**hour** <*hour*>
Number from 0 – 23 for the hour. (The number **0** means midnight.)

**minute** <*minute*>
Number from 0 – 59 for the hour.

**second** <*second*>
Number from 0 – 59 for the second.

## Restrictions

None.

# system set daylight-saving

## Purpose

Enable daylight saving for the local time zone.

## Format

**system set daylight-saving**

## Mode

Configure

## Description

If daylight savings time is in effect in the local time zone, use the **system set daylight-saving** command to enable it on the GSR. When daylight savings time is in effect, an additional hour is subtracted from your UCT offset. This command may be required if you use NTP (Network Time Protocol) to synchronize the system's real time clock. To disable daylight savings time on the GSR negate this command.

## Parameters

**None.**

## Restrictions

None.

## Example

When daylight savings time begins in the local time zone, enable it on the GSR with the following command:

```
gs/r(config)# system set daylight-saving
```

When daylight savings time ends in the local time zone, disable it on the GSR with the following command:

```
gs/r(config)# no system set daylight-saving
```

# system set dns

## Purpose

Configure the GSR to reach up to three DNS servers.

## Format

**system set dns server** [“]<IPaddr> [<*IPaddr*>] [<*IPaddr*>][“] **domain** <*name*>

## Mode

Configure

## Description

The **system set dns** command configures the GSR to reach up to three DNS servers. You also can specify the domain name to use for each DNS query by GSR.

## Parameters

[“]<*IPaddr*> [<*IPaddr*>] [<*IPaddr*>][“]
IP address of the DNS server. Specify the address in dotted-decimal notation. You can specify up to three DNS servers separated by single spaces in the command line.

**Note:** If you specify more than one IP address, you must surround the IP address specification with a set of quotes.

<*domain-name*>
Domain name for which the server is an authority.

## Restrictions

None.

## Examples

To configure a single DNS server and configure the GSR's DNS domain name to "digitalsys.com":

```
gs/r(config)# system set dns server 10.1.2.3 domain digitalsys.com
```

To configure three DNS servers and configure the GSR's DNS domain name to "digitalsys.com":

```
gs/r(config)# system set dns server "10.1.2.3 10.2.10.12 10.3.4.5"
    domain digitalsys.com
```

# system set location

## Purpose

Set the system location.

## Format

system set location <*location*>

## Mode

Configure

## Description

The **system set location** command adds a string describing the location of the GSR. The system name and location can be accessed by SNMP managers.

## Parameters

<*location*> A string describing the location of the GSR. If the string contains blanks or commas, you must use quotation marks around the string. (Example: **"Bldg C, network control room"**.)

## Restrictions

None.

# system set login-banner

## Purpose

Set the system login banner.

## Format

**system set login-banner** <*string*> | **none**

## Mode

Configure

## Description

The **system set login-banner** command configures the initial login banner that one sees when logging into the GSR. The banner may span multiple lines by adding line-feed characters in the string, "**\n**".

## Parameters

<*string*> Is the text of the login banner for the GSR. The banner may span multiple lines by having line-feed characters in the string, "**\n**".

**none** Specifies that no login-banner be used on the GSR.

## Restrictions

None.

## Example

The following example configures a multi-line login banner:

```
gs/r(config)# system set login-banner "Server network GSR\nUnauthorized
    Access Prohibited"
```

The next person to log into the GSR would see the following:

```
Server network GSR
Unauthorized Access Prohibited

Press RETURN to activate console...
```

If you do not want any login-banner at all, enter the following:

```
gs/r(config)# system set login-banner none
```

# system set name

## Purpose

Set the system name.

## Format

system set name <*system-name*>

## Mode

Configure

## Description

The **system set name** command configures the name of the GSR. The GSR name will use the name as part of the command prompt.

## Parameters

<*system-name*> The hostname of the GSR. If the string contains blanks or commas, you must use quotation marks around the string. (Example: **"Mega-Corp GSR #27"**.)

## Restrictions

None.

# system set password

## Purpose

Set passwords for various CLI access modes.

## Format

system set password <*mode*> <*string*> | none

## Mode

Configure

## Description

The **system set password** command sets or changes the passwords for the Login and Enable access modes.

**Note:** If a password is configured for the Enable mode, the GSR prompts for the password when you enter the **enable** command. Otherwise, the GSR displays a message advising you to configure an Enable password, then enters the Enable mode. From the Enable mode, you can access the Configure mode to make configuration changes.

## Parameters

<*mode*>
The access mode for which you are setting a password. Specify one of the following:

**login** The password required to start a CLI session. The GSR prompts for this password when the system finishes booting.

**enable** The password for entering the Enable mode.

<*string*> | **none**
The password. If you specify **none**, no password is required.

**Note:** You cannot use the string "none" as a password.

## Restrictions

The GSR stores passwords in the Startup configuration file. If you copy a configuration file from one GSR to another, the passwords in the file also are copied and will be required on the new GSR.

When you activate a new password by copying the password set command to the active configuration, the GSR replaces the command with a **system set hashed-password** command, which hides the password text in the configuration file so that the password is not visible to others if they examine the configuration file.

To remove a password, enter the following command while in Configure mode:

```
gs/r(config)# system set password <mode> none
```

# system set poweron-selftest

## Purpose

Specify the type of Power-On-Self-Test (POST) to perform during system bootup.

## Format

**system set poweron-selftest [on | quick]**

## Mode

Configure

## Description

The **system set poweron-selftest** command configures the type of Power-On-Self-Test (POST) the GSR should perform during the next system bootup. By default, no POST is performed during system bootup. To perform POST, you must use this command to specify which type of test to run, **quick** or **full**. Once POST enabled, to turn off POST, you simply negate this command (using the **negate** command).

## Parameters

**on** The GSR will perform a **full** test during the next system bootup.

**quick** The GSR will perform a **quick** test during the next system bootup.

## Restrictions

None.

# system set syslog

## Purpose

Identify a Syslog server to which the GSR can send Syslog messages

## Format

system set syslog [server <*hostname-or-IPaddr>]*
[level <*level-type*>] **[facility** <*facility-type*>**]**
**[buffer-size** <*size*>**]**

## Mode

Configure

## Description

The **system set syslog** command identifies the Syslog server to which the GSR should send system messages. You can control the type of messages to send as well as the facility under which the message is sent. The type of messages to send is based on the severity of the message (controlled by the option **level**). Messages can also be sent under a specific facility. There are 11 facilities supported by the GSR. On the Syslog server, you can decide what to do with these messages based on the level as well as the facility. For example, you might choose to discard the messages, write them to a file or send them out to the console.

The GSR keeps the last <*n*> messages in a local circular buffer. By default, this buffer keeps the last 10 Syslog messages. You can change the buffer size to hold anywhere from 10 – 50 messages. To view the current buffer size, enter the **system show syslog buffer** command.

## Parameters

*<hostname-or-IP-addr>*
Hostname or IP address of the SYSLOG server.

*<level-type>*
Level of messages you want the GSR to log. Specify one of the following:

**fatal**
Logs only fatal messages.

**error**
Logs fatal messages and error messages.

**warning**
Logs fatal messages, error messages, and warning messages. This is the default.

**info**
Logs all messages, including informational messages.

*<facility-type>*
Type of facility under which you want messages to be sent. By default, unless specified otherwise, messages are sent under facility *local7*. The facility-type can be one of the following:

| | |
|---|---|
| **kern** | kernel messages |
| **user** | user messages |
| **daemon** | daemon messages |
| **local0** | Reserved for local use |
| **local1** | Reserved for local use |
| **local2** | Reserved for local use |
| **local3** | Reserved for local use |
| **local4** | Reserved for local use |
| **local5** | Reserved for local use |
| **local6** | Reserved for local use |
| **local7** | Reserved for local use |

*<size>*
The Syslog message buffer size. The size specifies how many messages the Syslog buffer can hold. You can specify a number from 10 – 50, giving the buffer a capacity to hold from 10– 50 Syslog messages. The default is 10.

## Restrictions

None.

## Example

To log on error level messages to the syslog server on 10.1.43.77:

```
gs/r(config)# system set syslog server 10.1.43.77 level error
```

# system set terminal

## Purpose

Sets global terminal parameters.

## Format

system set terminal baud <*baud-rate*> | columns <*num*> | rows <*num*>

## Mode

Configure

## Description

The **system set terminal** command globally sets parameters for a serial console's baud rate, output columns, and output rows.

## Parameters

**baud** <*baud-rate*>
Sets the baud rate. You can specify one of the following:

- **300**
- **600**
- **1200**
- **2400**
- **4800**
- **9600**
- **19200**
- **38400**

**columns** <*num*>
Sets the number of columns displayed at one time.

**rows** <*num*>
Sets the number of rows displayed at one time.

## Restrictions

None.

## Example

The command in the following example sets the baud rate, number of columns, and number of rows for the management terminal connected to the System Control module.

```
gs/r(config)# system set terminal baud 38400 columns 132 rows 50
```

# system set timezone

## Purpose

Sets time zone information or time offset.

## Format

**system set timezone** <*timezone*> | <*minutes*>

## Mode

Configure

## Description

The **system set timezone** command sets the local time zone for the GSR. You can use one of the time zone keywords to specify the local time zone or specify the time offset in minutes. You must configure the time zone in order to use NTP (Network Time Protocol) to synchronize the GSR's real time clock.

## Parameters

<*timezone*>

Sets the time zone using one of the following keywrods:

| | |
|---|---|
| **est** | Eastern Standard Time (UCT -05:00) |
| **cst** | Central Standard Time (UCT -06:00) |
| **mst** | Mountain Standard Time (UCT -07:00) |
| **pst** | Pacific Standard Time (UCT -08:00) |
| **uct-12** | Eniwetok, Kawajalein (UCT -12:00) |
| **uct-11** | Midway Island, Samoa (UCT -11:00) |
| **uct-10** | Hawaii (UCT -10:00) |
| **uct-9** | Alasaka (UCT -09:00) |
| **uct-8** | Pacific Standard Time (UCT -08:00) |
| **uct-7** | Mountain Standard Time (UCT -07:00) |

| | |
|---|---|
| **uct-6** | Central Standard Time (UCT -06:00) |
| **uct-5** | Eastern Standard Time (UCT -05:00) |
| **uct-4** | Caracas, La Paz (UCT -04:00) |
| **uct-3** | Buenos Aires, Georgetown (UCT -03:00) |
| **uct-2** | Mid-Atlantic (UCT -02:00) |
| **uct-1** | Azores, Cape Verde Island (UCT -01:00) |
| **uct** | Greenwich, London, Dublin (UCT) |
| **uct+1** | Berlin, Madrid, Paris (UCT +01:00) |
| **uct+2** | Athens, Helsinki, Istanbul, Cairo (UCT +02:00) |
| **uct+3** | Moscow, Nairobi, Riyadh (UCT +03:00) |
| **uct+4** | Abu Dhabi, Kabul(UCT +05:00) |
| **uct+5** | Pakistan (UCT +05:00) |
| **uct+5:30** | India (UCT +05:30) |
| **uct+6** | Bangladesh (UCT +06:00) |
| **uct+7** | Bangkok, Jakarta (UCT +07:00) |
| **uct+8** | Beijing, Hong Kong, Singapore(UCT +08:00) |
| **uct+9** | Japan, Korea (UCT +09:00) |
| **uct+10** | Sydney, Guam (UCT +10:00) |
| **uct+11** | Solomon Is. (UCT +11:00) |
| **uct+12** | Fiji, Marshall Is. Auckland (UCT +12:00) |

*<minutes>*
Specify the time zone offset in minutes. Valid values are between -720 minutes to + 720 minutes.

## Restrictions

None.

## Example

To set the local time zone to Pacific Standard Time (UCT -8:00).

```
gs/r(config)# system set timezone pst
```

# system show

## Purpose

Show system information.

## Format

**system show** <*system-parm*>

## Mode

Enable

## Description

The **system show command** shows the active settings for the following system parameters:

- Active configuration (CLI configuration of the running system)
- Size of the Syslog message buffer
- Contact information for the GSR administrator (if you set one using the **system set contact** command)
- Current system time and date (if you set them using **system set date** command)
- Time that has elapsed since the GSR was rebooted and the system time and date when the last reboot occurred
- IP address(es) and domain name of DNS servers the GSR can use (if you set them using **system set dns** command)
- Hardware information
- Location of the GSR (if you set one using the **system set location** command)
- System name of the GSR (if you set one using the **system set name** command)
- IP address or hostname of SYSLOG server and the message level (if you set these parameters using the **system set syslog** command)
- Configuration changes in the scratchpad that are waiting for activation
- Software version running on the Control Module
- Last five Telnet connections to the GSR

## Parameters

<system-parm>
System parameter you want to display. Specify one of the following:

**active-config**
Shows the active configuration of the system

**buffer**
Shows how many Syslog messages the GSR's Syslog message buffer can hold

**bootlog**
Shows the contents of the boot log file, which contains all the system messages generated during bootup

**bootprom**
Shows boot PROM parameters for TFTP downloading of the system image. This information is useful only if you have configured the system to download the system image via TFTP.

**contact**
Shows the contact information (administrator name, phone number, and so on)

**date**
Shows the system time and date

**uptime**
Show how much time has elapsed time since the most recent reboot

**dns**
Shows the IP addresses and domain names for the DNS servers the GSR can use

**environmental**
Shows environmental information, such as temperature and power supply status.

**hardware**
Shows hardware information

**location**
Shows the GSR's location

**login-banner**
Shows the GSR's login banner. The login banner can be configured using the **system set login-banner** command.

**name**
Shows the GSR's name

**poweron-selftest-mode**
Shows the type of Power-On Self Test (POST) that should be performed, if any

**startup-config**
Shows the contents of the Startup configuration file

**switching-fabric**
Shows the status of the switching fabric module. This command is valid only for the GSR-16.

**syslog**
Shows the IP address of the SYSLOG server and the level of messages the GSR sends to the server

**telnet-access**
Lists the last five Telnet connections to the GSR

**terminal**
Shows the default terminal settings (number of rows, number of columns, and baud rate)

**timezone**
Shows the time zone offset from UCT in minutes.

**scratchpad**
Shows the configuration changes in the scratchpad. These changes have not yet been activated.

**version**
Shows the software version running on the GSR

## Restrictions

None.

# Chapter 48

# tacacs Commands

The **tacacs** commands let you secure access to the GSR using the Terminal Access Controller Access Control System (TACACS) protocol. When TACACS authentication is activated on the GSR, the user is prompted for a password when he or she tries to access Enable mode. The GSR queries a TACACS server to see if the password is valid. If the password is valid, the user is granted access to Enable mode.

## Command Summary

Table 34 lists the **tacacs** commands. The sections following the table describe the command syntax.

**Table 34. tacacs commands**

| |
|---|
| **tacacs enable** |
| **tacacs set host** <*IPaddr*><br><br>**tacacs set [timeout** <*number*>**] [last-resort password \| succeed]** |
| **tacacs show stats \| all** |

# tacacs enable

## Purpose

Enables TACACS authentication on the GSR. TACACS authentication is disabled by default on the GSR.

## Format

**tacacs enable**

## Mode

Configure

## Description

The **tacacs enable** command starts TACACS authentication on the GSR. When you issue this command, the TACACS-related parameters set with **tacacs set** commands become active.

## Parameters

None.

## Restrictions

None.

## Example

The following commands set TACACS-related parameters on the GSR. The commands are then activated with the **tacacs enable** command:

```
tacacs set host 207.135.89.15
tacacs set timeout 30
tacacs enable
```

# tacacs set

## Purpose

Sets parameters for authenticating the GSR through a TACACS server.

## Format

**tacacs set host** <*IPaddr*>

**tacacs set [timeout** <*number*>**] [last-resort password | succeed]**

## Mode

Configure

## Description

The **tacacs set** command allows you to set TACACS-related parameters on the GSR, including the IP addresses of up to five TACACS servers, how long to wait for the TACACS server to authenticate the user, and what to do if the TACACS server does not reply by a given time.

## Parameters

**host** <*IPaddr*> Is the IP address of a TACACS server. You can enter up to five TACACS servers. Enter one server per **tacacs set host** command.

**timeout** <*number*> Is the maximum time (in seconds) to wait for a TACACS server to reply. The default is 3 seconds.

**last-resort** Is the action to take if a TACACS server does not reply within the time specified by the **timeout** parameter. Specify one of the following:

- **password** The user is prompted for the Enable mode password set with **system set password** command (if one exists).
- **succeed** Access to the GSR is granted.

## Restrictions

None.

## Example

The following commands specify that hosts 137.72.5.9 and 137.72.5.41 are TACACS servers, and the GSR should wait no more than 30 seconds for a response from one of these servers. If a response from a TACACS server doesn't arrive in 30 seconds, the user is prompted for the password that was set with the GSR **system set password** command.

```
tacacs set host 137.72.5.9
tacacs set host 137.72.5.41
tacacs set timeout 30
tacacs set last-resort password
```

# tacacs show

## Purpose

Displays information about TACACS configuration on the GSR.

## Format

**tacacs show stats | all**

## Mode

Enable

## Description

The **tacacs show** command displays statistics and configuration parameters related to TACACS configuration on the GSR. The statistics displayed include:

accepts Number of times each server responded and validated the user successfully.

rejects Number of times each server responded and denied the user access, either because the user wasn't known, or the wrong password was supplied.

timeouts Number of times each server did not respond.

## Parameters

**stats** Displays the number of accepts, rejects, and timeouts for each TACACS server.

**all** Displays the configuration parameters set with the **tacacs set** command, in addition to the number of accepts, rejects, and timeouts for each TACACS server.

## Restrictions

None.

## Example

To display configuration parameters and TACACS server statistics:

```
tacacs show all
```

# Chapter 49

# tacacs-plus Commands

The **tacacs-plus** commands let you secure access to the GSR using the TACACS Plus protocol. When a user logs in to the GSR or tries to access Enable mode, he or she is prompted for a password. If TACACS Plus authentication is enabled on the GSR, it will contact a TACACS Plus server to verify the user. If the user is verified, he or she is granted access to the GSR.

**Note:** The GSR currently supports the Password Authentication Protocol (PAP) method of authentication but not the Challenge Handshake Authentication Protocol (CHAP) method.

## Command Summary

Table 35 lists the **tacacs-plus** commands. The sections following the table describe the command syntax.

**Table 35. tacacs-plus commands**

| |
|---|
| **tacacs-plus accounting shell start \| stop \| all** |
| **tacacs-plus authentication login \| enable** |
| **tacacs-plus enable** |
| **tacacs-plus set host** <*IPaddr*><br><br>**tacacs-plus set [timeout** <*number*>**] [key** <*string*>**] [last-resort password \| succeed]** |
| **tacacs-plus show stats \| all** |

# tacacs-plus accounting shell

## Purpose

Causes an entry to be logged on the TACACS Plus server when a shell is stopped or started on the GSR.

## Format

**tacacs-plus accounting shell start | stop | all**

## Mode

Configure

## Description

The **tacacs-plus accounting shell** command allows you to track shell usage on the GSR. It causes an entry to be logged on the TACACS Plus server when a shell is started or stopped. You can specify that an entry be logged when a shell is started, when a shell is stopped, or when a shell is either started or stopped.

## Parameters

**start** Logs an entry when a shell is started.

**stop** Logs an entry when a shell is stopped

**all** Logs an entry when a shell is either started or stopped

## Restrictions

None.

## Example

To cause an entry to be logged on the TACACS Plus server when a shell is either started or stopped on the GSR:

```
gs/r(config)# tacacs-plus accounting shell all
```

# tacacs-plus authentication

## Purpose

Causes TACACS Plus authentication to be performed at either the GSR login prompt or when the user tries to access Enable mode.

## Format

**tacacs-plus authentication login | enable**

## Mode

Configure

## Description

The **tacacs-plus authentication** command allows you to specify when TACACS Plus authentication is performed: either when a user logs in to the GSR, or tries to access Enable mode.

## Parameters

**login** Authenticates users at the GSR login prompt.

**enable** Authenticates users when they try to access Enable mode.

## Restrictions

None.

## Example

To perform TACACS Plus authentication at the GSR login prompt:

```
gs/r(config)# tacacs-plus authentication login
```

# tacacs-plus enable

## Purpose

Enables TACACS Plus authentication on the GSR. TACACS Plus authentication is disabled by default on the GSR.

## Format

**tacacs-plus enable**

## Mode

Configure

## Description

The **tacacs-plus enable** command causes TACACS Plus authentication to be activated on the GSR. You set TACACS Plus-related parameters with the **tacacs-plus set**, **tacacs-plus accounting shell**, and **tacacs-plus authorization** commands, then use the **tacacs-plus enable** command to activate TACACS Plus authentication.

## Parameters

None.

## Restrictions

None.

## Example

The following commands set TACACS Plus-related parameters on the GSR. The commands are then activated with the **tacacs-plus enable** command:

```
gs/r(config)# tacacs-plus set host 207.135.89.15
gs/r(config)# tacacs-plus set timeout 30
gs/r(config)# tacacs-plus authentication login
gs/r(config)# tacacs-plus accounting shell all
gs/r(config)# tacacs-plus enable
```

# tacacs-plus set

## Purpose

Sets parameters for authenticating the GSR through a TACACS Plus server.

## Format

**tacacs-plus set host** <*IPaddr*>

**tacacs-plus set [timeout** <*number*>**] [key** <*string*>**] [last-resort password | succeed]**

## Mode

Configure

## Description

The **tacacs-plus set** command allows you to set TACACS Plus-related parameters on the GSR, including the IP address of the TACACS Plus server, how long to wait for the TACACS Plus server to authenticate the user, an encryption key, and what to do if the TACACS Plus server does not reply by a given time.

## Parameters

**host** <*IPaddr*> Is the IP address of a TACACS Plus server. You can enter up to five TACACS Plus servers. Enter one server per **tacacs-plus set host** command.

**timeout** <*number*> Is the maximum time (in seconds) to wait for a TACACS Plus server to reply. The default is 3 seconds.

**key** <*string*> Is an encryption key to be shared with the TACACS Plus server.

**last-resort** Is the action to take if a TACACS Plus server does not reply within the time specified by the **timeout** parameter. Specify one of the following:

- **password** The user is prompted for the password set with **system set password** command (if one has been set).
- **succeed** Access to the GSR is granted.

## Restrictions

None.

## Example

The following commands specify that hosts 137.72.5.9 and 137.72.5.41 are TACACS Plus servers, and the GSR should wait no more than 30 seconds for a response from one of these servers. If a response from a TACACS Plus server doesn't arrive in 30 seconds, the user is prompted for the password that was set with the GSR **system set password** command.

```
gs/r(config)# tacacs-plus set host 137.72.5.9
gs/r(config)# tacacs-plus set host 137.72.5.41
gs/r(config)# tacacs-plus set timeout 30
gs/r(config)# tacacs-plus set last-resort password
```

# tacacs-plus show

## Purpose

Displays information about TACACS Plus configuration on the GSR.

## Format

**tacacs-plus show stats | all**

## Mode

Enable

## Description

The **tacacs-plus show** command displays statistics and configuration parameters related to TACACS Plus configuration on the GSR. The statistics displayed include:

accepts Number of times each server responded and validated the user successfully.

rejects Number of times each server responded and denied the user access, either because the user wasn't known, or the wrong password was supplied.

timeouts Number of times each server did not respond.

## Parameters

**stats** Displays the accepts, rejects, and timeouts for each TACACS Plus server.

**all** Displays the configuration parameters set with the **tacacs-plus set** command, in addition to the accepts, rejects, and timeouts for each TACACS Plus server.

## Restrictions

None.

## Example

To display configuration parameters and TACACS Plus server statistics:

```
gs/r# tacacs-plus show all
```

# Chapter 50

# traceroute Command

The **traceroute** command traces the path a packet takes to reach a remote host.

## Format

**traceroute** <*host*> **[max-ttl** <*num*>**] [probes** <*num*>**] [size** <*num*>**] [source** <*host*>**] [tos** <*num*>**] [wait-time** <*secs*>**] [verbose] [noroute]**

## Mode

User

## Description

The **traceroute** command traces the route taken by a packet to reach a remote IP host. The **traceroute** command examines the route taken by a packet traveling from a source to a destination. By default, the source of the packet is the GSR. However, one can specify a different source and track the route between it and a destination. The route is calculated by initially sending a probe (packet) from the source to the destination with a TTL of 1. Each intermediate router that is not able to reach the final destination directly will send back an ICMP Time Exceeded message. Subsequent probes from the source will increase the TTL value by 1. As each Time Exceeded message is received, the program keeps track of the address of each intermediate gateway. The probing stops when the packet reaches the destination or the TTL exceeds the **max-ttl** value.

## Parameters

*<host>*
Hostname or IP address of the destination

**max-ttl** *<num>*
Maximum number of gateways ("hops") to trace

**probes** *<num>*
Number of probes to send

**size** *<num>*
Packet size of each probe

**source** *<host>*
Hostname or IP address of the source

**tos** *<num>*
Type of Service value in the probe packet

**wait-time** *<secs>*
Maximum time to wait for a response

**verbose**
Displays results in verbose mode

**noroute**
Ignores the routing table and sends a probe to a host on a directly attached network. If the destination is not on the local network, an error is returned.

## Restrictions

None.

## Example

To display the route from the GSR to the host *othello* in verbose mode:

```
gs/r# traceroute othello verbose
```

# Chapter 51

# vlan Commands

The vlan commands let you perform the following tasks:

- Create VLANs
- List VLANs
- Add ports to VLANs
- Change the port membership of VLANs
- Make a VLAN port either a trunk port or an access port

## Command Summary

Table 36 lists the vlan commands. The sections following the table describe the command syntax.

**Table 36. vlan commands**

| |
|---|
| **vlan add ports** <*port-list*> **to** <*vlan-name*> |
| **vlan create** <*vlan-name*> <*type*> *id* <*num*> |
| **vlan list** |
| **vlan make** <*port-type*> <*port-list*> |

# vlan add ports

## Purpose

Add ports to a VLAN.

## Format

**vlan add ports** <*port-list*> **to** <*vlan-name*>

## Mode

Configure

## Description

The **vlan add ports** command adds ports to an existing VLAN. You do not need to specify the VLAN type when you add ports. You specify the VLAN type when you create the VLAN (using the **vlan create** command).

## Parameters

<*port-list*>
The ports you are adding to the VLAN. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

<*vlan-name*>
Name of the VLAN to which you are adding ports.

## Restrictions

The VLAN to which you add ports must already exist. To create a VLAN, use the **vlan create** command. An access port can be added to only one IP VLAN, one IPX VLAN, and one bridged-protocols VLAN.

# vlan create

## Purpose

Create a VLAN based on ports or protocol.

## Format

**vlan create** <*vlan-name*> <*type*> **id <*num*>**

## Mode

Configure

## Description

The **vlan create** command creates a VLAN definition. You can create a port-based VLAN or a protocol-based VLAN.

## Parameters

<*vlan-name*> Name of the VLAN. The VLAN name is a string up to 32 characters long.

**Note:** The VLAN name cannot begin with an underscore ( _ ) or the word "SYS_".

<*type*> The type of VLAN you are adding. The VLAN type determines the types of traffic the GSR will forward on the VLAN. Specify any combination of the first three types that follow ***or*** specify **port-based**:

**ip**
Use this VLAN for IP traffic

**ipx**
Use this VLAN for IPX traffic

**bridged-protocols**
Use this VLAN for bridged protocols

**port-based**
Use this VLAN for all the traffic types listed above (port-based VLAN)

**Note:** You can specify an combination of **ip**, **ipx**, and **bridged-protocols** or you can specify **port-based**; you cannot specify **port-based** with any of the other options.

**id <*num*>** ID of this VLAN. The ID must be unique. You can specify a number from 2 – 4093. If more than one GSR will be configured with the same VLAN, you must specify the same VLAN ID on each GSR.

## Restrictions

None.

# vlan make

## Purpose

Configures the specified ports into either trunk or access ports.

## Format

**vlan make <*port-type*> <*port-list*>**

## Mode

Configure

## Description

The **vlan make** command turns a port into a VLAN trunk or VLAN access port. A VLAN trunk port can forward traffic for multiple VLANs. Use trunk ports when you want to connect GSR switches together and send traffic for multiple VLANs on a single network segment connecting the switches.

## Parameters

*<port-type>*
The port type. You can specify one of the following types:

**trunk-port**
The port will forward traffic for multiple VLANs. The GSR will encapsulate all traffic in IEEE 802.1Q tag headers.

**access-port**
The port will forward traffic only for the VLANs to which you have added the ports and the traffic will be untagged. This is the default.

*<port-list>*
The ports you are configuring. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

## Restrictions

None.

# vlan show

## Purpose

Display a list of all VLANs active on the GSR.

## Format

vlan show

## Mode

User or Enable

## Description

The **vlan show** command lists all the VLANs that have been configured on the GSR.

## Parameters

None.

## Restrictions

None.